Intelligent Typography: Artistic Text Style Transfer for Complex Texture and Structure

Wendong Mao, *Graduate Student Member, IEEE*, Shuai Yang, *Member, IEEE*, Huihong Shi, Jiaying Liu, *Senior Member, IEEE*, and Zhongfeng Wang, *Fellow, IEEE*

***Abstract*—Text style transfer is an important task to render artistic texts from a reference image or style, and is widely desired in many visual creations. Previous works have brought some efficient methods for text style transfer, which facilitate users to design various artistic texts automatically. However, these works mainly focus on relatively simple text effects, and do not perform well on complex reference styles. In this paper, we propose a coarse-to-fine framework to generate exquisite texts with complex texture and structure in an unsupervised way, achieving real-time control of style scales (i.e., text stylistic degree or deformation degree). The key idea is to decouple the overall task into two steps, prototype generation and detail refinement, and explore delicate networks for each step to imitate the features at different levels. Based on this idea, in the first step, we present a novel pro-gen GAN to generate prototypes of artistic texts using the reference style, and develop a deformable module to empower the pro-gen GAN to continuously characterize the multi-scale shape features without network retraining. Furthermore, we propose a mix-attention training scheme for text style transfer, which can avoid artifacts and retain a clear text background. In the second step, we introduce two optimized networks for detail refinements. Experimental results show that the proposed method can synthesize exquisite stylized texts with complex reference styles, and surpass the state of the arts in texture reconstruction, contour imitation, and text image quality drastically.**

***Index Terms*—Complex reference style, controllable style transfer, generative adversarial network, text style transfer, unsupervised learning.**

I. Introduction

NOWADAYS, style transfer has developed rapidly and attracted widespread attention in many different fields [1], [2], [3], [4]. Text style transfer, an important sub-topic of image style transfer, is widely desired in many visual tasks, such as poster, web page and advertisement design. Such visual tasks aim to extract reference styles from either existing well-designed artistic texts or general free-form style images, and apply them to the target raw texts for artistic text design. The reference styles include colors, shapes, text shadows and textures. These exquisite text effects make the posters or web pages visually attractive, and provide more artistic merit and creativity.

To transfer well-designed text effects to new target texts, Yang et al. [5] first designed a novel algorithm to exploit the effect analytics for text synthesis in 2017. In 2018, a bidirectional flexible TET-GAN [6] was proposed for text stylization and de-stylization, which can extend new text styles and generate high-quality stylized texts automatically. Then, FET-GAN [7] was developed to implement font variation among multiple text effects domains by a few-shot fine-tuning strategy, transferring the style of a pre-trained model to new effects. Interestingly, Wang et al. [8] paid attention to decorative elements. They designed a robust framework to separate, transfer, and recombine decors and basal text effects, obtaining new stylized texts with exquisite decors. These methods require specialized text datasets for text style transfer, but text effects in the existing text dataset [5], [8], [9] are relatively simple, as shown in Fig. 2(a), which is hard to be applied for complex text effect transfer.

For general free-form style images, Atarsaikhan et al. [10] directly applied the method in [11] to text style transfer and achieved visually fuzzy results. Then, Yang et al. investigated a new problem of fast controllable text style transfer, and developed bidirectional shape-matching [12] and shape-matching++ [13] frameworks for multi-scale artistic text synthesis, which can imitate the visual features from general free-form style images to obtain stylized texts. Furthermore, Yang et al. [14] explored the stylization of text-based binary images for automatic artistic typography creation. It can combine the stylized geometric shape with a background image, achieving visually appealing images. However, those methods are developed for relatively simple reference styles, and directly using them for complex text styles leads to dissatisfactory results.

In view of the above, previous works mainly focus on relatively simple text effects to generate images with limited styles. In addition, real-time control for the stylistic degree of the glyph can only be realized on the simple structure-free styles, like fire and water in [12], [13], but real-time response and adjustment for complex styles is equally crucial. To the best of our knowledge,

Manuscript received 26 August 2021; revised 2 March 2022, 9 May 2022, and 18 July 2022; accepted 14 September 2022. Date of publication 30 September 2022; date of current version 1 November 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62174084 and 62104097, in part by the High-Level Personnel Project of Jiangsu Province under Grant JSSCBS20210034, and in part by the Key Research Plan of Jiangsu Province of China under Grant BE2019003-4. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. Catherine Zhao. *(Corresponding authors: Jiaying Liu; Zhongfeng Wang.)*

Wendong Mao, Huihong Shi, and Zhongfeng Wang are with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210008, China (e-mail: wdmao@smail.nju.edu.cn; shihh@smail.nju.edu.cn; zfwang@nju.edu.cn).

Shuai Yang and Jiaying Liu are with the Wangxuan Institute of Computer Technology, Peking University, Beijing 100080, China (e-mail: williamyang@pku.edu.cn; liujiaying@pku.edu.cn).

The code is available on the page: https://github.com/WendongMao/Intelligent_Typography.

This article has supplementary downloadable material available at https://doi.org/10.1109/TMM.2022.3209870, provided by the authors.

Digital Object Identifier 10.1109/TMM.2022.3209870

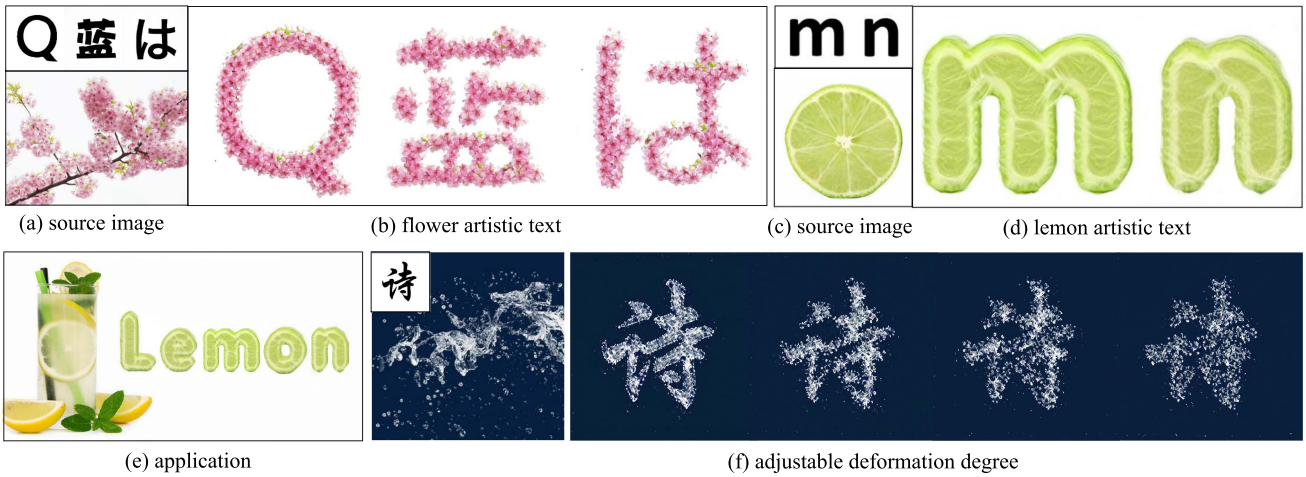


Fig. 1. Generated diverse artistic texts via the proposed framework. Our framework can tackle complex reference styles, allowing users to generate artistic texts (b)(d) via binary text masks and style images (a)(c). The artistic texts can be applied to exquisite typography (e). Moreover, the deformation degree can be adjusted to obtain multi-scale stylized texts (f).

there has been no work discussing the transformation and application for **the style effects with relatively complex texture and structure**. The challenges of complex text style transfer lie in three aspects. On one hand, for transferring complex text effects or general free-form style images, the existing simple text datasets can not be utilized directly, and it is expensive and even unrealistic to develop a specialized large-scale text dataset with high resolution images and diversified style components. On the other hand, complex reference styles contain variable structure and texture characteristics, which is difficult to be extracted and imitated. Further, the complex features are sensitive and are easy to be twisted by previous style scale-controllable methods. Thus, the complex but exquisite style components, which are commonly used in art design, are hard to be transferred into targeted texts via previous methods.

To tackle this problem, in this paper, we pay attention to the style images with complex texture and structure. Fig. 1 shows some examples of our framework, where the complex styles are transferred to texts and real-time control of style scales can be achieved. The key idea is a coarse-to-fine framework to transfer style at different levels, where delicate networks are developed for feature imitation of each level. For lack of text dataset with complex effects, an one-shot unsupervised method is developed to generate artistic texts with one style image. To imitate complex characteristics, we skillfully propose a $2\times$ magnification network, and convert the challenging style transfer task into texture expanding and contours recovering issues, achieving a high-resolution output meanwhile.

The coarse-to-fine framework decomposes the overall task into two sub-tasks, **prototype generation** and **detail refinement**. Three delicate networks are developed to accomplish those sub-tasks in order. First, the prototype generation is performed by a $2\times$ magnification network, pro-gen GAN, so that a coarse-level prototype is obtained. To control the stylistic degree, we develop a deformable module to build multi-shape text masks so that multi-scale stylized texts can be obtained. In addition, to consider both edge and inner characteristics, a mix-attention scheme is proposed to eliminate artifacts in key foreground as well as to keep clear background. Second, a structure network is developed to refine structure features of the text prototype in fine-level. To enhance the adaptability of the network, we exploit the general image dataset as content images so that the features can be applied into various text prototypes. Third, based on the structure refinement results, texture details are refined by the style optimization over shallow layers of VGG network, where a background-fixed strategy is developed to make the background cleaner.

In summary, the contributions of this work are three-fold:

- We raise a coarse-to-fine framework to generate artistic texts based on free-form style images. Our one-shot framework can synthesize complex text effects based on a single style image in an unsupervised way, without requiring large-scale special artistic text datasets.
- We present a pro-gen GAN to generate the stylized text prototype, which exploits a $2\times$ magnified network to learn complex style elements intelligently. The network provides more space for structure reconstruction and texture transfer, and can achieve high-resolution results. In terms of controllable text style transfer, a deformation module is developed for glyph deformation without network retraining.
- We propose a mix-attention training scheme to learn style features, which can retain both edge and inner characteristics. With the training scheme, the framework can generate exquisite images with vivid details and clear background.

