TET-GAN: Text Effects Transfer via Stylization and Destylization

Shuai Yang, Jiaying Liu,[*] Wenjing Wang, Zongming Guo
Institute of Computer Science and Technology, Peking University, Beijing, China
{williamyang, liujiaying, daooshee, guozongming}@pku.edu.cn

Abstract

Text effects transfer technology automatically makes the text dramatically more impressive. However, previous style transfer methods either study the model for general style, which cannot handle the highly-structured text effects along the glyph, or require manual design of subtle matching criteria for text effects. In this paper, we focus on the use of the powerful representation abilities of deep neural features for text effects transfer. For this purpose, we propose a novel Texture Effects Transfer GAN (TET-GAN), which consists of a stylization subnetwork and a destylization subnetwork. The key idea is to train our network to accomplish both the objective of style transfer and style removal, so that it can learn to disentangle and recombine the content and style features of text effects images. To support the training of our network, we propose a new text effects dataset with as much as 64 professionally designed styles on 837 characters. We show that the disentangled feature representations enable us to transfer or remove all these styles on arbitrary glyphs using one network. Furthermore, the flexible network design empowers TET-GAN to efficiently extend to a new text style via one-shot learning where only one example is required. We demonstrate the superiority of the proposed method in generating high-quality stylized text over the state-of-the-art methods.

Introduction

Text effects are additional style features for text, such as colors, outlines, shadows, reflections, glows and textures. Rendering text in the style specified by the example stylized text is referred to as text effects transfer. Applying visual effects to text is very common yet important in graphic design. However, manually rendering text effects is labor intensive and requires great skills beyond normal users. In this work, we propose a neural network architecture that automatically synthesizes high-quality text effects on arbitrary glyphs.

The success of the pioneering Neural Style Transfer (Gatys, Ecker, and Bethge 2016) has sparked a research boom of deep-based image stylization. The key idea behind it is to match the global feature distributions between the style image and the generated image (Li et al. 2017a) by minimizing the difference of Gram matrices (Gatys, Ecker, and Bethge 2015). However, this global statistics representation for general styles does not apply to the text effects. Text effects are highly structured along the glyph and cannot be simply characterized as the mean, variance or other global statistics (Li et al. 2017a; Dumoulin, Shlens, and Kudlur 2016; Huang and Belongie 2017; Li et al. 2017c) of the texture features. Instead, the effects should be learned with the corresponding glyphs. For this reason, we develop a new text effects dataset, driven by which we show our network can learn to properly rearrange textures to fit new glyphs.

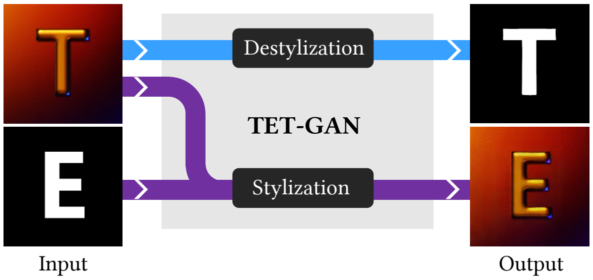


Figure 1: Overview: Our TET-GAN implements two functions: destylization for removing style features from the text and stylization for transferring the visual effects from highly stylized text onto other glyphs.

From a perspective of texture rearrangement, modelling the style of text effects using local patches seems to be more suitable than global statistics. Valuable efforts have been devoted to patch-based style transfer (Li and Wand 2016a; Chen and Schmidt 2016; Yang et al. 2017). The recent work of (Yang et al. 2017) is the first study of text effects transfer, where textures are rearranged to correlated positions on text skeleton. However, matching patches in the pixel domain, this method fails to find proper patches if the target and example glyphs differ a lot. The recent deep-based methods (Li and Wand 2016a; Chen and Schmidt 2016) address this issue by matching glyphs in the feature domain, but they use a greedy optimization, causing the disorder of the texture in the spatial distribution. To solve this problem, we introduce a novel distribution-aware data augmentation strategy to constrain the spatial distribution of textures.

To handle a particular style, researchers have looked at

[*] Corresponding author. This work was supported by National Natural Science Foundation of China under contract No. 61772043 and Peking University - Tencent Rhino Bird Innovation Fund.
Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

style modeling from images rather than using general statistics or patches, which refers to image-to-image translation (Isola et al. 2017). Early attempts (Isola et al. 2017; Zhu et al. 2017) train generative adversarial networks (GAN) to map images from two domains, which is limited to only two styles. StarGAN (Choi et al. 2018) employs one-hot vectors to handle multiple pre-defined styles, but requires expensive data collection and retraining to handle new styles. We improve these approaches by designing a novel Texture Effects Transfer GAN (TET-GAN), which characterizes glyphs and styles separately. By disentangling and recombining glyph and visual effects features, we show that our network can simultaneously support stylization and destylization on a variety of text effects as shown in Fig. 1. In addition, having learned to rearrange textures based on glyphs, a trained network can easily be extended to new user-specified text effects.

