Destylization of Text with Decorative Elements

Yuting Ma
NLPR, Institute of Automation, CAS
School of Artificial Intelligence, UCAS
mayuting2018@ia.ac.cn

Fan Tang*
School of Artificial Intelligence, Jilin University
tangfan@jlu.edu.com

Weiming Dong*
NLPR, Institute of Automation, CAS
CASIA-LLVision Joint Lab
weiming.dong@ia.ac.cn

Changsheng Xu
NLPR, Institute of Automation, CAS
CASIA-LLVision Joint Lab
csxu@nlpr.ia.ac.cn

ABSTRACT

Style text with decorative elements has a strong visual sense, and enriches our daily work, study and life. However, it introduces new challenges to text detection and recognition. In this study, we propose a text destylized framework, that can transform the stylized texts with decorative elements into a type that is easily distinguishable by a detection or recognition model. We arranged and integrate an existing stylistic text data set to train the destylized network. The new destylized data set contains English letters and Chinese characters. The proposed approach enables a framework to handle both Chinese characters and English letters without the need for additional networks. Experiments show that the method is superior to the state-of-the-art style-related models.

CCS CONCEPTS

• **Applied computing** → **Fine arts**; Computer-assisted instruction; • **Computing methodologies** → *Image representations.*

KEYWORDS

Text destylization; Style transfer; Decorative elements

ACM Reference Format:
Yuting Ma, Fan Tang, Weiming Dong, and Changsheng Xu. 2021. Destylization of Text with Decorative Elements. In *ACM Multimedia Asia (MMAsia '20), March 7–9, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3444685.3446324

1 INTRODUCTION

Text destylization restores text with various complex styles to a style-less state, that can be handled by text detection and recognition approaches. With the development of deep learning, the replication and transfer of styles have become increasingly popular. Ordinary people can "create" their desired art forms without professional training. Text is a prominent visual element in 2D design, and is almost everywhere. Compared with ordinary text, text with artistic rendering are required by people. Artistic effects, such as color, outline, shadow, reflection, luminescence, and texture, are additional stylistic features of text. Artistic writing is a type of art widely used in design and media. Through color, texture, shading and other text effects, combined with additional decorative elements, artistic text becomes visually pleasing and can convey more semantic information vividly. Visual effects are commonly applied to text in graphic design because of their special role in visual design. These effects are also applied in painting synthesis and photography post processing.

Artificial intelligence has developed rapidly in the past decade, and various social software and self-media applications have been created and popularized. As a result, even ordinary people can become micro social media developers. At the same time, the application of numerous text styles and easy production have produced many different styles of text, such as posters, advertising design, e-commerce platforms and visual creation tasks. However, although these text styles make our visual experience colorful, they bring new challenges to automatic text detection and recognition. Traditionally, text detection and recognition approaches are designed for ordinary text and do not address rotation, bending, mirroring and other transformations. Text detection and recognition tasks can be challenging in an open environment, and cannot meet the increasingly complex needs of users. In related engineering tasks, the study of text destylization can partly improve the accuracy and speed of depth detection and recognition models, which have a wide range of application scenarios and practical usage [33].

Text destylization is a new problem, which is the reverse application of text style transfer. It can be viewed as a problem of transforming from the stylized domain to the source domain, but it has been disregarded by scholars.Yang et al. [30] were the first to raise the issue by designing a subnetwork that removed the basic style. The destylization problem with decorative elements has not been studied, due to the neglect of many important attributes, such as decorative elements, orientation, and rule structure. To solve this problem, this study pays special attention to decorative elements in text designs and proposes a new text style transformation framework to remove such elements.

In summary, our contributions are threefold. First, we define a new problem of text destylization with decorative elements and propose a new framework to solve this problem. The framework

*Co-corresponding authors

MMAsia '20, March 7–9, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8308-0/21/03…$15.00
https://doi.org/10.1145/3444685.3446324

can effectively solve the impact of decorative elements on partial pixel occlusion of text. Second, our framework can destylize English letters and Chinese characters without training additional subnetworks. Third, through the permutation and combination of an existing text style data set, our data set contains both English letters and Chinese characters. Experiments show that our model can handle not only a single character, but also several simple words.