The rest of this paper is organized as follows. In Section II, we review related works in style transfer and artistic text synthesis. Section III describes the proposed framework to tackle the text effects with complex texture and structure, and illustrates the detailed architecture of each module in the overall framework. In Section IV, we validate our method by conducting extensive experiments, and make comparison with state-of-the-art methods for text style transfer and multi-scale style control. Finally, we conclude the work in Section V.

II. Related Work

A. Image Style Transfer

The task of image style transfer is combining the style feature of a style image and the content information of a content image to obtain a stylized image. Gatys et al. [11] pioneered on the neural style transfer (NST) by exploiting the powerful representativeness of deep neural networks, where the style feature of an image is computed by Gram matrix [15], [16]. Then, a fast feed-forward style network is proposed by Johnson et al. [17]. It combine the benefits of per-pixel loss and the style loss based on Gram matrix, improving the processing speed of network significantly.

By leveraging the powerful generation ability of GAN, a series of GAN-based architectures are raised for various style transfer tasks. For example, a style-aware content loss was proposed in [18] for real-time and high-resolution stylization of images/videos, which makes the GAN model better captures the subtle nature features. In [19], a new method was presented to capture the particularities of style and separates style and content. In addition, [20] introduced a content transformation module between the encoder and decoder to realize content-and style-aware stylization.

The two kinds of approaches have different emphasis, where Gram-based method is good at texture representation, and GAN-based method can better capture local information and generate new similar elements. Directly applying image style transfer methods for artistic text synthesis leads to obvious artifacts and unclear background, while GAN-based methods need large-scale style image datasets for training. Thus, specialized frameworks are required for efficient one-shot text stylization. Combined with the Gram-based and GAN-based methods, our framework, targeting for text style transfer, can yield more artistically vivid results.

B. Artistic Text Synthesis

The task aims to render texts with the style of a reference image, which is first explored by Yang et al. [5] in 2017. The authors exploited a non-parametric method to analyze features and derive their correlation for text synthesis. Moreover, Yang et al. [21] explored the problem of fantastic special-effects synthesis for the typography, and developed distribution-aware method for various text effects. Then, UT-Effect [14] was proposed to imitate visual features from more general free-form style images, which expands the task for icon/symbol rendering and image inpainting.

Some methods based on deep neural networks, such as MC-GAN [22], TET-GAN [6], Decor [8], FET-GAN [7], and AGIS-Net [23], were proposed to generate stylized texts automatically. To stylize target texts from only a few referenced samples, MC-GAN [22] proposed an end-to-end GAN model to generate a set of multi-content images. Similarly, Zhu et al. [24] proposed a novel approach by decoding weighted deep features and calculating the similarity scores of the target texts, which can transfer the style of given styles to the contents of unseen texts. To generate exquisite artistic texts, Decor [8] defined a new problem of text style transfer with decorative elements, and proposed a framework to separate and recombine the basal text effects and the decorative elements, which can adapt to different styles and glyphs. TET-GAN [6] proposed a bidirectional framework for text stylization and de-stylization, achieving high-quality stylized texts. These works need special text effect datasets for network training as shown in Fig. 2(a), and fail on generating complex text effects from free-form styles, such as "lemon" and "flower". For free-form style images as reference, Yang et al. [12] introduced a new problem of real-time control of glyph deformations, and proposed a novel bidirectional shape-matching framework to generate multi-scale stylized texts. Furthermore, Yang et al. [13] introduced a scale-aware Shape-Matching GAN++ to animate a static text image based on the reference style video, generating high-quality and controllable artistic texts. Nevertheless, these works mainly focus on relatively simple text effects or style images as shown in Fig. 2(b) and (c), and can not synthesize artistic texts with complex reference styles. Our work explores a coarse-to-fine framework to tackle the issue, which is more robust for various text styles, achieving satisfactory visual results.

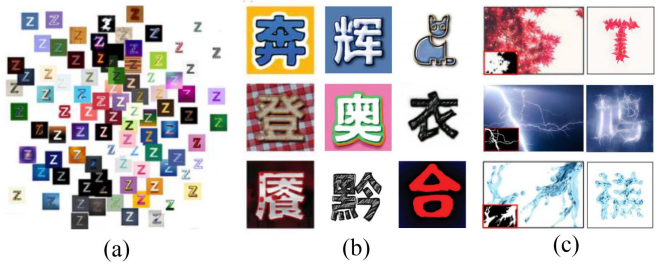


Fig. 2. (a) Images in the existing text datasets [9]. (b) Examples of transferring well-designed text effects to texts [7]. (c) Examples of transferring general free-form style images to texts [12] and [13].

C. Controllable Text Style Transfer

The controllable style transfer is first mentioned by Gatys et al. [11] and the stylistic degree is controlled by changing the weights of the content loss and style loss. Obviously, this kind of methods need to re-train the model for different weights. Further, Jing et al. [25] developed a stoke-controllable network for fast style transfer, which can adjust receptive fields to obtain different style-scale outputs. For artistic text design, Yang et al. [12] proposed a scale-controllable module to adjust the stylistic degree of the glyph, which needs to train the network only once. In this work, we raise a deformable module that exploits a simple but efficient scheme to adjust glyph deformation without network re-training. It can achieve controllable complex style transfer with shaper and more natural styles than [12].

III. Method

The proposed coarse-to-fine framework is shown in Fig. 3, which decomposes the style transfer task into two sub-tasks:

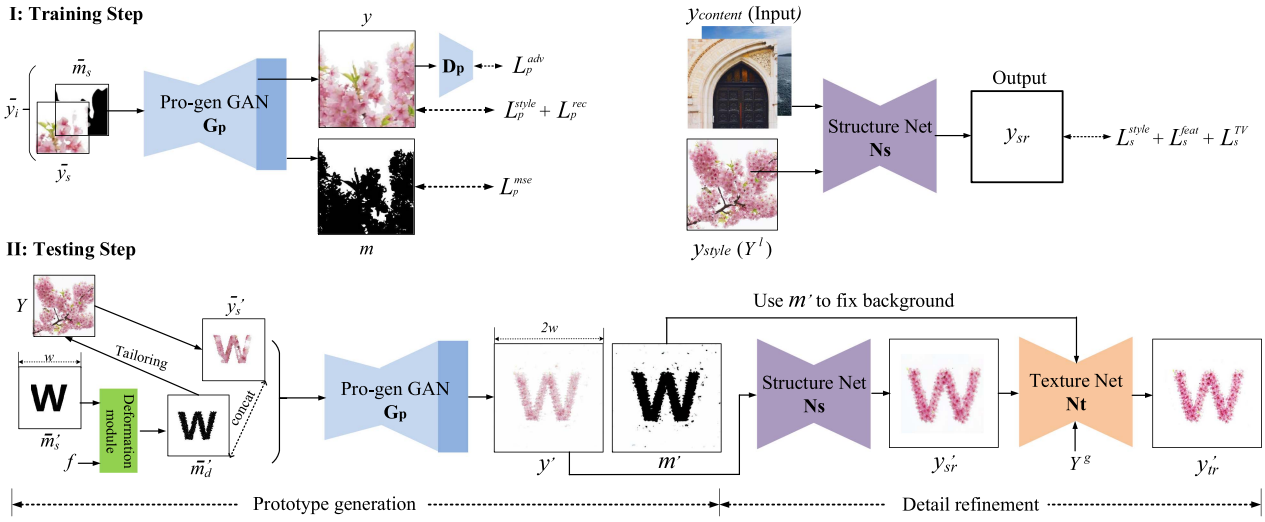


Fig. 3. Overview of the proposed coarse-to-fine framework. (I) The training step of the framework; (II) The testing step of the framework.

1) prototype generation and 2) detail refinement. The prototype generation is modeled by a novel GAN model, namely pro-gen GAN (G_p), which generates a coarse-level stylized prototype from the given mask and tailored texture. The second sub-task is performed by a structure network (N_s) and a texture network (N_t), which refine the structure and texture features in fine-level, respectively. Our multi-stage framework can extract style features at multiple levels by the three delicate networks, so that each network can focus on a specific level to accomplish the overall transferring task in order. Due to the lack of text effect datasets with paired data, the proposed framework develops an one-shot learning method for the unsupervised training. For the training step, we preprocess the style image (Y) and binary mask (M) to generate paired data, and then use them to backwards update G_p. Afterwards, N_s is trained with the general image dataset to learn the ability of structure refinement. In the testing step, a text mask is given to the deformable module to adjust the style scale, obtaining the tailored mask ($\bar{m}_d'$) and style image ($\bar{y}_s'$), and then they are sent into G_p, N_s, and N_t, to perform forward style transfer.

In the following, we will first introduce the input preprocessing for generating the training data. Then, we will present the backward prototype generation process, containing the architecture of the proposed pro-gen GAN. The backward structure refinement will be introduced in Section III-C. Finally, the testing step for generating artistic texts will be described in Section III-D, including a deformable module that enables the network to achieve multi-scale glyph deformations.