In this paper, we propose a novel approach for text effects transfer with three distinctive aspects. First, we develop a novel TET-GAN built upon encoder-decoder architectures. The encoders are trained to disentangle content and style features in the text effects images. Stylization is implemented by recombining these two features while destylization by solely decoding content features. The task of destylization to completely remove styles guides the network to precisely extract the content feature, which in turn helps the network better capture its spatial relationship with the style feature in the task of stylization. Second, in terms of data, we develop a new text effects dataset with 53,568 image pairs to facilitate training and further study. In addition, we propose a distribution-aware data augmentation strategy to impose a distribution constraint (Yang et al. 2017) for text effects. Driven by the data, our network learns to rearrange visual effects according to the glyph structure and its correlated position on the glyph as a professional designer does. Finally, we propose a self-stylization training scheme for one-shot learning. Leveraging the skills that have been learned from our dataset, the network only needs to additionally learn to reconstruct the texture details of one example, and then it can generate the new style on any glyph.

In summary, our contributions are threefold:

- We raise a novel TET-GAN to disentangle and recombine glyphs and visual effects for text effects transfer. The explicit content and style representations enable effective stylization and destylization on multiple text effects.
- We introduce a new dataset containing thousands of professionally designed text effects images, and propose a distribution-aware data augmentation strategy for distribution-aware style transfer.
- We propose a novel self-stylization training scheme that requires only a few or even one example to learn a new style upon a trained network.

Related Work

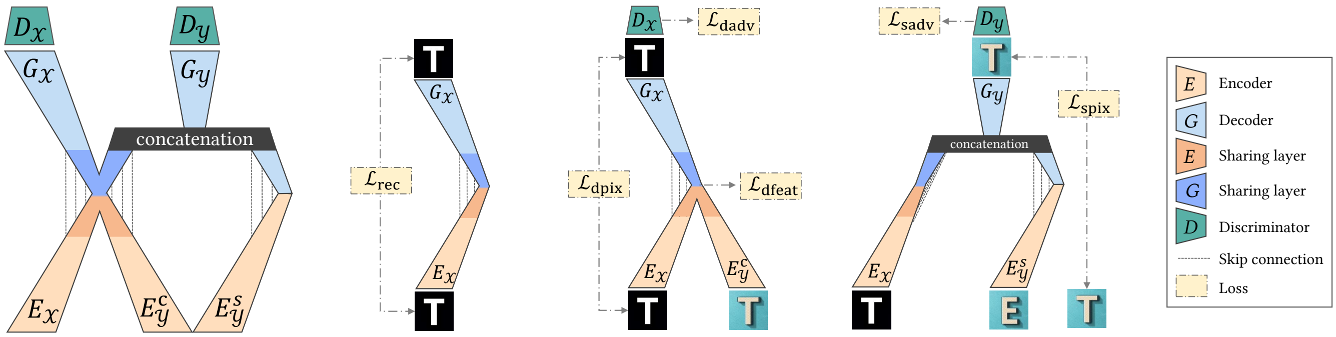
Neural style transfer. Style transfer is the task of migrating styles from an example style image to a content image, which is closely related to texture synthesis. The pioneering work of (Gatys, Ecker, and Bethge 2016) demonstrates the powerful representation ability of convolutional neural networks to model textures. Gatys *et al.* formulated textures as the correlation of deep features in the form of a Gram matrix (Gatys, Ecker, and Bethge 2015), and transferred styles by matching high-level representations of the content image and the Gram matrices. Since then, deep-based style transfer has become a hot topic, and many follow-up work improves it in different aspects such as acceleration (Johnson, Alahi, and Li 2016; Ulyanov et al. 2016; Wang et al. 2017), user controls (Gatys et al. 2017) and style diversification (Li et al. 2017b). In parallel, Li *et al.* (2016a; 2016b) modelled textures by local patches of feature maps, which can transfer photo-realistic styles.

Image-to-image translation. Image-to-image translation is a domain transfer problem, where the input and output are both images. Driven by the great advances of GAN, once been introduced by (Isola et al. 2017), it has been widely studied. Recent work (Murez et al. 2018) has been able to generate very high-resolution photo-realistic images from semantic label maps. Zhu *et al.* (2017) proposed a novel cycle loss to learn the domain translation without paired input-output examples. While most researches focus on the translation between two domains, Choi *et al.* (2018) utilized a one-hot vector to specify the target domain, so that the network can learn the mapping between multiple domains, which provides more flexibility. However, extension to new domains is still expensive. In this paper, we introduce a self-stylization training scheme to efficiently learn a new style with only one example required.

Text style transfer. Text is one of the most important visual elements in our daily life and there is some work on style transfer specific to the text. Taking advantage of the accessibility of abundant font images, many works (Lian, Zhao, and Xiao 2016; Sun et al. 2018; Zhang, Zhang, and Cai 2018) trained neural networks to learn stroke styles for font transfer. However, another type of style, namely text effects, was not studied much. It was not until 2017 that the work of (Yang et al. 2017) first raised text effects transfer problem. The authors proposed to match and synthesize image patches based on their correlated position on the glyph, which is vulnerable to glyph differences and has a heavy computational burden. Meanwhile, Azadi *et al.* (2018) combined font transfer and text effects transfer using two successive subnetworks and end-to-end trained them using a synthesized gradient font dataset. However, they can only handle 26 capital letters with a small size of 64×64, and their synthesized dataset differs greatly from the actual text effects. By contrast, we build our dataset using in-the-wild text effects with a size of 320×320, supporting our network to render exquisite text effects for any glyph.