2 RELATED WORK

2.1 Neural Style Transfer

In computer vision, style transfer is usually studied as a generalized texture synthesis problem, that is, to extract texture from source images and transfer it to targets [7–11]. Gatys et al. [12] were the first to study the use of CNN in reproducing a famous painting style on natural images. The groundbreaking work of their team demonstrated the power of CNN in representing texture modeling. Their results showed that CNN can extract content information from an arbitrary photograph and style information from a famous artwork. On the basis of this discovery, Gatys et al [12] proposed the use of feature activation of CNN to recombine the content of a given photo and the style of famous artworks. The main purpose of CNN is to extract style features, such as texture. However, its structure does not contribute much to synthesis. To address this problem, Champandard et al. [2] improved the network structure of CNN by using a composite image that was enhanced with semantic information during the generation phase. The aim was to narrow the gap between the generation model and the pixel level classification neural network. Recently, many arbitrary style transfer methods were proposed [5, 6, 27, 35].

2.2 Image-to-Image Translation

The purpose of image-to-image translation is to learn an image generation function that maps the input image in the source domain to the target domain: examples include sketch to portrait [3], image colorization [32, 34], and rain removal [23, 31]. Hertzmann et al. [17] proposed a single image pair non-parametric framework. Isola et al. [18] developed a universal framework called Pix2Pix, in conjunction with GANs [13]. Pix2Pix combines an L1 loss and an adversarial loss with paired data samples from two domains. This approach is driven by paired data, which are sometimes difficult to obtain. To overcome this limitation, Zhu et al. [36] designed a CycleGAN that can learn to translate images without pairs of ground truth. Choi et al. [4] proposed StarGAN using a single model to handle multi-domain translation; it utilizes a one-hot vector to specify the target domain. However, the extension to a new domain is still expensive. To address this issue, Liu et al. [21] presented a few-shot, unsupervised image-to-image translation algorithm that works on a previously unseen target domain. Based on these image-to-image translation methods, our work focuses on the text destylization problem.

2.3 Text style transfer

The research on text style transfer developed relatively late. Before 2017, studies in the field of text style mainly focused on the strokes of text fonts. HelpHanding [22] is used to study the authoring of strokes from six degrees of freedom by employing graphic methods. EasyFont [20], whice is mainly about handwriting font style transfer can transfer the users' handwriting font style to the specified text so that the text looks like the users' own handwriting. GlyphGAN [15] uses DCGAN [24] as the base model for font generation. Given that the same style vector is used, the resulting fonts tend to have the same style. Yang et al. [29] were the first to raise the issue of text effect transfer. A matching and synthesis method based on the relative positions of image patches on hieroglyphics was proposed. This method is susceptible to the difference in hieroglyphics and requires a large amount of computation. Mc-GAN [1] combines font transmission and text effect transmission by adopting two continuous subnetworks and trains them end-to-end by using the synthesized font data and the collected text effect data. An unsupervised artistic word generation algorithm [25] has also been developed; the algorithm is different from the supervised method that required a pixel-level aligned original text image as the guide. The unsupervised method can deal with arbitrary style images without the corresponding original texts. However, due to the lack of relevant data sets, only a few studies have been conducted on text effect style migration. Yang proposed TET-GAN [30] that uses the learning method of GAN to build a data-driven model, which can learn the accurate mapping relationship between glyphics and character effects from a large amount of data. Although some research has been performed on text style transfer, only a few studies focused on text destylization. A new framework of character destylization is proposed in this study in accordance with actual project needs.