A. Input Preprocessing

Because the complex reference styles contain rich texture and structure features, they are hard to be transferred by the simple input guidances. For example, [8], [12], [13] only exploited text masks to generate text distribution guidances, and those guidances are hard to help models map complex style elements into targeted texts. Hence, a delicate input preprocessing method is developed to provide more useful information for feature mapping and texture guiding. Since there is no specialized text effect dataset for network training, the input preprocessing is to provide paired training data from a form-free style image for the one-shot learning. The paired training data, $([\bar{y}_s; \bar{m}_s], [y; m])$, aim to imitate the characteristic of input/output texts in the forward text style transfer process, so that G_p can learn the text stylization from those paired data. The preprocessing is described in Fig. 4, which can be listed as follows:

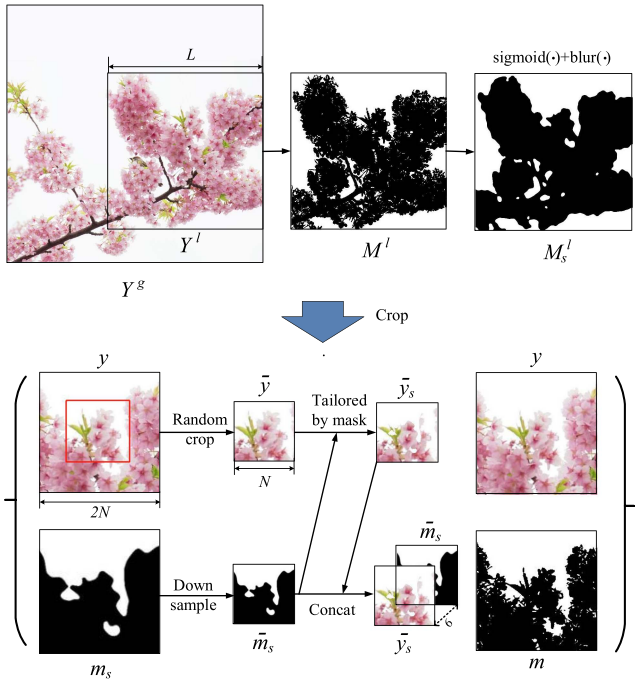


Fig. 4. The illustration of input data preprocessing, where the generated input data is $[\bar{y}_s; \bar{m}_s]$, and the ground truth is $[y; m]$.

Step 1: To obtain the contour characteristic of style elements, we extract the binary mask M^g for the raw style image Y^g,

where the style component is in black color and background is in white color. Further, depending on the area of black color, we obtain an $L \times L$ local image with the smallest background, namely Y^l, and the mask of Y^l, represented as M^l. The purpose of the step is to get global and local images for mix-attention training.

Step 2: To imitate the smoothness of raw text mask edges, Gaussian scale-space representation [26], [27] is applied to simplify the edges of M^g and M^l, obtaining the smooth masks M_s^g and M_s^l. The representation method mainly explores Gaussian blur and the $sigmoid(\cdot)$ function for edge simplification.

Step 3: To create a training set from a single style image, $2\,N \times 2\,N$ local patches are cropped from Y^g or Y^l in a predefined probability, where $2\,N < L$ (see mix-attention training scheme for details). Similarly, $2\,N \times 2\,N$ patches in the corresponding positions of M^g(or M^l) and M_s^g(or M_s^l) are cropped. The local patch of Y^g(or Y^l) is denoted by y, the patch in M^g(or M^l) is represented by m, and the patch in M_s^g(or M_s^l) is represented by m_s. y has the actual style characteristics and m can describe the real contours of style elements, so we concatenate y and m to obtain the six-channel ground truth $[y; m]$ for the training set.

Step 4: G_p can generate stylized prototypes and perform $2\times$ super-resolution reconstruction to enhance the robustness of feature learning. To generate the paired training data for G_p, the input size in training set should be halved compared with ground truth $[y; m]$. Hence, we down-sample m_s to get an $N \times N$ mask $\bar{m}_s$. Different from m_s, we randomly crop an $N \times N$ patch in y to obtain the small image patch $\bar{y}$ instead of down-sampling. Then, the $N \times N$ patch $\bar{y}$ is tailored by the smooth mask $\bar{m}_s$, obtaining an $N \times N$ tailored style patch $\bar{y}_s$. Due to the simplification of mask edge, $\bar{y}_s$ and $\bar{m}_s$ have smooth contours, and the characteristic is similar to the raw text mask $\bar{m}'_s$ and its tailored style patch $\bar{y}'_s$, where the superscript $'$ represent the input or output data in the testing phase. Therefore, $\bar{y}_s$ and $\bar{m}_s$ are merged as six-channel input data $\bar{y}_i = [\bar{y}_s; \bar{m}_s]$ of the training set.

Mix-attention training scheme: Large image, Y^g, usually contains more global information such as contour features and background, while the local image, Y^l, pays more attention to the key style elements. Using local patches from only a single image (Y^g or Y^l) often fails on learning both global shapes and local style details thoroughly, as shown in Fig. 14(a) and Fig. 14(b). To tackle this problem, we propose a mix-attention training scheme as shown in Fig. 5, where "mix" means that we mix Y^g and Y^l, and then crop patches from them with a pre-defined probability to obtain the training data, and "attention" represents that Y^l has more probability ($i.e.$, 0.9) to be chosen so that the texture and structure of key foreground can be learned adequately. As described in step 1 of input preprocessing, we firstly search an $L \times L$ local mask with the smallest background within all the $L \times L$ patches in M^g, namely M^l. Secondly, the local style image Y^l and its corresponding smooth mask M_s^l, are determined depending on the position of M^l. Then, we choose $Y^g/M^g/M_s^g$ or $Y^l/M^l/M_s^l$ in a probability $[a : 1-a]$ to perform step 3 and 4, obtaining the paired patches to train G_p. The scheme can learn

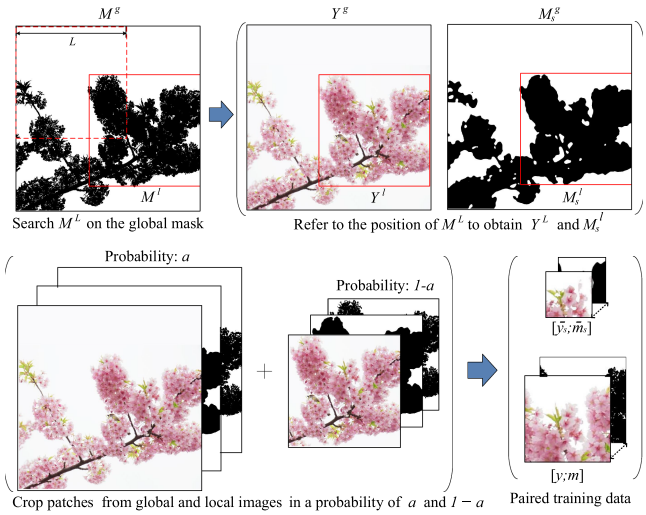


Fig. 5. The illustruation of the mix-attention training scheme.

style characteristics at both global and local levels, achieving boosted overall effects and texture details.

Compared with previous methods [8], [12], [13], the paired patch $(\bar{m}_s, m)$ provides explicit overall contour for shape mapping, and the paired patch $(\bar{y}_s, y)$ gives stronger guidances for texture and structure features, which provide more information for network training. In addition, the images Y^g and Y^l will be used in the following networks N_s and N_t to transfer the style features at different levels.

B. Backward Prototype Generation (G_p)

For complex reference styles, it is hard for $1\times$ networks to learning an robust pixel-to-pixel level relationship, since the network will easily over-fit the single style image. Hence, we develop a novel $2\times$ magnification network, pro-gen GAN, to generate coarse-level stylized prototypes. It smartly converts the problem of complex style imitation into the issue related to texture expansion and super-resolution, which relieves the pressure of one-shot learning dramatically.

Fig. 6 shows the architecture of the pro-gen GAN's generator, which contains two parts, namely G_{p1} and G_{p2}. G_{p1} is to generate the $2\times$ magnification stylized image, and G_{p2} is a segmentation module to extract the mask of the $2\times$ magnification stylized image. As shown in Fig. 3-I, G_{p1} is represented in light blue color, whose training exploits the discriminator D_p. G_{p2} is denoted in dark blue in Fig. 3-I, and it does not require the collaboration of D_p for network training.

G_{p1} is composed of several convolutional layers, down-sample layers, res-block layers [28], concat layers and up-sample layers. Borrowing the designs from super-resolution and texture expansion [29], [30], [31], an up-sample architecture is introduced in G_{p1} to obtain a natural stylized image with high resolution. In addition, the skip connection is involved in the architecture inspired by U-Net [32]. G_{p1} performs multi-task during the process, containing structure/texture synthesis and expansion, shape transfer and super-resolution, which can map

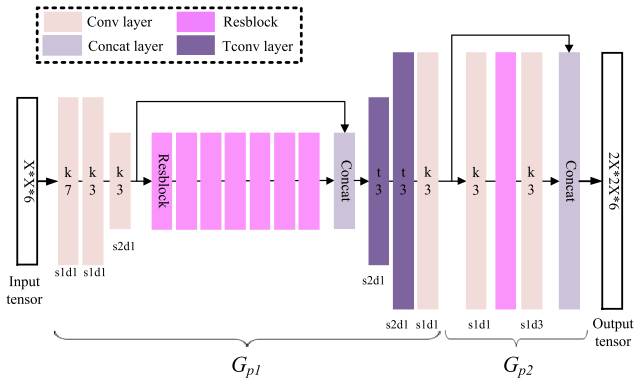


Fig. 6. The architecture of the pro-gen GAN's generator. Conv layer and Tconv layer denote the convolutional layer and transposed convolutional layer. $sidj$ implies that the stride and dilation rate of this layer are i and j, respectively. Conv layer with stride 2 can down-sample features, while Tconv with stride 2 can up-sample features. kj or tj means the kernel size is $j \times j$ for Conv or Tconv.

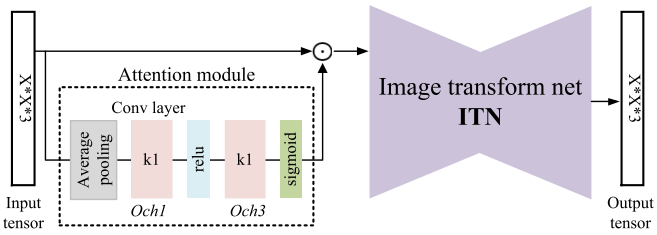


Fig. 7. The architecture of structure network (N_s), which consists of an attention module and image transform net [17]. $\odot$ indicates element-wise multiplication. ki represents that the size of convolutional kernel is $i \times i$ and $Ochj$ denotes that the number of output channels is j.

the reference styles from coarse to fine. Particularly, its advantages are two-fold: 1) The skip connection in the architecture can retain more useful information, extract and combine features at different levels, which improves the visual effects of coarse-level style transfer. 2) The up-sample design can obtain a $2\times$ magnification output feature map with high resolution. It provides more space for texture generation and characteristic imitation, achieving more natural fine-level details.