TET-GAN for Text Effects Transfer

Our goal is to learn a two-way mapping between two domains $\mathcal{X}$ and $\mathcal{Y}$, which represent a collection of text images and text effects images, respectively. Our key idea is to train a network to simultaneously accomplish two tasks: one to combine text effects (style) with glyphs (content) for stylization, and another to remove text effects for destylization. As



(a) Network Architecture (b) Autoencoder (c) Destylization (d) Stylization

Figure 2: The TET-GAN architecture. (a) An overview of TET-GAN architecture. Our network is trained via three objectives of autoencoder, destylization and stylization. (b) Glyph autoencoder to learn content features. (c) Destylization by disentangling content features from text effect images. (d) Stylization by combining content and style features.

shown in Fig. 2, our framework consists of two content encoders $\{E_{\mathcal{X}}, E^c_{\mathcal{Y}}\}$, a style encoder $\{E^s_{\mathcal{Y}}\}$, two domain generators $\{G_{\mathcal{X}}, G_{\mathcal{Y}}\}$ and two domain discriminators $\{D_{\mathcal{X}}, D_{\mathcal{Y}}\}$. $E_{\mathcal{X}}$ and $E^c_{\mathcal{Y}}$ map text images and text effects images onto a shared content feature space, respectively, while $E^s_{\mathcal{Y}}$ maps text effects images onto a style feature space. $G_{\mathcal{X}}$ generates text images from the encoded content features. $G_{\mathcal{Y}}$ generates text effects images conditioned on both the encoded content features and style features. The discriminators are trained to distinguish the generated images from the real ones.

Given these basic network components, we can define our two-way mapping. The forward mapping (stylization) $G_{\mathcal{Y}} \circ (E_{\mathcal{X}}, E^s_{\mathcal{Y}}) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{Y}$ requires a target raw text image x and an example text effects image y' as input, and transfers the text effects in y' onto x, obtaining y. Meanwhile, the backward mapping (destylization) $G_{\mathcal{X}} \circ E^c_{\mathcal{Y}} : \mathcal{Y} \rightarrow \mathcal{X}$ takes y as the input, and extracts its corresponding raw text image x. In addition, we further consider an autoencoder $G_{\mathcal{X}} \circ E_{\mathcal{X}} : \mathcal{X} \rightarrow \mathcal{X}$, which helps guide the training of destylization. Our objective is to solve the min-max problem:

$$\min_{E,G} \max_{D} \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{desty}} + \mathcal{L}_{\text{sty}}, \tag{1}$$

where $\mathcal{L}_{\text{rec}}$, $\mathcal{L}_{\text{desty}}$, and $\mathcal{L}_{\text{sty}}$ are loss functions related to the autoencoder reconstruction, destylization, and stylization, respectively. In the following sections, we present the detail of the loss functions and introduce our one-shot learning strategy that enables the training with only one example.

Autoencoder

Reconstruction loss. First of all, the encoded content feature is required to preserve the core information of the glyph. Therefore, we impose a reconstruction constraint that forces the content feature to completely reconstruct the input text image, leading to the standard autoencoder L_1 loss:

$$\mathcal{L}_{\text{rec}} = \lambda_{\text{rec}} \mathbb{E}_x[\|G_{\mathcal{X}}(E_{\mathcal{X}}(x)) - x\|_1]. \tag{2}$$

Destylization

In the training of our destylization subnetwork, we sample from the training set a text-style pair (x, y). We would like to map x and y onto a shared content feature space, where the feature can be used to reconstruct x. To achieve it, we apply two strategies: weight sharing and content feature guidance. First, as adopted in other domain transfer works (Liu, Breuel, and Kautz 2017; Murez et al. 2018), the weights between the last few layers of $E_{\mathcal{X}}$ and $E^c_{\mathcal{Y}}$ as well as the first few layers of $G_{\mathcal{X}}$ and $G_{\mathcal{Y}}$ are shared. Second, we propose a feature loss to guide $E^c_{\mathcal{Y}}$ using the content feature extracted by the autoencoder. The total loss takes the following form:

$$\mathcal{L}_{\text{desty}} = \lambda_{\text{dfeat}} \mathcal{L}_{\text{dfeat}} + \lambda_{\text{dpix}} \mathcal{L}_{\text{dpix}} + \lambda_{\text{dadv}} \mathcal{L}_{\text{dadv}}, \tag{3}$$

where $\mathcal{L}_{\text{dfeat}}$ is the feature loss. Following the image-to-image GAN framework (Isola et al. 2017), $\mathcal{L}_{\text{dfeat}}$ and $\mathcal{L}_{\text{dadv}}$ are pixel and adversarial losses, respectively.

Feature loss. The content encoder is tasked to approach the ground truth content feature. Let $S_{\mathcal{X}}$ denote the sharing layers of $G_{\mathcal{X}}$. Then the content feature for guidance is defined as $z = S_{\mathcal{X}}(E_{\mathcal{X}}(x))$ and our feature loss is:

$$\mathcal{L}_{\text{dfeat}} = \mathbb{E}_{x,y}[\|S_{\mathcal{X}}(E^c_{\mathcal{Y}}(y)) - z\|_1]. \tag{4}$$

Our feature loss guides the content encoder $E^c_{\mathcal{Y}}$ to remove the style elements from the text effects image, preserving only the core glyph information.