3 METHOD AND TRAINING

3.1 Destylized Networks

Following the network architecture of GANs [13], our text destylization model is a combination of a generator G, a discriminator D and a auxiliary classifier. Adopted from CycleGAN [36], our model has a generator network composed of two convolutional layers with a stride size of two for downsampling, six residual blocks [16], and two transposed convolutional layers with a stride size of two for upsampling. We use PatchGAN [18] as the discriminator. The auxiliary classifier consists of a convolutional layer and a three-layer MLP and is used to categorize text content to help the network improve its handling of details. The network structure is shown in Figure 1. Given D_x, which is a styled text image with extra decorative elements, generator G learns to generate a fake raw text image $F_x = G(D_x)$. Discriminator D needs to distinguish whether the input is real or generated. We use instance normalization [26] for the generator but no normalization is applied for the discriminator. The loss function is a combination of WGAN-GP [14], L1 loss and CrossEntropy loss, as follows:

$$L = \lambda_{adv} L_{adv} + \lambda_{L1} L_1 + \lambda_{L_{au}} L_{au} \tag{1}$$

where.

$$L_1 = \| F_x - \widetilde{F}_x \|_1 \tag{2}$$

$$\begin{aligned} L_{adv} = & E_{\widetilde{F}_x}[D(\widetilde{F}_x, D_x)] \\ & - E_{F_x}[D(F_x, D_x)] \\ & + \lambda_{gp} E_{\widehat{F}_x}[(\| \nabla D(\widehat{F}_x, D_x) \|_2 - 1)^2] \end{aligned} \tag{3}$$

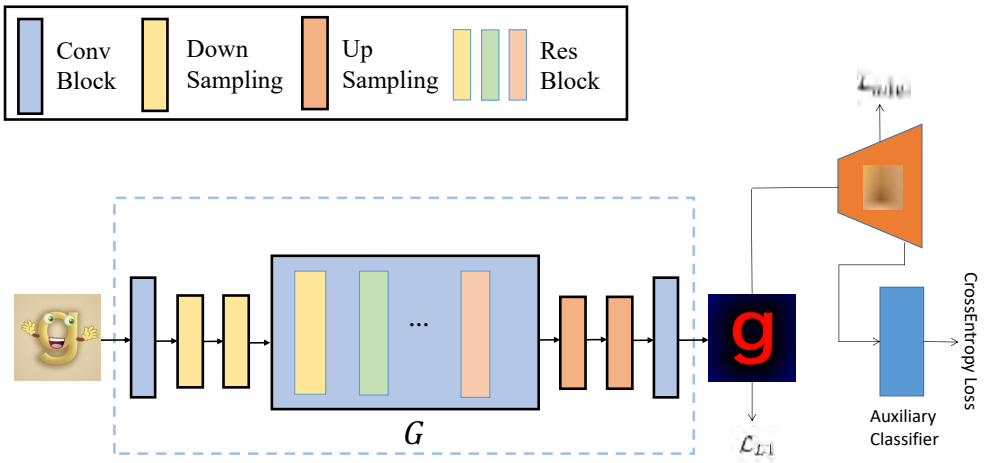

Figure 1: The network structure of our model.

$$L_{au} = -\sum_{i=1}^{C} y^{(i)} * log\, \widehat{y}^{(i)} \tag{4}$$

where. $\widetilde{F}_x$ is the ground truth, and $\widehat{F}_x$ is uniformly sampled along the straight lines between the sampling of F_x and $\widetilde{F}_x$. L_{au} is the cross-entropy loss. $\widehat{y}$ is the value that the output value of the auxiliary classifier is processed by Softmax. y is the true label, and C is the total number of categories.

3.2 Training

Our models are trained using the Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We also use a fixed learning rate of 2e-4 throughout the entire training stage. For all experiments, the resolution of images increases from 64, 128, up to 256, and layers are gradually added to the front and rear of the generator. To enhance the effect of network learning and fully utilize GPU and CPU resources, we use different batch sizes for images with different resolutions. When the resolutions of the images are 64*64, 128*128 and 256*256, the corresponding batch size are 200, 90 and 40, respectively. A progressive growing strategy [19] is used in the destylized network to stabilize the training process. The networks are trained on NVIDIA Tesla M40 GPU. We rearrange and combine an existing text style data set [28, 30], which contains about 60,000 images of Chinese characters and English letters, for our network training and testing. For the training of the auxiliary classifier, we define different Chinese characters and English letters(case sensitive) as one category respectively. The class labels use the form of one-shot. For all experiments, we set $\lambda_{gp} = 10$, $\lambda_{L1} = 100$, and $\lambda_{adv} = \lambda_{L_{au}} = 1$.