For training G_{p1}, the paired data $(\bar{y_i}, y)$ is used to train for the texture synthesis and shape mapping. G_{p1} is tasked to reconstruct y:

$$\min_{G_{p1}} \max_{D_p} \lambda_p^{abv} L_p^{adv} + \lambda_p^{style} L_p^{style} + \lambda_p^{rec} L_p^{rec}. \tag{1}$$

Let $\delta_j(x, y)$ be the style loss measuring the style similarity of image x and image y in VGG layer j:

$$\delta_j(x, y) = ||G(\phi_j(x)) - G(\phi_j(y))||_2^2, \tag{2}$$

where $G(\cdot)$ is the Gram matrix defined as in NST [11] to model the style feature, and $\phi_j(x)$ represents the features obtained from a pre-trained VGG model. Then we have:

$$\begin{aligned} L_p^{adv} &= E_{y,\bar{y_i}}[logD_p(y)] \\ &\quad + E_{y,\bar{y_i}}[log(1 - D_p(G_{p1}(\bar{y_i})))], \\ L_p^{style} &= E_{y,\bar{y_i}} \sum_j [w_j \delta_j(G_{p1}(\bar{y_i}), y)], \\ L_p^{rec} &= E_{y,\bar{y_i}} ||G_{p1}(\bar{y_i}) - y||_1. \end{aligned} \tag{3}$$

In our pro-gen GAN, D_p learns to discriminate the authenticity of the output of G_{p1}. The adversarial loss is imposed to force G_{p1} to generate a vivid stylized prototype. $\delta_j(\cdot)$ in (3) is computed with the outputs activated by the $relu1\_1$, $relu2\_1$, $relu3\_1$, $relu4\_1$, and $relu5\_1$ layers of a pre-trained VGG model, and w_j is the weight of each layer. The style loss L_p^{style} exploits the VGG model to extract high-level features, helping the network better perceive the characteristics of style images.

G_{p2} is a well-designed segmentation module. Considering the architectures of semantic segmentation networks [33], [34], [35], the segmentation module makes use of dilated convolution layer to extract the mask of the stylized images, which can expand the receptive field to obtain the global information. G_{p2} uses the original mask patch m for network training, which aims to extract the mask of text prototype:

$$\min_{G_{p2}} \lambda_p^{mse} L_p^{mse}, \tag{4}$$

where

$$L_p^{mse} = E_{\bar{y_i},m} ||G_{p2}(y) - m||_2^2. \tag{5}$$

The training of G_{p2} only needs a simple MSE loss [36] to learn the mask extraction. For a stable network training, we first train G_{p1} until the network convergence and fix the parameters of G_{p1}, and then train G_{p2} for mask extraction.

In a nutshell, the processing of prototype generation has three advantages: 1) It can generate the prototype of stylized texts, and imitate contour characteristics from simplifying text masks and tailored textures. 2) More texture details can be extended by the $2\times$ magnification network so that a high-resolution text image is obtained. 3) The mask of output stylized text can be extracted so that the background-fixed strategy is performed in the following step to obtain a clear background.

C. Backward Structure Refinement (N_s)

The shape and distribution of style elements, *e.g.*, the contour and direction of each small leaf, are important structure characteristics. As shown in Fig. 14(c), the primary result y' has coarse style characteristics after prototype generation, but the fine-level characteristics, such as structure and texture features, are ill-defined. Hence, we first develop a structure network (N_s) to repair the structure features of y'. The method in [17] transfers a reference style into content images in an unsupervised manner, which inspires us that the rendering ability can be remoulded to refine the structure features by setting text prototypes as content images.

As shown in Fig. 7, the architecture of N_s is built based upon the image transformation network [17], where we further design an attention module [37] to help N_s concentrate more on the complex style elements instead of background. The loss function of N_s contains feature reconstruction loss, style reconstruction

loss, and variation regularization. The objective of N_s can be defined as:

$$\min_{N_s} \lambda_s^{feat} L_s^{feat} + \lambda_s^{style} L_s^{style} + \lambda_s^{TV} L_s^{TV}, \tag{6}$$

where L_s^{feat} is the feature reconstruction loss, L_s^{style} denotes the style reconstruction loss, and L_s^{TV} represents the total variation regularization. They can be computed as:

$$L_s^{style} = E_{Y^l, y_{sr}} \sum_{j=0}^{j} [w_j \delta_j(Y^l, y_{sr})],$$

$$L_s^{feat} = E_{y_{sr}} \left|\left| \sum_{j} [\phi_j(y_{sr}) - \phi_j(y_{content})] \right|\right|_2^2, \tag{7}$$

where $\delta_j(\cdot)$ is defined in (2), whose $\phi_j(\cdot)$ is computed at layers $relu1\_1$, $relu2\_1$, and $relu3\_1$. Following [17], we use total variation regularization to encourage spatial smoothness in the output image. For network training, the general image dataset, denoted as $y_{content}$, is exploited as content images to enhance the adaptability of the network, as shown in Fig. 3-I.

The difference between our refinement task and the task in [17] is that N_s should pay more attention on structure details instead of global information. Shallow layers in VGG model can extract more detail features, such as color, structure, and texture, which are usful for the text stylization. Thus, compared with [17], we adopt shallow output feature maps of the VGG [38] when computing L_s^{style} and L_s^{feat} of N_s.

D. Testing Step

The testing step for text style transfer is shown in Fig. 3-II. G_p, N_s and N_t are used to perform prototype generation, structure refinement and texture refinement in order, transferring the reference style from coarse to fine level. Note that we can obtaion multi-scale stylized texts by feeding binary text masks and a style image into a deformable module under the control of the deformable factor f.

Forward prototype generation with deformable module: To obtain the multi-scale stylized texts, a deformable module is developed to control the stylistic degree of input masks. As illustrated in Fig. 8, three factors in deformable module are related to the style control. Two are related to the edge eroding and the rest is inner deformation degree. Specifically, we first erode the edge of the mask, and add noise in the erode edge. Secondly, the mask with noise is dilated as shown in Fig. 8(a). The scale of edge deformation is controlled by (f_0, f_1), where f_0 is the kernel size of eroding and dilating and f_1 is the degree of noise addition in edge. For inner deformation, noises are added inside the text under the control of the factor f_2. We name the combination of (f_0, f_1, f_2) as vector f, which determines the level of deformation. Thus, the multi-scale text masks can be obtained by changing f as shown in Fig. 8(b). The deformation module is only used in the testing step to process text masks, and then the deformable mask tailors Y to obtain rough stylistic text $\bar{y}'_s$. After deformation, the multi-scale $\bar{y}'_s$ are sent to G_p, N_s and N_t in order, to obtain the multi-scale artistic texts without network retraining, as presented in Fig. 8(c).

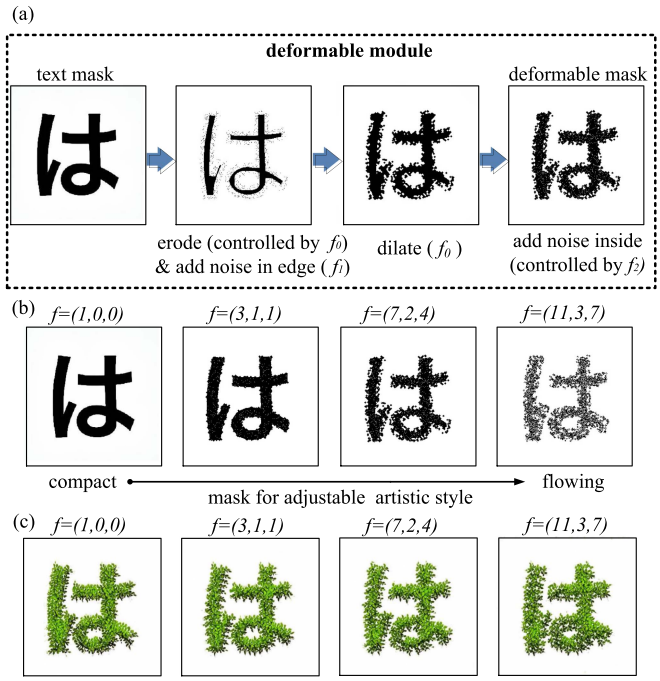


Fig. 8. Deformable module for controllable style transfer. (a) The steps of the style scale control. (b) The multi-scale text masks. (c) The multi-scale stylized texts.

By the deformable module and G_p, the style characteristics are firstly mapped to source texts in coarse-level, generating a text prototype. The first step applies binary text mask $\bar{m}'_s$ with size $w \times w$ and a style image Y as inputs. As shown in Fig. 3-II, the contour of $\bar{m}'_s$ is straight. $\bar{m}'_s$ can be sent into the deformable module to obtain the deformable mask $\bar{m}'_d$, and the shape scale can be controlled by a parameter f. Then Y is down-sampled and cropped to $w \times w$ to match the size of $\bar{m}'_d$. The result is tailored by $\bar{m}'_d$ to obtain the rough artistic text $\bar{y}'_s$. It means that we mask out the background information of Y according to the white region of $\bar{m}'_d$ to form rough stylistic text. Then, we concatenate the tailored texture $\bar{y}'_s$ and text mask $\bar{m}'_d$ to obtain a six-channel input of G_p. Finally, G_p generates a high-resolution $2\,w \times 2\,w$ stylized text prototype y' and its mask m'. By adjusting f, users can change the irregularity degree of the contours for $\bar{m}'_d$, thus to control the stylistic degree of prototype y', i.e., deformation degree to mimic the structure feature of the style elements in Y. Compared with $\bar{m}'_s$, the edge of m' has been transferred the shape characteristic of style elements. For clarity, the forward prototype generation can be described as follows:

$$\begin{aligned} \bar{m}'_d &= D[\bar{m}'_s; f], \\ \bar{y}'_s &= T[Y; \bar{m}'_d], \\ (y'; m') &= G_p[\bar{y}'_s; \bar{m}'_d], \end{aligned} \tag{8}$$

where D and T represent the processes of deforming text masks and tailoring texture, respectively.