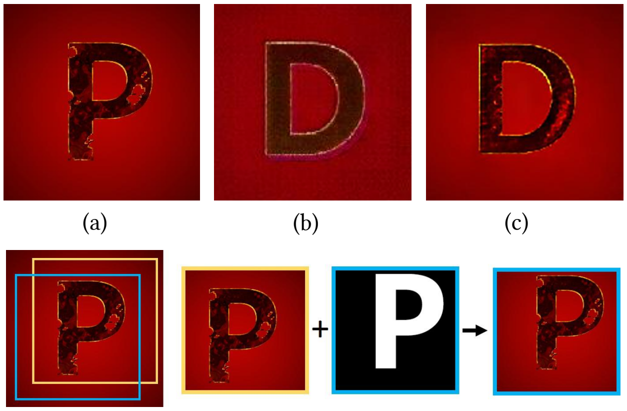
Pixel loss. The destylization subnetwork is tasked to approach the ground truth output in an L_1 sense:

$$\mathcal{L}_{\text{dpix}} = \mathbb{E}_{x,y}[\|G_{\mathcal{X}}(E^c_{\mathcal{Y}}(y)) - x\|_1]. \tag{5}$$

Adversarial loss. We impose conditional adversarial loss to imporve the quality of the generated results. We adopt a conditional version of WGAN-GP (Gulrajani et al. 2017) as our loss function, where $D_{\mathcal{X}}$ learns to determine the authenticity of the input text image and whether it matches the given text effects image. At the same time, $G_{\mathcal{X}}$ and $E^c_{\mathcal{Y}}$ learn to confuse $D_{\mathcal{X}}$:

$$\begin{aligned}\mathcal{L}_{\text{dadv}} =& \mathbb{E}_{x,y}[D_{\mathcal{X}}(x, y)] - \mathbb{E}_y[D_{\mathcal{X}}(G_{\mathcal{X}}(E^c_{\mathcal{Y}}(y)), y)] \\ &- \lambda_{gp} \mathbb{E}_{\hat{x},y}[(\|\nabla_{\hat{x}} D_{\mathcal{X}}(\hat{x}, y)\|_2 - 1)^2],\end{aligned} \tag{6}$$

where $\hat{x}$ is defined as a uniformly sampling along the straight line between the sampled real data x and the sampled generated data $G_{\mathcal{X}}(E^c_{\mathcal{Y}}(y))$.



(a) (b) (c)

(d) self-stylization training scheme

Figure 3: One-shot text effects transfer. (a) User-specified new text effects. (b) Stylization result on an unseen style. (c) Stylization result after one-shot finetuning. (d) Randomly crop the style image to generate image pairs for training.

Stylization

For the stylization subnetwork, we sample from the training set a text-style pair (x, y) and a text effects image y' that shares the same style with y but has a different glyph. We first extract the content feature from x and the style feature from y, which are then concatenated and fed into $G_{\mathcal{Y}}$ to generate a text effects image to approximate to the ground truth y. The standard image-to-image GAN loss is used:

$$\mathcal{L}_{\text{sty}} = \lambda_{\text{spix}}\mathcal{L}_{\text{spix}} + \lambda_{\text{sadv}}\mathcal{L}_{\text{sadv}}. \tag{7}$$

Pixel loss. The stylization subnetwork is tasked to approach the ground truth output in an L_1 sense:

$$\mathcal{L}_{\text{spix}} = \mathbb{E}_{x,y,y'}[\|G_{\mathcal{Y}}(E_{\mathcal{X}}(x), E^s_{\mathcal{Y}}(y')) - y\|_1]. \tag{8}$$

Adversarial loss. Similar to the destylization subnetwork, WGAN-GP (Gulrajani et al. 2017) is employed where the discriminator's decision is conditioned by both x and y':

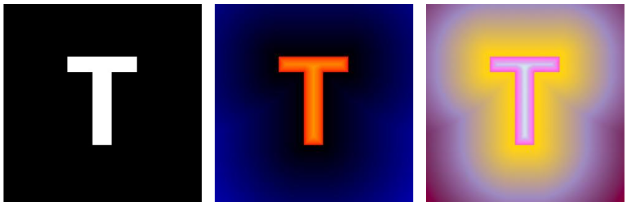
$$\begin{aligned}\mathcal{L}_{\text{sadv}} =& \mathbb{E}_{x,y,y'}[D_{\mathcal{Y}}(x, y, y')]\\ &- \mathbb{E}_{x,y'}[D_{\mathcal{Y}}(x, G_{\mathcal{Y}}(E_{\mathcal{X}}(x), E^s_{\mathcal{Y}}(y')), y')]\\ &- \lambda_{gp}\mathbb{E}_{x,\hat{y},y'}[(\|\nabla_{\hat{y}}D_{\mathcal{Y}}(x, \hat{y}, y')\|_2 - 1)^2].\end{aligned} \tag{9}$$

In the above equation, $\hat{y}$ is similarly defined as $\hat{x}$ in Eq. (6). Note that we do not impose any other constraints (for example, style autoencoder reconstruction: $\mathcal{Y} \to \mathcal{Y}$ and cycle consistency: $\mathcal{X} \times \mathcal{Y} \to \mathcal{Y} \to \mathcal{X}$) on the style feature extraction. We would like the network to learn appropriate style representations purely driven by the data. In fact, we have considered employing other common losses, but found the results have little changes, which proves that our objective design has been robust enough to learn a smart way of stylization from the data.