4 EXPERIMENTS AND RESULTS

We selected five the state-of-the-art models of image-to-image translation and text style transfer for comparisons. The five models are NST [12], Doodle [2], CycleGAN [36], Pix2pix [18], and TET-GAN [30]. We first present the generated results of our model together with those of the compared models. Second, we compare our model with the five other models through subjective and objective evaluation. Lastly, we show the destylized results of our model for different combinations of text and decorations and several complex words.

4.1 Qualitative analysis

We compare the proposed text effect transfer network with five state-of-art transfer methods in Figure 2. At present, no individual or team has conducted extensive research on text destylization, and comparative methods for reference are few. In view of this situation, we select the most advanced methods of image-to-image translation and text style transfer to verify the superiority of our model in the destylization of decorative text.

Image-to-image translation and text destylization are currently the two most relevant research directions for text destylization. Neural Style Transfer (NST) [12] and Neural Doodle [2] are image style transfer methods. NST uses CNNs to transfer the style of an image to another. Doodles [2] uses neural-based patch fusion and has a context-sensitive manner in the algorithm. However, NST and Doodles do not learn the relationship between text style and font, resulting in a fuzzy and confused texture structure. CycleGAN [36] and Pix2Pix [18] are image-to-image translation methods based on GAN, and they are all re-trained on our dataset. The inputs of Pix2Pix [18] and CycleGAN [36] are revised to be the same as our input. CycleGAN only captures some of the texture features of the style and font. Thus, it fails in text destylization. Pix2Pix produces irregular textures and fonts. Benefiting from the progressive growing strategy [19] and the WGAN-GP [14], our model is more stable than Pix2Pix. Given that TET-GAN [30] involves the migration and removal of the basic style without considering decorative elements, the destylization of decorative text could fail because decorative elements block the font. From these comparisons, we could conclude that our model can not only reconstruct the details of text content, but also eliminate the influence of decorative elements.

In addition to these comparative experiments, we present the results of our model for more complex cases, as follows:

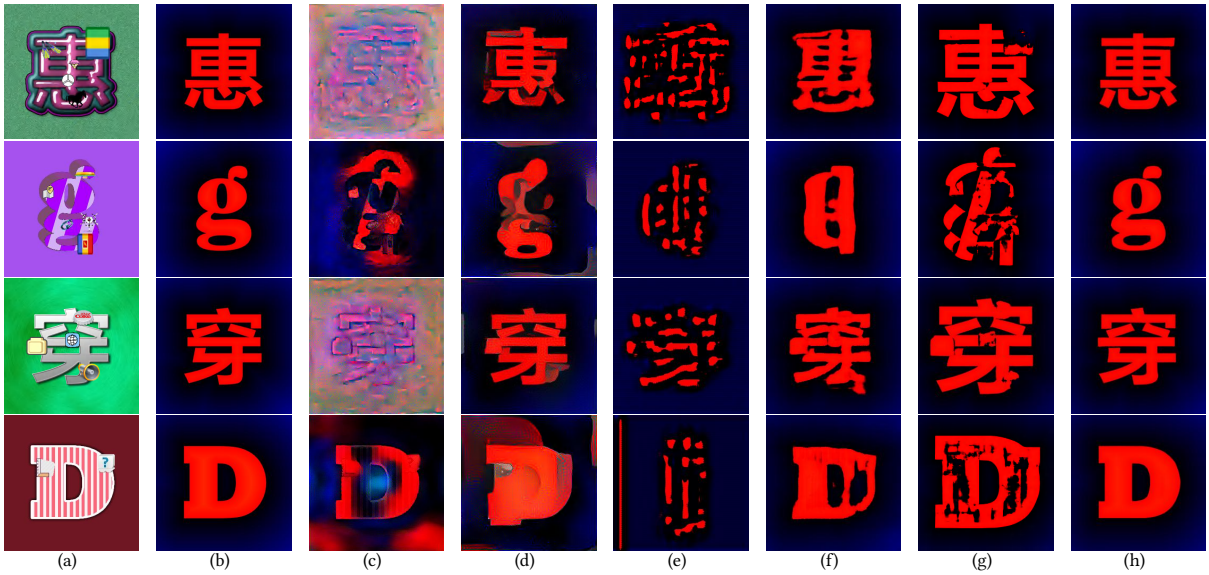

Figure 2: Comparison with current more advanced methods. The first column is the style image with decorative elements that the model inputs. The second column is the corresponding ground truth. The third column is the result of NST [12]. The fourth column is the result of the Doodle [2]. The fifth column is the result of CycleGAN [36]. The sixth column is the result of Pix2Pix [18]. The seventh column is the result of TET-GAN [30]. The last column is the result of our model.

- Same word, different fonts, same decoration. In this case, we use different fonts with the same decoration for the same word. We demonstrate that our model can remove decorations for different fonts, and can be unaffected by font changes. See Figure 3.
- Word combinations. In addition to dealing with individual Chinese characters and English letters, our model can also deal with simple vocabulary. See Figure 4.
- Same word, different decorations. We use different decorative elements to form different decorative styles for words with the same background, to demonstrate the capability of our model to remove decorative elements. As shown in Figure 5, both Chinese characters and English letters are included.

4.2 Subjective and Objective Evaluations

We conduct a user survey and objective index calculation to prove the superiority of our model over other models comprehensively and objectively.

User Study. From the results of the qualitative experiments, we randomly selected 100 groups of different text destylized renderings to make the questionnaire. We receive 40 responses, among which 37 responses are vaild. A total of 3,700 votes are obtained. Our model has 3276 votes, NST has 126 votes, Doodle has 48 votes, CycleGAN has 6 votes, Pix2Pix has 73 votes, and TET-GAN has 171 votes. On the basis of the feedback results, we compute the vote statistics and perform result comparisons, as shown in the Table 1.

Objective Indicators. We selected four metrics, namely, FID, mIoU, RMSE and PSNR. FID is the mean value between the ground truth and the generated value after extracting the feature vector, and evaluating the distance of the covariance. The closer the generated results are to the truth features, the smaller the square of the mean difference is, the smaller their covariance is, and the smaller the

Figure 3: Same word, different font, same decoration. The first and third lines have the same word, and four different font styles can be seen. The second and fourth rows are the destylized results of our model.

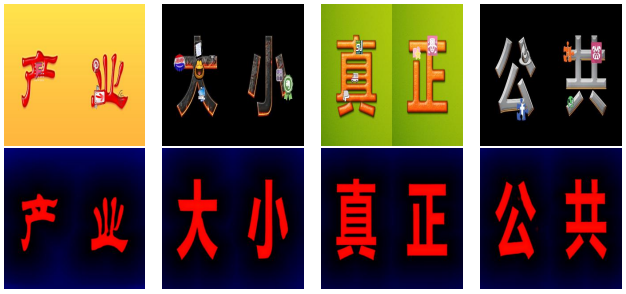

Figure 4: Word combinations. Destylized results of our model for complex words.

Figure 5: Same word, different decorations. The first and third lines respectively show two Chinese characters and two English letters have three different styles of decorative elements. The second and fourth rows are the destylized results of our model.

Table 1: Our user study results. The first and second column are respectively the number of votes and the winning rate of each method.

Method	Votes	Win Rate
NST	126	3.41%
Doodle	48	1.30%
CycleGAN	6	0.16%
Pix2pix	73	1.97%
TET-GAN	171	4.62%
Ours	**3276**	**88.54%**

Table 2: Model performance evaluation index.

Method	FID	mIoU	RMSE	PSNR
NST	285.8983	0.1416	0.0426	30.8222
Doodle	239.7642	0.2290	0.0408	30.6476
CycleGAN	250.3068	0.1197	0.0419	36.3651
Pix2pix	150.5688	0.5452	0.0422	38.2636
TET-GAN	83.5101	0.5677	0.0386	39.7569
Ours	**21.0336**	**0.8746**	**0.0336**	**42.4457**

Table 3: U-net refers to the network generator using U-NET. Resnet-4, Resnet-6 and Resnet-8 respectively refer to a ResNet network that uses four, six and eight residual blocks. Au refers to the addition of auxiliary classifiers to the network.