Forward structure refinement: Having obtained the coarse-level result y', the second step is to repair its structure and shape features in fine-level, which is performed by N_s. The input is y'

and the structure-highlighted output is represented as y'_{sr}. The forward structure refinement can be expressed as follows:

$$y'_{sr} = N_s(y'). \tag{9}$$

Forward texture refinement: After reparing structure features, strengthening textures in fine-level is still important for a life-like visual effect. Given the structure refinement result y'_{sr}, the texture refinement is performed by a texture network, namely N_t, to obtain the final result y'_{tr}. Inspired by the style transfer method in [11], N_t is a pre-trained VGG network [38]. Like in N_s to emphasize the low-level texture details, we utilize lower convolutional layers of the pre-trained VGG, $relu1\_1$, $relu2\_1$, and $relu3\_1$, to compute the loss function of N_t. In our implementation, we use y'_{sr} as the input image to provide content information. Y^g is chosen as the style image. Because the overall style has been mapped into the target texts, the optimation of N_t only need $1 \sim 2$ iterations to refine texture features, which is different from the work [11] that costs hundreds of iterations. The objective of N_t is to use gradient descent to minimize the style difference between y'_{tr} and Y^g:

$$\min_{N_t} \ \lambda_{style} L_t^{style}, \tag{10}$$

where L_t^{style} can be computed as:

$$L_t^{style} = E_{y_{tr},' Y^g} \sum_j [w_j \delta_j(y_{tr},' Y^g)]. \tag{11}$$

Because Y^g is a free-form image, a background-fixed strategy is developed to make the background cleaner during the iteration of N_t. It means that only the foreground (text part) is updated during the iteration while the pixels of background is equal to y', where the foreground and background are distinguished by the binary mask m'. After $1 \sim 2$ iterations, the texture refined result y'_{tr} is obtained, which is also the final stylized texts. The forward texture refinement can be described as follows:

$$\mathbf{y'_{tr}} = \mathbf{m'y'} + (\mathbf{1} - \mathbf{m'})N_t(\mathbf{y'_{sr}}). \tag{12}$$

Because the texture refinement task does not require training a new network, the optimization only cost around $10 \sim 18$s, saving much time than [11].

IV. Experimental Results

A. Implementation Details

For the pro-gen GAN, we use the proposed architecture in Fig. 6 for generator, and the discriminator follows [30]. In input preprocessing, L is set to 500, and N is set to 128. It means that we randomly crop the style image to 256×256 sub-images and next crop 128×128 local patches for network training. The Aadm optimizer is adopted with a fixed learning rate of 0.002, and it will be halved after 10,000 iterations to stabilize training of pro-gen GAN. In (1), λ_p^{adv}, λ_p^{style}, and λ_p^{rec} are set to 1, 100, and 1, respectively. Following [11], we compute L_p^{style} using Gram matrices for the feature maps output by the $relu1\_1$, $relu2\_1$, $relu3\_1$, $relu4\_1$, and $relu5\_1$ layers, and their weights to sum up the corresponding Gram losses are given to 0.244, 0.061, 0.015, 0.004, and 0.004, respectively. The probability $1 - a$ of choosing Y^l is set to 0.9 to perform the mix-attention training scheme. To train the structure network, we exploit COCO dataset[1] as content images for the unsupervised learning. Note that structure and texture refinements are alternative. For some simple style images with smooth structures or monotonous textures, the prototype image is visually pleasing enough, and N_s or/and N_t can be skipped for faster text stylization.

B. Comparisons With State-of-The-Art Methods

Artistic style transfer: In Fig. 9, we show the qualitative comparison with five state-of-the-art methods, NST [11], T-effect [5], TET-GAN [6], Shape-Matching GAN [12], and UT-Effect [14]. The first one is an image style transfer method and the others are the methods for artistic text synthesis. NST fails to render the valid texture details in the text region, and can not find the spatial relationship between style elements and texts. In addition, NST performs poorly on imitating structure characteristics. Therefore, its results have twisted texture and confused structure. The following two methods [5] and [6], directly map the simple characteristics, such as color, to the text region, missing the texture details. Moreover, the two methods also fail to imitate shape features, yielding rigid contours. Shape-Matching GAN [12] is proposed to render artistic texts from a source style image, which is similar to our task. However, after reconstructing the text contours, the style details are overfitting by learning a pixel-to-pixel level mapping. Thus the method fails on the style images with complex texture and structure, yielding unnatural contour and blurry texture. UT-Effect [14] transfers structure and texture, balancing shape legibility with texture consistency. It can retain texture features of style images for artistic texts. However, for some complex style images, UT-Effect [14] can not extract the characteristic of shape well, causing disordered text contours. In addition, UT-Effect can not learn global features effectively. For example, in leaf style, the overall characteristic, such as the direction and distribution of the leaf, are not be perceived, and thus the leaf texture is directly mapped into the vertical stroke. By comparison, our framework is able to reconstruct the vivid details and contour characteristics accurately in an unsupervised manner, achieving the most visually appealing texts.

To quantitatively measure the performance, we conducted a user study on a public platform.[2] The study gives image pairs to various observers, and tasks to choose which one is of the best style similarity with the reference style image while maintaining legibility. Then, we compute preference ratio as quantitative metrics. It is the percentage of times a method is selected in all of its related selections, whose range is from 0 to 1. If it is significantly better than all other methods, then its mean preference ratio will exceed 0.5; if it is perfect, its mean preference ratio will be near 1. The preference ratio over 11 test styles is shown in Table I. Obviously, our method surpasses other works in most of cases significantly. The average preference ratio of 0.556 quantitatively verifies the superiority of our method.

Controllable text style transfer: In Fig. 10, we demonstrate the qualitative comparison with the state-of-the-art

[1][Online]. Available: https://cocodataset.org.

[2][Online]. Available: https://www.wenjuan.com/publish

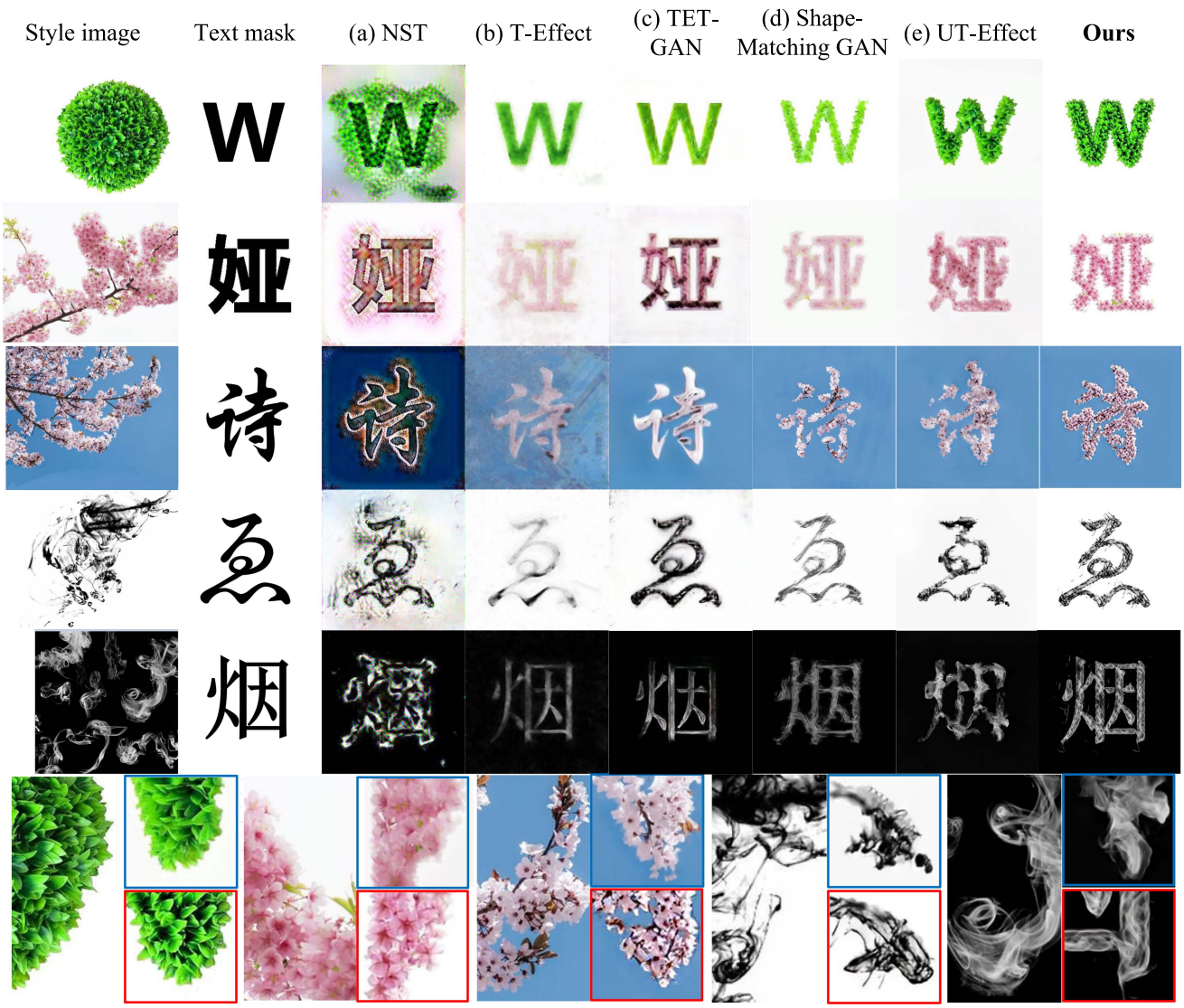


Fig. 9. Qualitative comparisons with state-of-the-art methods on various styles. The first two columns are style images and binary masks of target texts. In the last two row, we present the comparison of enlarged local patch in source style images, UT-Effect (marked in blue boxes) and our results (marked in red boxes).