One-Shot Text Effects Transfer

Learning-based methods are heavily dependent on dataset by nature and usually require thousands of images for training. We have collected a text effects dataset, and our TET-GAN can be well trained using this dataset to generate as much as 64 different text effects. However, it is still quite expensive with respect to date collection of user-customized styles. To develop a system that supports personalized text effects transfer, we build upon our well-trained TET-GAN, and propose a novel "self-stylization" training scheme for one-shot learning, where only one training pair is required. Furthermore, we show that our network can be extended to solve a more challenging unsupervised problem where only one example style image is available.

Figure 4: An overview of our text effects dataset.



(a) (b) (c)

Figure 5: Distribution-aware data augmentation. (a) Raw text image. (b) Result of distribution-aware text image preprocessing. (c) Result of distribution-aware text effects augmentation by tinting (b) using random colormaps.

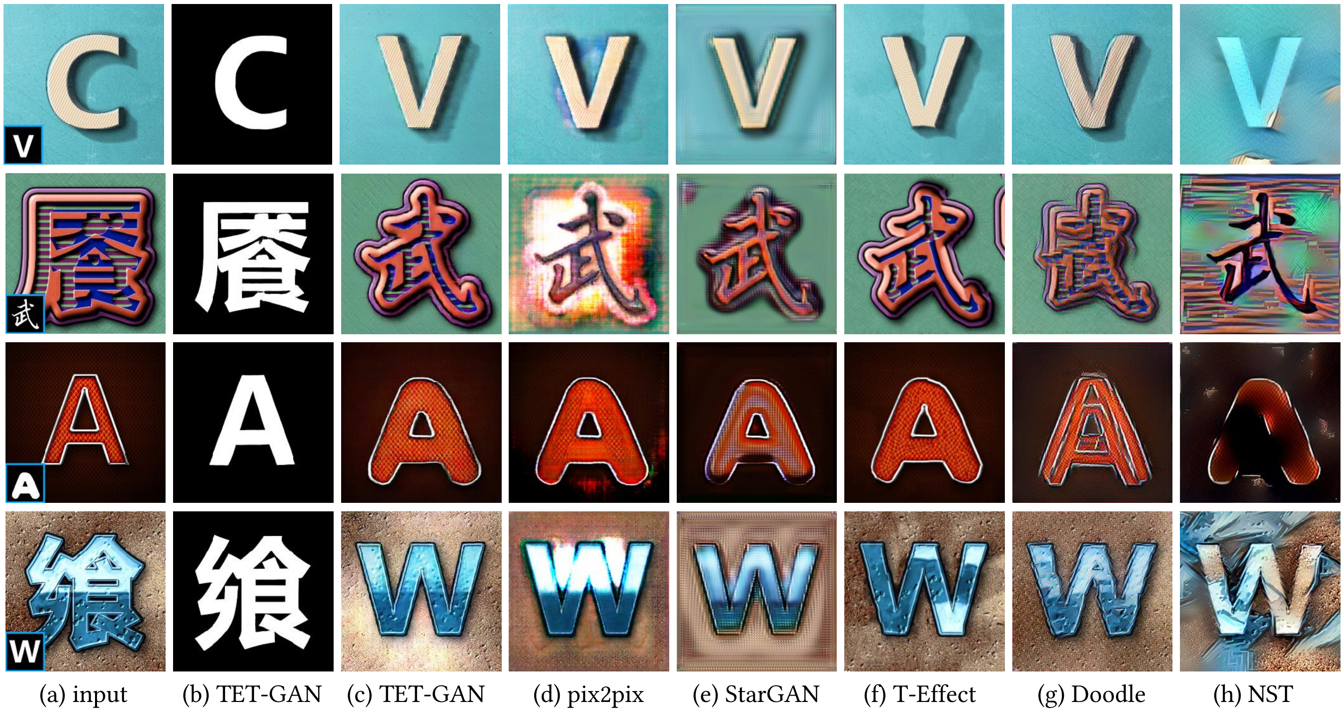
One-shot supervised learning. As shown in Fig. 3(b), for an unseen user-specified style, the network trained on our dataset has learned to generate the basic structure of the text effects. It only needs to be finetuned to better reconstruct the texture details of the specified style. To achieve this goal, we propose a simple and efficient "self-stylization" training scheme. Specifically, as shown in Fig. 3(d), we randomly crop the images to obtain a bunch of text effects images that have the same style but differ in the pixel domain. They constitute a training set to finetune our network to generate vivid textures as shown in Fig. 3(c).

One-shot unsupervised learning. Our network architecture gives us great flexibility. It is intuitive to exploit the destylization subnetwork to generate the text image from the new text effects image, and use this image pair for one-shot supervised learning. In other word, $\tilde{x} = G_{\mathcal{X}}(E^c_{\mathcal{Y}}(y))$ is used as an auxiliary x during the finetuning. However, the accuracy of $\tilde{x}$ cannot be guaranteed, which may mislead the extraction of content features. To solve this problem, a style autoencoder reconstruction loss is employed, which further constrains the content features to reconstruct the input text effects image with the style features:

$$\mathcal{L}_{\text{srec}} = \lambda_{\text{srec}}\mathbb{E}_y[\|G_{\mathcal{Y}}(E^c_{\mathcal{Y}}(y), E^s_{\mathcal{Y}}(y)) - y\|_1]. \tag{10}$$