Method	FID	mIoU	RMSE	PSNR
U-Net	40.7418	0.6379	0.0371	40.4102
ResNet-4	108.7667	0.5169	0.0427	37.7571
ResNet-8	95.5848	0.6543	0.0401	37.2107
ResNet-6	38.8962	0.8659	0.0340	40.8679
ResNet-6+Au	**23.6686**	**0.8993**	**0.0299**	**42.7519**

sum of FID is. MIoU is the ratio of the intersection and union of two sets of ground truth and generated values. The larger the mIoU is, the more the intersection is and the closer the generated value is to the ground truth. RMSE measures the root-mean-square error between the generated value and ground truth. The smaller RMSE is, the closer the generated value is to the ground truth. PSNR is the peak signal-to-noise ratio of ground truth and the generated value. The larger the PSNR is, the better the generated results are. The performance of our model in comparison with that of the five other models is shown in Table 2.

As can be seen from the experimental results in the table, our model is superior to the five other models in four aspects in terms of the decorative text destylization task.

4.3 Ablation Study

The architecture of our model comprises a generator, a discriminator and an auxiliary classifier. In this section, we discuss the influence of each part of our model. We verify the effect of different model structures from three aspects: the influence of different generators, number of residual blocks, and branches of auxiliary classifier on the effect of the model.

U-Net and ResNet. We used U-Net and ResNet as model generators to perform experiments. The objective indicators of the experimental results between U-Net and ResNet-6 are compared in Table 3. Additional experimental results are provided in Supplementary Materials.

Auxiliary classifier. We verify the role of the auxiliary classifier by comparing the addition and non-addition of the auxiliary classifier to the model. Table 3 compares the objective indicators of the experimental results of ResNet-6 and ResNet-6+Au. From the results, we can see that the helper classifier can be added to help the model to learn additional details on the content. Other experimental results are provided in Supplementary Materials.

Number of residual blocks. Our model uses ResNet as a generator. We set the number of residual blocks as 4, 6 and 8 in our ablation study to verify the impact of the number of residual blocks on the network. Table 3 compares the objective indicators of the experimental results of ResNet-4, ResNet-8 and ResNet-6. Additional experimental results are provided in Supplementary Materials.

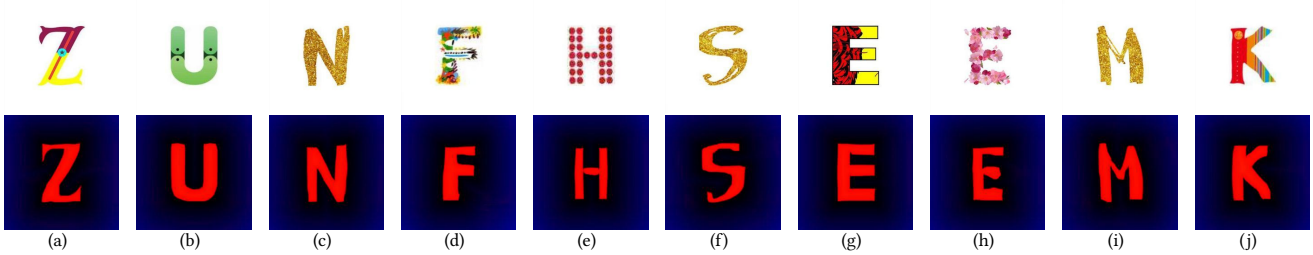

Figure 6: The first line is the style images in the wild. The second line is the processing results of our model.