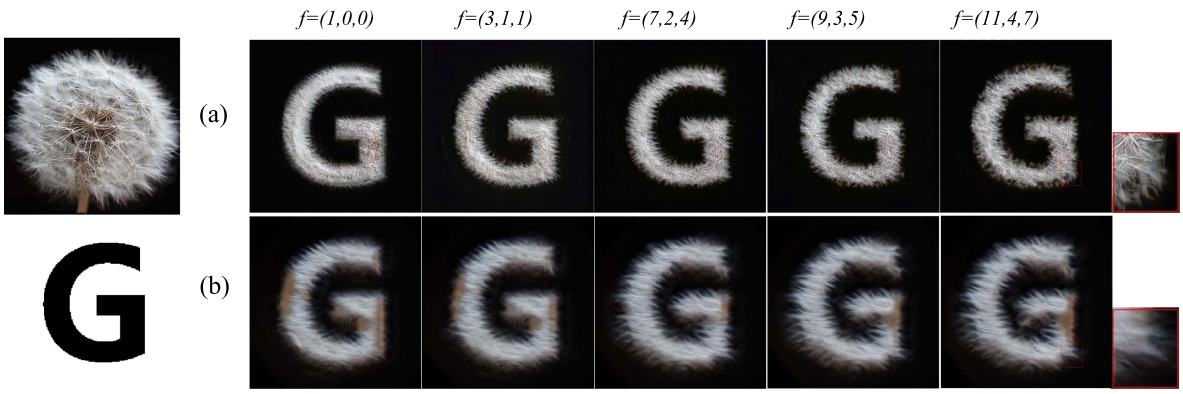


Fig. 10. Comparison between our method and previous scale-controllable style transfer method, where (a) shows the results of our method and (b) gives the outputs of Shape-Matching GAN [12]. Local patches (marked in red boxes) are enlarged in the right for a better comparison.

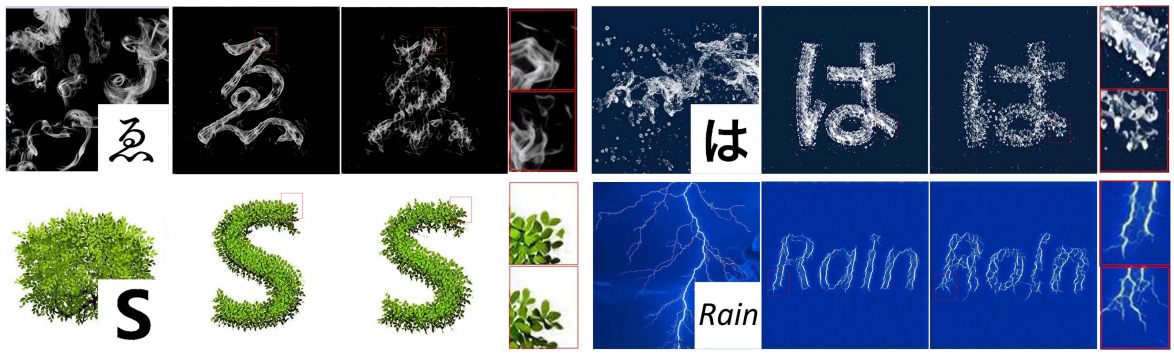


Fig. 11. Visual results of the proposed method for controllable style transfer. Local patches (marked in red boxes) are enlarged in the right of their results.

TABLE I
PREFERENCE RATIOS OF NST [11], T-EFFECT [5], TET-GAN [6], SHAPE-MATCHING GAN [12], UT-EFFECT [14]. FOR EACH ROW, WE SHOW THE BEST PREFERENCE RATIO IN BOLD AND THE SECOND UNDERLINED

Style	[11]	[5]	[6]	[12]	[14]	Ours
leaf	0	0.014	0.014	0.014	0.229	**0.729**
flower	0.020	0.014	0.029	0.014	0.1	**0.843**
dandelion	0	0.057	0.086	0.057	0.086	**0.714**
cookie	0	0.029	0	0.029	0.457	**0.486**
sakura	0.029	0.029	0.029	0.129	0.271	**0.514**
smoke	0.085	0.071	0.029	0.172	**0.358**	0.286
plant	0.029	0.057	0.057	0.057	0.029	**0.771**
wave	0.086	0.029	0.057	0	0.400	**0.429**
peach blossom	0.029	0.0	0.029	0.1	0.342	**0.571**
thunder	0.057	0.114	0.029	0.371	0.057	**0.400**
cloth	0.057	0.029	0.343	0.086	0.114	**0.371**
Average	0.036	0.040	0.064	0.093	0.222	**0.556**

TABLE II
RUNNING TIME OF THE DEFORMABLE MODULE

Image Size	Test Number	Average Time (per frame)
256×256	100	17.2ms
320×320	100	17.3ms

scale-controllable method, Shape-Matching GAN [12]. As shown in Fig. 10(b), Shape-Matching GAN loses the texture details, yielding rigid and blurry texture. In addition, the glyph deformation of Shape-Matching GAN is incomplete for complex style elements, and thus the shapes are not fully transferred to the vertical stroke. On the contrary, the proposed framework achieves continuous multi-scale stylized texts with exquisite details and vivid contours as shown in Fig. 10(a). The multi-scale artistic styles vividly shows the natural process as if the dandelion grows from compact to fluffy. Fig. 11 presents the multi-scale artistic results of the proposed framework on different style images. For complex and varied style characteristics, such as the fluid shape (smoke), the dense and irregularity style (plants), and the spindly contour (thunder), our framework is robust to map these complex style effects into continuous transformation with a controllable scale, creating fine artistic presentations.

In Table II, we report the running time of the proposed style control method with 4 cores of Intel i9-9900 CPU @3.5 GHz. The average running time of the deformable module is listed for processing 100 images in the testing phase, where the sizes of input images are 256×256 and 320×320.

As shown in Table II, our style control method requires less than 50 ms to process per frame without retraining networks, which implies a potential of nearly real-time control.

C. Ablation Study

Network architecture: To analyze the effect of each component in our framework, we perform the following experiments with different configurations of network architecture.

- Only G_p: The model architecture only contains the pro-gen GAN G_p. The binary mask, together with tailored texture is sent to G_p to obtain the output images.
- Only G_p and N_s: This model architecture is composed of a pro-gen GAN and a structure network. The binary mask and tailored texture are sent to G_p and N_s in order, obtaining the output images.
- Only G_p and N_t: This model architecture contains G_p and N_t. It means that the results obtained by G_p are directly sent into N_t to get the output images.
- Only N_s and N_t: This architecture only contains N_s and N_t. It does not apply G_p to generate prototypes, and the binary mask is directly used in N_s and N_t to reconstruct the structure and texture characteristics.
- Full model: All the proposed models in the framework, G_p, N_s and N_t, are used to obtain the final results.

Fig. 12 shows the results of different network configurations. As shown in Fig. 12(a), generating stylized texts only using G_p causes rigid texture and poor structure effect. In Fig. 12(b), G_p together with N_s, could create the vivid contours and overall structure features, but it can not transfer the fine texture characteristics, such as the different levels of petal colors, leading to monotonous color effect. Fig. 12(c) shows the visual result obtained by only using G_p and N_t. The major role of N_s is to repair the basic structure of style elements for text prototypes. Without N_s, the structure of stylized texts become rigid. In addition, G_p plays a quite important role in the framework. As shown in Fig. 12(d), without the prototype generation from G_p, the architecture fails on shape transferring and texture reconstruction, generating confused stripes and unclear edges. Figs. 12(e) are the results of full model for two style images. Obviously, by exploiting G_p, N_s and N_t to transfer styles at different levels, our full model can synthesize high-quality artistic texts, with delicate textures and vivid contours.

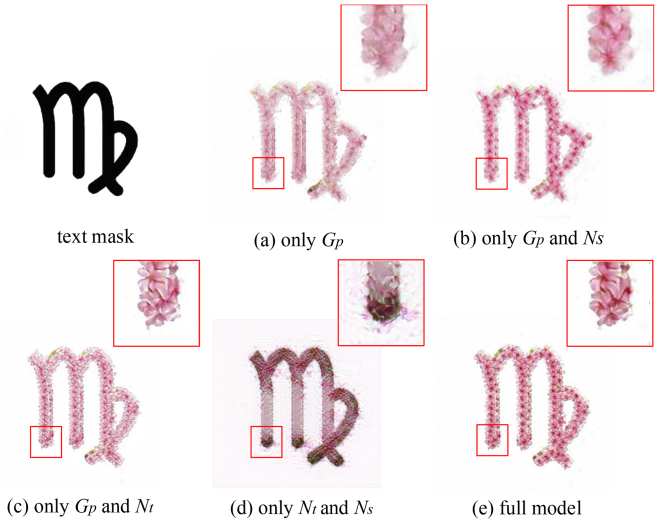


Fig. 12. Analysis for the effect of each component in the proposed framework.