And our objective for unsupervised learning takes the form

$$\min_{E,G}\max_{D} \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{desty}} + \mathcal{L}_{\text{sty}} + \mathcal{L}_{\text{srec}}. \tag{11}$$



(a) input (b) TET-GAN (c) TET-GAN (d) pix2pix (e) StarGAN (f) T-Effect (g) Doodle (h) NST

Figure 6: Comparison with state-of-the-art methods on various text effects. (a) Input example text effects with the target text in the lower-left corner. (b) Our destylization results. (c) Our stylization results. (d) pix2pix-cGAN (Isola et al. 2017). (e) StarGAN (Choi et al. 2018). (f) T-Effect (Yang et al. 2017). (g) Neural Doodles (Champandard 2016). (h) Neural Style Transfer (Gatys, Ecker, and Bethge 2016).

Distribution-Aware Data Collection and Augmentation

We propose a new dataset including 64 text effects each with 775 Chinese characters, 52 English letters and 10 Arabic numerals, where the first 708 Chinese characters are for training and others for testing. Fig. 4 shows an overview of these text effects. Each text effects image has a size of 320×320 and is provided with its corresponding text image. Our dataset contains two text effects kindly provided by the authors of (Yang et al. 2017). To generate other 62 text effects, we first collected psd files released by several text effects websites, or created psd files ourselves following the tutorials on these websites. Then we used batch tools and scripts to automatically replace characters and produced 837 text effects images for each psd file.

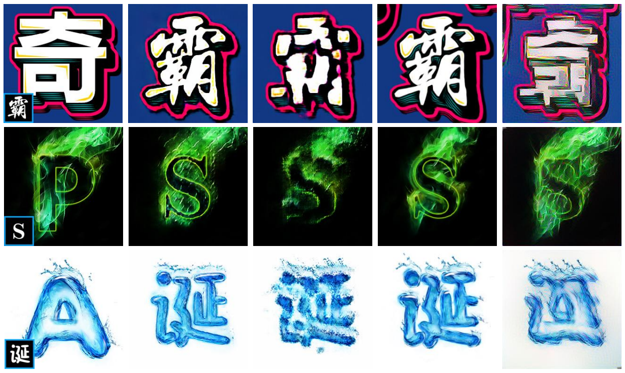
Distribution-aware text image preprocessing. As reported in (Yang et al. 2017), the spatial distribution of the texture in text effects is highly related to its distance from the glyph, forming an effective prior for text effects transfer. To leverage this prior, we propose a distribution-aware preprocessing for the text image to directly feed our network with distance cues. As shown in Fig. 5, we extend the raw text image from one channel to three channels. The R channel is the original text image, while G channel and B channel are distance maps where the value of each pixel is its distance to the background black region and the foreground white glyph, respectively. Another advantage of the preprocessing is that our three-channel text images have much fewer saturated areas than the original ones, which greatly facilitates the extraction of valid features.

Distribution-aware text effects augmentation. Besides the text images, we further propose the distribution-aware augmentation of the text effects images. The key idea is to augment our training data by generating random text effects based on the pixel distance from the glyph. Specifically, we first establish a random colormap for each of the R and G channels, which maps each distance value to a corresponding color. Then we use the colormaps of the R and G channels to tint the background black region and the foreground white glyph in the text image separately. Fig. 5(c) shows an example of the randomly generated text effects images. These images whose colors are distributed strictly according to distance can effectively guide our network to discover the spatial relationship between the text effects and the glyph. In addition, data augmentation could also increase the generalization capabilities of the network.

Experimental Results

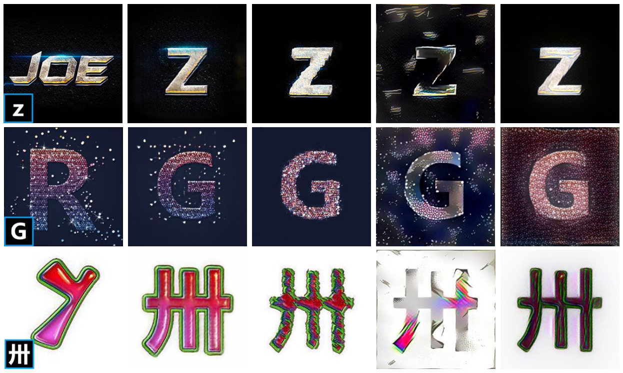
Implementation Details

Network architecture. We adapt our network architectures from pix2pix-cGAN (Isola et al. 2017). The three encoders use a same structure built with Convolution-BatchNorm-ReLU layers, and the decoders are built with Deconvolution-BatchNorm-LeakyReLU layers. The architecture of our two



(a) input (b) TET-GAN (c) TET-GAN2 (d) T-Effect (e) Doodle

Figure 7: Comparison with other methods on one-shot supervised style transfer. (a) Example text effects and the target text. (b) Our results. (c) Results of our network without pretraining. (d) T-Effect (Yang et al. 2017). (e) Neural Doodles (Champandard 2016)



(a) input (b) TET-GAN (c) Quilting (d) NST (e) CNNMRF

Figure 8: Comparison with other methods on one-shot unsupervised style transfer. (a) Example text effects and the target text. (b) Our results. (c) Image Quilting (Efros and Freeman 2001). (d) Neural Style Transfer (Gatys, Ecker, and Bethge 2016). (e) CNNMRF (Li and Wand 2016a)

discriminators follows PatchGAN (Isola et al. 2017). We add skip connections between the sharing layers of encoders and decoders so that they form a UNet (Ronneberger, Fischer, and Brox 2015). By doing so, our network can capture both low-level and high-level features. Considering Instance Normalization (IN) (Ulyanov, Vedaldi, and Lempitsky 2017) can better characterize the style of each image instance for a robust style removal than Batch Normalization (BN) (Ioffe and Szegedy 2015), we further replace BN with IN in $E^c_{\mathcal{Y}}$, which can effectively improve the destylization results.