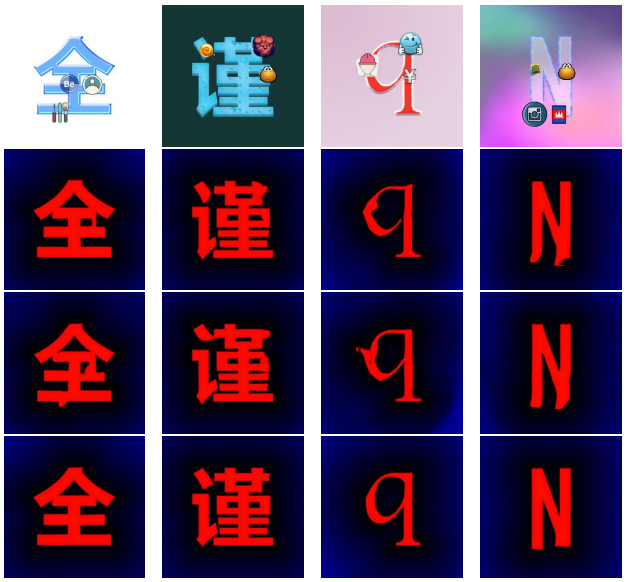

Figure 7: The first row is a text-style image. The second row is the result of the C64-C128 discriminator experiment. The third row is the result of the C64-C128-C256-C512-C512-C512 discriminator experiment. The fourth row is the result of the C64-C128-C256-C512 discriminator experiment.

Single-scale and multi-scale training. In the training process, multi-scale training is adopted to cut out patches of 64, 128, and 256 and send them into the model. A single-scale comparison experiment is conducted with patch 256 to prove the effectiveness of multi-scale training. The objective indicators of the different scale training methods are compared in Table 4. Additional experimental results are provided in Supplementary Materials.

Different Discrimators. In our model, the structure of the discriminator is C64-C128-C256-C512. In order to compare the effects of different discriminators on the model, ablation experiments of three different discrimators were carried out. The other two discriminators have the following structure: C64-C128 and C64-C128-C256-C512-C512-C512. All other discriminators follow the same basic architecture, with depth varied to modify the receptive field size. The experimental results are shown in Figure 7.

Table 4: The first and second lines respectively show the results of the single-scale training and the multi-scale training.

Method	FID	mIoU	RMSE	PSNR
single scale	95.7025	0.2441	0.0405	38.0133
multiple scale	**23.6686**	**0.8993**	**0.0299**	**42.7519**

Table 5: Discrimator1 is C64-C128. Discrimator2 is C64-C128-C256-C512-C512-C512.

Method	FID	mIoU	RMSE	PSNR
Discrimator1	55.0458	0.8244	0.0375	42.2382
Discrimator2	90.7623	0.7737	0.0396	40.7921
Ours	**23.6686**	**0.8993**	**0.0299**	**42.7519**

4.4 Unseen Styles

We also collected 1K artistic text of various text effects from the Internet. These styled text effects in the wild are used to verify the generalization of our model. The experimental results are shown in Figure 6. As can be seen from the results, our model performs well in the unseen style data.

5 CONCLUSION

In this study, we address the problem of destylization of decorative text and propose a novel framework for the text destylization. Our network combines residual block and PatchGAN. At the same time, we demonstrate the advantages of our model in the destylization of decorative text from three aspects, namely, comparative experiment, user study and performance evaluation. Finally, we show the results of the model for some more complex cases, and show more of the application of the model.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China under no. 2020AAA0106200, and by National Natural Science Foundation of China under nos. 61832016, U20B2070 and 61672520.

REFERENCES

[1] S. Azadi, M. Fisher, V. Kim, Z. Wang, E. Shechtman, and T. Darrell. 2018. Multi-content GAN for Few-Shot Font Style Transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7564–7573.

[2] Alex J Champandard. 2016. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768* (2016).

[3] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. 2009. Sketch2photo: Internet image montage. *ACM Transactions on Graphics* 28, 5 (2009), 1–10.

[4] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8789–8797.

[5] Yingying Deng, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, and Changsheng Xu. 2021. Arbitrary Video Style Transfer via Multi-Channel Correlation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*.

[6] Yingying Deng, Fan Tang, Weiming Dong, Wen Sun, Feiyue Huang, and Changsheng Xu. 2020. Arbitrary Style Transfer via Multi-Adaptation Network. In *Proceedings of the 28th ACM International Conference on Multimedia* (Seattle, WA, USA). Association for Computing Machinery, New York, NY, USA, 2719–2727.