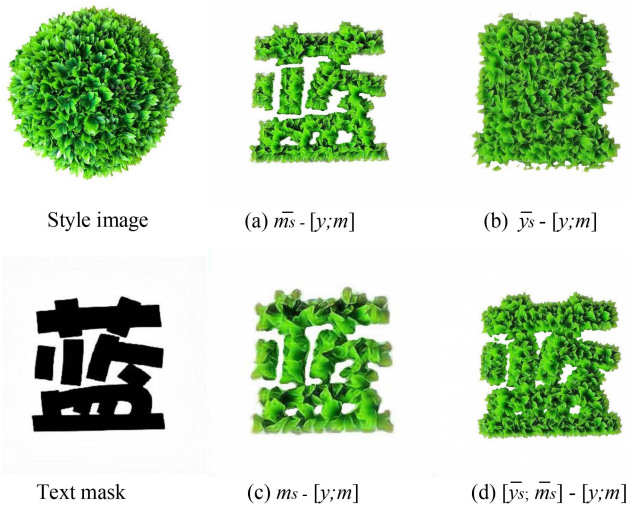


Fig. 13. Analysis for the effect of each data processing step, where (a), (b), (c) and (d) are the stylized results of the framework.[3]

Input preprocessing: To analyze the effect of each step in the preprocessing method, we perform the following preprocessing methods to train G_p, and then the testing result y' is sent to N_s and N_t to obtain the final result y'_{tr}.

- $\bar{m}_s$ - $[y; m]$: We only use $\bar{m}_s$ as input and $[y; m]$ as ground truth to train G_p.
- $\bar{y}_s$ - $[y; m]$: We set $\bar{y}_s$ as input of training data, and $[y; m]$ as ground truth.
- m_s - $[y; m]$: We modify G_p into a $1\times$ network, where an up-sample layer is removed. Then, to train G_p, we use m_s as input, and $[y; m]$ is used as ground truth, whose patch size is same as m_s.
- $[\bar{y}_s; \bar{m}_s]$ - $[y; m]$: All the proposed processing steps are applied to generate the training data. Specifically, we use $[\bar{y}_s; \bar{m}_s]$ as input, and $[y; m]$ as ground truth to train G_p.

[3]The style image source: [online]. Available: https://jingjia.tmail.com

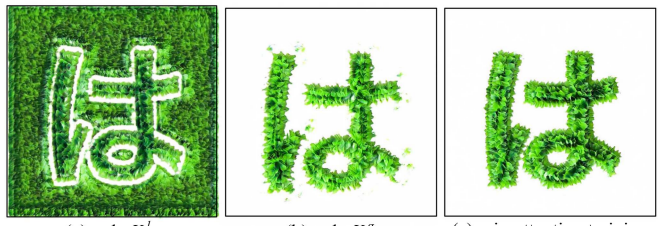
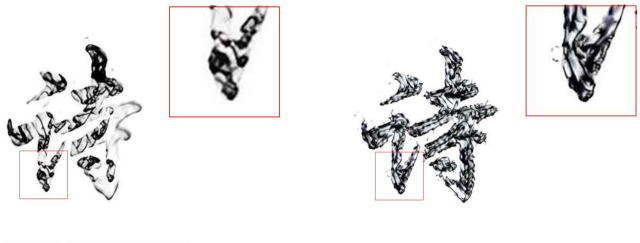


Fig. 14. Effect of our mix-attention training scheme, where (a), (b) and (c) are the test results of G_p.

In Fig. 13(a), for lack of the down-sampled style patch $\bar{y}_s$, some structure features of the stylized texts are fuzzy compared with Fig. 13(d). It proves that the down-sampled style patch $\bar{y}_s$ can provide a crucial guidance for feature reconstruction. In Fig. 13(b), the down-sampled mask patch $\bar{m}_s$ is removed during the network training. Compare with Fig. 13(d), the contours of stylized texts are poor, which verifies that the mask patch $\bar{m}_s$ can give explicit contour information for shape mapping. In Fig. 13(c), a $1\times$ network is used and the down-sampled style patch $\bar{y}_s$ is removed. Obviously, the configuration generate ambiguous stylized texts with vague textures and rigid contours. This may be attributed to that using $1\times$ training data is easy for the network to overfit, thus failing to learn the valid features. On the contrary, the proposed preprocessing method, provides enough style and shape information and spaces for network learning, thus achieving the most visually appealing results.

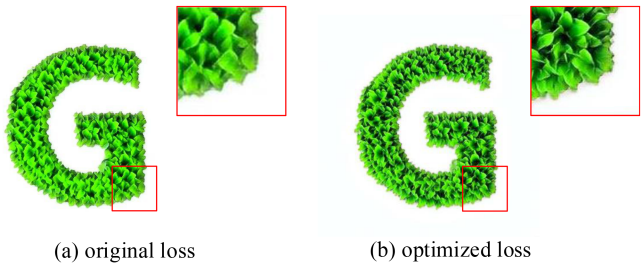
Mix-attention training scheme for G_p: Fig. 14 demonstrates the effect of the mix-attention training scheme. In Fig. 14(a), using cropped patches only in Y^l brings unclear background and smooth contour, which can not imitate global features effectively. The reason may be that Y^l cannot provide enough background information to learn, causing G_p to incorrectly identify the relationship between background and style elements. Thus, G_p will go according to the characteristic of Y^l, and map the most area of input image into style elements. Conversely, in Fig. 14(b), training only with Y^g can render the vivid overall contours with clear background, but texture in the foreground is fuzzy with some artifacts. Fig. 14(c) shows the output of G_p with our mix-attention training scheme, which has clear overall contours, delicate textures and local details, surpassing the training schemes with only one image drastically.

The $2\times$ magnification G_p: One of the most important improvements of the proposed framework, is introducing a $2\times$ magnification design to G_p. Fig. 15 demonstrates the effect of the proposed $2\times$ design, where Fig. 15(a) represents the stylized text using a $1\times$ network for prototype generation (here we remove the up-sample layer in G_p to obtain a $1\times$ network, and use paired data $[y_s; m_s]$ - $[y; m]$ to train G_p), and Fig. 15(b) is the output of the full model. Compared with $2\times$ network, the $1\times$ network does not have extra space for style imitation, and it is hard to establish a pixel-to-pixel corresponding relationship between input data and style images. Due to the limited ability of representation and learning, the $1\times$ network occurs overfitting and can not learn the valid information from training data. On



(a) 1x network (b) 2x network

Fig. 15. Effect of the proposed $2\times$ magnification G_p.



(a) original loss (b) optimized loss

Fig. 16. Effect of the perceptual loss L_s^{style} and L_s^{feat} in N_s.

the contrary, the $2\times$ generator is more free to learn the mapping relationship from the training data, and extends the input texture for a natural details, thus achieving the most visually appealing results.

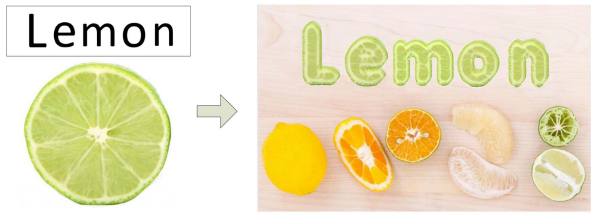
Loss function: We examine the effect of different layers of a VGG to compute the perceptual loss (L_s^{style} and L_s^{feat}) in N_s. When the features extracted from deeper layers in VGG are used to compute perceptual loss, the ability of structure repairation will be lowered. Fig. 16 shows the stylized results of computing L_s^{style} and L_s^{feat} by different combinations of layers. Fig. 16(a) is obtained by training N_s using deeper features (the original loss function in [17]), and those layers of VGG are $relu1\_2$, $relu2\_2$, $relu3\_2$, and $relu4\_2$. Fig. 16(b) is the result with our optimized loss, using layers $relu1\_1$, $relu2\_1$, $relu3\_1$. Applying deeper features for the perceptual loss, the structure of style element becomes ambiguous and the overall visual effects are poor. As shown in Fig. 16(b), the modified perceptual loss achieves a better ability of structure refinement.

D. Control of the Distribution of Style Elements

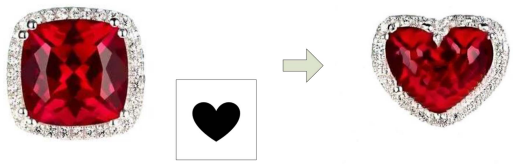
We study the effect of the size proportion of style image L and cropped patch $2\,N$. For a fair comparison, we enlarge the original style image by $1.5\times$, and test the results for the $1.5\times$ and $1\times$ style image under the same cropped size N. In the experiment, the size of $1\times$ style image (Y^l) L is 500×500, and the crop size of local patch, $2\,N$ is set to 256. The proportion $L/2\,N$ in Fig. 17(a) is equal to $750/256$, which is relatively larger. Therefore, the distribution of style elements is relatively scattered and the size of each element is larger. Otherwise, as shown in Fig. 17(b), the proportion $L/2\,N$ equals $500/256$, and the smaller proportion brings tight arrangement and small size of basic style elements in artistic texts. The study reflects that the form of reconstructed style elements is related to the size of local receptive field. In the real application, we can adjust the proportion for the requirements of practical tasks.

(a)1.5x (b)1x

Fig. 17. Analysis for different size proportions of style image and cropped patch, where the same size local patches in the style image (marked in black boxes) and stylized texts (marked in red boxes) are listed for comparison.



(a) Advertisement and typography



(b) Fashion design

Fig. 18. Applications of our framework.

E. Applications

Advertisement and typography: Our method can be easily applied to stylize various source texts. Since no special artistic text dataset is required, we can flexibly choose the style image according to the needs of tasks. In Fig. 18(a), we show an example of text synthesizing to match the style of background for advertisement and typography.

Fashion design: The proposed framework can not only transfer the style characteristics into various texts, but also can transfer the source style into the more general shapes. As shown in Fig. 18(b), it enables users to generate new designs and perform pattern conversion depending on the existing templates, which can be used for art design, such as food, cloth or other fashion applications [4].