Network training. We train the network on the proposed dataset. All images are cropped to 256×256 and one quarter of the training samples use the augmented text effects. To stabilize the training of GAN, we follow the very recent works of progressive training strategies (Karras et al. 2018). The inner layers of our generators are first trained on downsampled 64×64 images. Outer layers are then added progressively to increase the resolution of the generated images until the original resolution is reached. When new layers are added to the encoders, we fade them in smoothly to avoid drastic network changes (Karras et al. 2018). Adam optimizer is applied with a fixed learning rate of 0.0002 and a batch size of 32, 16 and 8 for image size of 64×64, 128×128 and 256×256, respectively. For all experiments, we set $\lambda_{\text{dfeat}} = \lambda_{\text{dpix}} = \lambda_{\text{spix}} = \lambda_{\text{rec}} = \lambda_{\text{srec}} = 100$, $\lambda_{gp} = 10$, and $\lambda_{\text{dadv}} = \lambda_{\text{sadv}} = 1$.

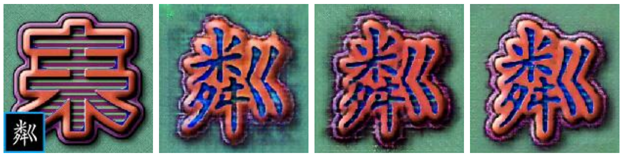
Comparison with State-of-the-Art

In Fig. 6, we present a comparison of our network with five state-of-the-art style transfer methods. The first two methods, pix2pix-cGAN and StarGAN, both employ GAN for domain transfer, and can be trained on our dataset to handle text effects. To order to allow pix2pix-cGAN to handle multiple styles, we change its input from a single text image to a concatenation of three images: the example text effects image, its corresponding glyph and the target glyph. Pix2pix-cGAN fails to completely adapt the style image to the new glyphs, creating some ghosting artifacts. Meanwhile, the texture details are not fully inferred, leaving some flat or over-saturated regions. StarGAN learns some color mappings, but fails to synthesize texture details and suffers from distinct checkerboard artifacts. The following three methods are designed for zero-shot style transfer. T-Effect and Neural Doodles synthesize textures using local patches under the glyph guidance of the example image. T-Effect processes patches in the pixel domain, leading to obvious color discontinuity. Instead, Neural Doodles uses deep-based patches for better patch fusion but fails to preserve the shape of the text. Neural Style Transfer cannot correctly find the correspondence between the texture and text, creating interwoven textures. By comparison, our network learns valid glyph features and style features, thus precisely transferring text effects with the glyph well protected. We additionally show our destylization results in the second column, where the style features are effectively removed.

We compare our network with T-Effect and Neural Doodles on supervised stylization with only one observed example pair in Fig. 7. Our method is superior to Neural Doodles in glyph preservation and is comparable to T-Effect. More importantly, in terms of efficiency, T-Effect takes about one minute per image, while our method only takes about 20ms per image after a three-minute finetuning. In addition, as shown in Fig. 7(c), if trained from scratch, the performance of our network drops dramatically, verifying that pretraining on our dataset successfully teaches our network the domain knowledge of text effects synthesis. In Fig. 8, we further compare with three methods on the challenging unsupervised stylization with only one observed example, where the advantages of our approach are more pronounced.

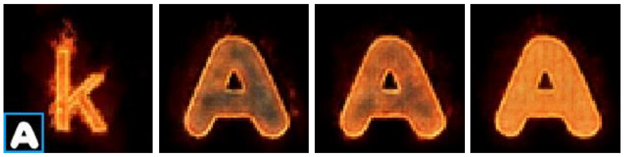
Ablation Study

In Fig. 9, we study the effect of the reconstruction loss (Eq. (4)) and the feature loss (Eq. (2)). Without these two losses, even the color palette of the example style is not cor-



(a) input (b) (c) (d)

Figure 9: Effect of the reconstruction loss and feature loss. (a) Input. (b) Model without $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{dfeat}}$. (c) Model without $\mathcal{L}_{\text{dfeat}}$. (d) Full model.



(a) input (b) w/o pre+aug (c) w/o aug (d) full model

Figure 10: A comparison of results with and without our distribution-aware data preprocessing and augmentation.

rectly transferred. In Fig. 9(c), the glyph is not fully disentangled from the style, leading to annoying bleeding artifacts. The satisfying results in Fig. 9(d) verify that our feature loss effectively guides TET-GAN to extract valid content representations to synthesize clean text effects.