[7] Lars Doyle, Forest Anderson, Ehren Choy, and David Mould. 2019. Automated pebble mosaic stylization of images. *Computational Visual Media* 5, 1 (2019), 33–44.

[8] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. 2003. Example-based style synthesis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2. IEEE, II–143.

[9] Alexei A Efros and William T Freeman. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 341–346.

[10] Michael Elad and Peyman Milanfar. 2017. Style transfer via texture synthesis. *IEEE Transactions on Image Processing* 26, 5 (2017), 2338–2351.

[11] Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier. 2016. Split and match: Example-based adaptive patch sampling for unsupervised style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 553–561.

[12] L. A. Gatys, A. S. Ecker, and M. Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2414–2423.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*. 2672–2680.

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems (NIPS)*. 5767–5777.

[15] Hideaki Hayashi, Kohtaro Abe, and Seiichi Uchida. [n.d.]. GlyphGAN: Style-Consistent Font Generation Based on Generative Adversarial Networks. 186 ([n. d.]).

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[17] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. 2001. Image analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. 327–340.

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1125–1134.

[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive Growing of GANs for Improved Quality, Stability, and Variation.

[20] Zhouhui Lian, Bo Zhao, Xudong Chen, and Jianguo Xiao. 2019. EasyFont: A Style Learning-Based System to Easily Build Your Large-Scale Handwriting Fonts. *ACM Transactions on Graphics* 38, 1 (2019), 6:1–6:18.

[21] Ming Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. 2019. Few-Shot Unsupervised Image-to-Image Translation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. 10550–10559.

[22] Jingwan Lu, Fisher Yu, Adam Finkelstein, and Stephen DiVerdi. 2012. HelpingHand: Example-Based Stroke Stylization. *ACM Transactions on Graphics* 31, 4, Article 46 (July 2012), 10 pages.

[23] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. 2018. Attentive generative adversarial network for raindrop removal from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2482–2491.

[24] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*.

[25] Shuai, Yang, Jiaying, Liu, Wenhan, Zongming, and Guo. 2018. Context-Aware Text-Based Binary Image Stylization and Synthesis. *IEEE Transactions on Image Processing* 28, 2 (Feb. 2018), 952–964.

[26] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022* (2016).

[27] H. Wang, Y. Li, Y. Wang, H. Hu, and M. H. Yang. 2020. Collaborative Distillation for Ultra-Resolution Universal Style Transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1857–1866.

[28] Wenjing Wang, Jiaying Liu, Shuai Yang, and Zongming Guo. 2019. Typography with Decor: Intelligent text style transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5889–5897.

[29] Shuai Yang, Jiaying Liu, Zhouhui Lian, and Zongming Guo. 2017. Awesome Typography: Statistics-Based Text Effects Transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2886–2895.

[30] Shuai Yang, Jiaying Liu, Wenjing Wang, and Zongming Guo. 2019. TET-GAN: Text Effects Transfer via Stylization and Destylization. In *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*. 1238–1245.

[31] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. 2017. Deep joint rain detection and removal from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1357–1366.

[32] Richard Zhang, Phillip Isola, and Alexei A. Efros. 2016. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*. Springer, 649–666.

[33] Rui Zhang, Mingkun Yang, Xiang Bai, Baoguang Shi, and Minghui Liao. 2019. ICDAR 2019 Robust Reading Challenge on Reading Chinese Text on Signboard. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*.

[34] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S. Lin, Tianhe Yu, and Alexei A. Efros. 2017. Real-Time User-Guided Image Colorization with Learned Deep Priors. *ACM Transactions on Graphics* 36, 4, Article 119 (July 2017), 11 pages.

[35] Y. Zhang, C. Fang, Y. Wang, Z. Wang, Z. Lin, Y. Fu, and J. Yang. 2019. Multimodal Style Transfer via Graph Cuts. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. 5942–5950.

[36] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*. 2223–2232.