F. Limitation

Although the proposed approach has generated visually appealing results, some limitations still exist. When multiple style elements in a source image have distinctively different features and some features are not suitable for all the text strokes, shape mismatching and characteristic mixture will occur. It might be

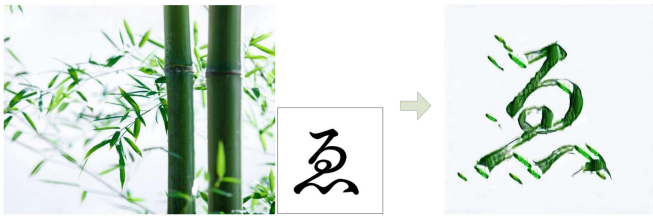


Fig. 19. An example of failure case.

beyond its capability to analyze the relevance between style characteristics and text strokes, and then map different styles into corresponding areas. As shown in Fig. 19, we try to transfer a bamboo style into a targeted text. However, the characteristics of trunks and leaves in the image differ considerably, and the shape feature of slender leaves is not suitable for all the text strokes. It is hard for our method to search a best matching scheme to transfer leaf and trunk into different strokes, respectively. Hence, the transferred features in the text are disordered, lowering the legibility and artistry. The problem may be addressed by exploring reasonable matching strategy for different style areas, or adding the specialized guidance for each feature by user interaction.

V. Conclusion and Future Work

In this paper, we present a coarse-to-fine framework to synthesize artistic texts with complex texture and structure, which can also control the scale of style mapping. Particularly, we exploit a $2\times$ magnified pro-gen GAN and one-shot learning method to transfer the complex style features in an unsupervised way. In addition, a mix-attention training scheme is introduced to enhance the visual results. Our framework breaks a barrier of complex text style transfer, allowing users to create fine artistic presentations by a single style image.

There are still some interesting issues for further work. Transferring the complex styles with multiple different characteristics into texts bring more vitality for art design, so a future direction is investigating the visual consistency and relevance between different style areas and text strokes to achieve a best matching results. In addition, mapping exquisite decors to the complex stylized texts is worthy of investigation, which will contribute more aesthetic interests.

References

[1] S. Meyer, V. Cornillère, A. Djelouah, C. Schroers, and M. Gross, "Deep video color propagation," in *Proc. Brit. Mach. Vis. Conf.*, Newcastle upon Tyne, U.K., Sep. 2018.

[2] J. Ren, X. Shen, Z. Lin, R. Mech, and D. J. Foran, "Personalized image aesthetics," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 638–647.

[3] X. Wang, G. Oxholm, D. Zhang, and Y.-F. Wang, "Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5239–5247.

[4] S. Jiang and Y. Fu, "Fashion style generator," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 3721–3727.

[5] S. Yang, J. Liu, Z. Lian, and Z. Guo, "Awesome typography: Statistics-based text effects transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7464–7473.

[6] S. Yang, J. Liu, W. Wang, and Z. Guo, "Tet-GAN: Text effects transfer via stylization and destylization," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 1238–1245.

[7] 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. AAAI Conf. Artif. Intell.*, 2020, vol. 34, pp. 1717–1724.

[8] W. Wang, J. Liu, S. Yang, and Z. Guo, "Typography with decor: Intelligent text style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5889–5897.

[9] S. Yang, W. Wang, and J. Liu, "TE141K: Artistic text benchmark for text effect transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3709–3723, Oct. 2021.

[10] G. Atarsaikhan, B. K. Iwana, A. Narusawa, K. Yanai, and S. Uchida, "Neural font style transfer," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit.*, 2017, vol. 5, pp. 51–56.

[11] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2414–2423.

[12] S. Yang et al., "Controllable artistic text style transfer via shape-matching GAN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4442–4451.

[13] S. Yang, Z. Wang, and J. Liu, "Shape-matching GAN++: Scale controllable dynamic artistic text style transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3807–3820, Jul. 2022.

[14] S. Yang, J. Liu, W. Yang, and Z. Guo, "Context-aware text-based binary image stylization and synthesis," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 952–964, Feb. 2019.

[15] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[16] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 49–70, 2000.

[17] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, Berlin, Germany, 2016, pp. 694–711.

[18] A. Sanakoyeu, D. Kotovenko, S. Lang, and B. Ommer, "A style-aware content loss for real-time HD style transfer," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 698–714.

[19] D. Kotovenko, A. Sanakoyeu, S. Lang, and B. Ommer, "Content and style disentanglement for artistic style transfer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4422–4431.

[20] D. Kotovenko, A. Sanakoyeu, P. Ma, S. Lang, and B. Ommer, "A content transformation block for image style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10032–10041.

[21] S. Yang, J. Liu, Z. Lian, and Z. Guo, "Text effects transfer via distribution-aware texture synthesis," *Comput. Vis. Image Understanding*, vol. 174, pp. 43–56, 2018.

[22] S. Azadi et al., "Multi-content GAN for few-shot font style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7564–7573.

[23] Y. Gao, Y. Guo, Z. Lian, Y. Tang, and J. Xiao, "Artistic glyph image synthesis via one-stage few-shot learning," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–12, 2019.

[24] A. Zhu et al., "Few-shot text style transfer via deep feature similarity," *IEEE Trans. Image Process.*, vol. 29, pp. 6932–6946, 2020.

[25] Y. Jing et al., "Stroke controllable fast style transfer with adaptive receptive fields," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 238–254.

[26] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the Gaussian kernel for scale-space filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 1, pp. 26–33, Jan. 1986.

[27] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[29] A. Frühstück, I. Alhashim, and P. Wonka, "TileGAN: Synthesis of large-scale non-homogeneous textures," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–11, 2019.

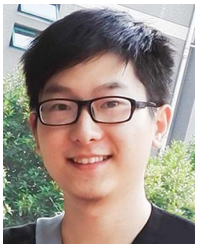
[30] Y. Zhou et al., "Non-stationary texture synthesis by adversarial expansion," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–13, 2018.

[31] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.

[32] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, Berlin, Germany, 2015, pp. 234–241.

[33] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Representations*, 2016.
[34] C. Liang-Chieh, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. Int. Conf. Learn. Representations*, May 2015.
[35] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
[36] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.*, Berlin, Germany, 2016, pp. 391–407.
[37] J. Liu, W. Zhang, Y. Tang, J. Tang, and G. Wu, "Residual feature aggregation network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2359–2368.
[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

Wendong Mao (Graduate Student Member, IEEE) received the B.S. degree in information engineering from Jilin University, Changchun, China, in 2018. She is currently working toward the Ph.D. degree in electronic and communications engineering, Nanjing University, Nanjing, China. She was a Visiting Student with the Wangxuan Institute of Computer Science and Technology, Peking University, Beijing, China, in 2019. Her research interests include image processing algorithm and VLSI design for deep learning.



Shuai Yang (Member, IEEE) received the B.S. and Ph.D. degrees (Hons.) in computer science from Peking University, Beijing, China, in 2015 and 2020, respectively. He is currently a Postdoctoral Research Fellow with the NTU AI Corporate Laboratory, Nanyang Technological University, Singapore. He was a Visiting Scholar with the Texas A&M University, College Station, TX, USA, from September 2018 to September 2019. He was a Visiting Student with the National Institute of Informatics, Chiyoda City, Japan, from March 2017 to August 2017. His research interests include image stylization and image inpainting. He was the recipient of the IEEE ICME 2020 Best Paper Awards and IEEE MMSP 2015 Top10% Paper Awards.

Huihong Shi received the B.S. degree in electronic engineering from Jilin University, Changchun, China, in 2020. She is currently working toward the Ph.D. degree with the School of Electronic Science and Engineering, Nanjing University, Nanjing, China. Her research interests include network compression, DNN accelerators, and image stylization.

Jiaying Liu (Senior Member, IEEE) received the Ph.D. degree (Hons.) in computer science from Peking University, Beijing, China, 2010. She is currently an Associate Professor, Boya Young Fellow with the Wangxuan Institute of Computer Technology, Peking University. She has authored more than 100 technical articles in refereed journals and proceedings, and holds 70 granted patents. Her research interests include multimedia signal processing, compression, and computer vision. She is a Senior Member of CSIG and CCF. She was a Visiting Scholar with the University of Southern California, Los Angeles, CA, USA, from 2007 to 2008. She was a Visiting Researcher with Microsoft Research Asia. She was a Member of Multimedia Systems and Applications Technical Committee, and Visual Signal Processing and Communications Technical Committee in IEEE Circuits and Systems Society. She is also a Member of Image, Video, and Multimedia and Signal and Information Processing Theory and Methods Technical Committee in APSIPA. In 2015, she was supported by the Star Track Young Faculties Award. She was the recipient of the IEEE ICME 2020 Best Paper Award and IEEE MMSP 2015 Top10% Paper Award. She was an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and *Journal of Visual Communication and Image Representation*, the Technical Program Chair of IEEE ICME-2021/ACM ICMR-2021, the Area Chair of CVPR-2021/ECCV-2020/ICCV-2019, and the CAS Representative at the ICME Steering Committee. She was the APSIPA Distinguished Lecturer (2016–2017).

Zhongfeng Wang (Fellow, IEEE) received the B.E. and M.S. degrees from the Department of Automation, Tsinghua University, Beijing, China, in 1988 and 1990, respectively, and the Ph.D. degree from the University of Minnesota, Minneapolis, MN, USA, in 2000. He has been with Nanjing University, Nanjing, China, as a Distinguished Professor since 2016. He was with Broadcom Corporation, San Jose, CA, USA, from 2007 to 2016 as a leading VLSI Architect. Before that, he was with Oregon State University, Corvallis, OR, USA, and National Semiconductor Corporation. Dr. Wang is a world-recognized Expert on Low-Power High-Speed VLSI Design for Signal Processing Systems. He has authored or coauthored more than 200 technical papers with multiple best paper awards. He has edited one book VLSI and held more than 20 U.S. and China patents. His research interests include optimized VLSI design for digital communications and deep learning. He was the recipient of multiple best paper awards from IEEE technical societies, among which is the VLSI Transactions Best Paper Award of 2007. In the current record, he has had many papers ranking among top 25 most (annually) downloaded manuscripts in IEEE Transaction on VLSI Systems. In the past, he was an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I,IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, and IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS for many terms. He also was the TPC Member and various chairs for tens of international conferences. Moreover, he has contributed significantly to the industrial standards. His technical proposals have been adopted by more than fifteen international networking standards. In 2015, he was elevated to the fellow of IEEE for contributions to VLSI design and implementation of FEC coding.