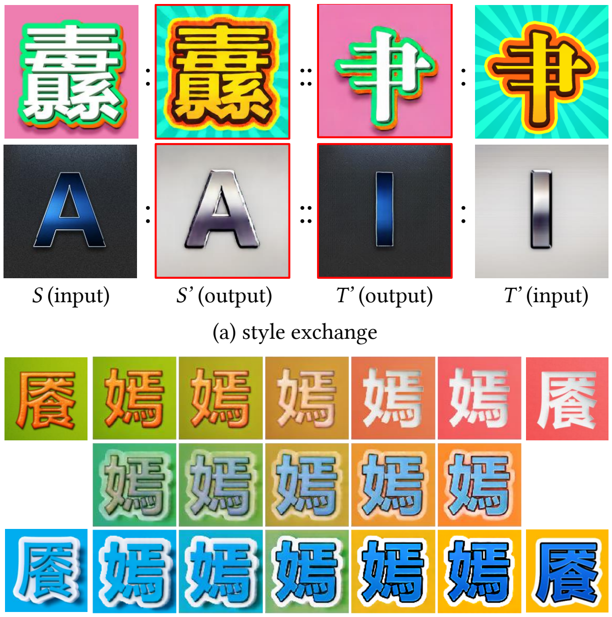
In Fig. 10, we examine the effect of our distribution-aware text image preprocessing and text effects augmentation through a comparative experiment. Without the preprocessing and augmentation, the inner flame textures are not synthesized correctly. As can be seen in Fig. 10(d), our distribution-aware data augmentation strategy helps the network learn to infer textures based on their correlated position on the glyph, and thus the problem is well solved.

Application

The flexibility of TET-GAN is further manifested by two applications: style exchange and style interpolation. First, we can exchange the styles from two text effects images as shown in Fig. 11(a). It is accomplished by extracting the glyphs using the destylization subnetwork and then applying the styles to each other using the stylization subnetwork. Second, the explicit style representations enable intelligent style editing. Fig. 11(b) shows an example of style fusion. We interpolate between four different style features, and decode the integrated features back to the image space, obtaining brand-new text effects.

Failure Case

While our approach has generated appealing results, some limitations still exist. Our destylization subnetwork is not fool-proof due to the extreme diversity of the text effects, which may totally differ from our collected text effects. Fig. 12 shows a failure case of one-shot unsupervised text effects transfer. Our network fails to recognize the glyph. As a result, in the stylization result, the text effects in the foreground and background are reversed. This problem can be possibly solved by user interaction. Users can simply paint a few strokes (Fig. 12(d)) to provide a priori information about the foreground and the background, which is then fed into the network as a guidance to constrain the glyph extraction and thereby improve the style transfer results (Fig. 12(f)).

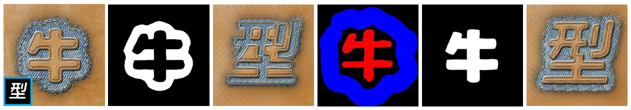


S (input) *S'* (output) *T'* (output) *T'* (input)

(a) style exchange

(b) style interpolation

Figure 11: Applications of TET-GAN.



(a) (b) (c) (d) (e) (f)

Figure 12: User-interactive unsupervised style transfer. (a) Example text effects and the target text. (b)(c) Our destylization and stylization results after finetuning. (d) A mask provided by the user, where the blue and red regions indicate the background and foreground, respectively. (e)(f) Our destylization and stylization results with the help of the mask.

Conclusion

In this paper, we present a novel TET-GAN for text effects transfer. We integrate stylization and destylization into one uniform framework to jointly learn valid content and style representations of the artistic text. Exploiting explicit style and content representations, TET-GAN is able to transfer, remove and edit dozens of styles, and can be easily customized with user-specified text effects. In addition, we develop a dataset of professionally designed text effects to facilitate researches. Experimental results demonstrate the superiority of TET-GAN in generating high-quality artistic typography. As a future direction, one may explore other more sophisticated style editing methods, such as background replacement, color adjustment and texture attribute editing.

References

Azadi, S.; Fisher, M.; Kim, V.; Wang, Z.; Shechtman, E.; and Darrell, T. 2018. Multi-content gan for few-shot font style transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*.

Champandard, A. J. 2016. Semantic style transfer and turning two-bit doodles into fine artworks. arXiv:1603.01768.

Chen, T. Q., and Schmidt, M. 2016. Fast patch-based style transfer of arbitrary style. arXiv:1612.04337.

Choi, Y.; Choi, M.; Kim, M.; Ha, J. W.; Kim, S.; and Choo, J. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*.

Dumoulin, V.; Shlens, J.; and Kudlur, M. 2016. A learned representation for artistic style. In *Proc. Int'l Conf. Learning Representations*.

Efros, A. A., and Freeman, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proc. ACM Conf. Computer Graphics and Interactive Techniques*, 341–346.

Gatys, L. A.; Ecker, A. S.; Bethge, M.; Hertzmann, A.; and Shechtman, E. 2017. Controlling perceptual factors in neural style transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, 262–270.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2414–2423.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, 5767–5777.

Huang, X., and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. Int'l Conf. Computer Vision*, 1510–1519.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167.

Isola, P.; Zhu, J. Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 5967–5976.

Johnson, J.; Alahi, A.; and Li, F. F. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Proc. European Conf. Computer Vision*, 694–711.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2018. Progressive growing of gans for improved quality, stability, and variation. In *Proc. Int'l Conf. Learning Representations*.

Li, C., and Wand, M. 2016a. Combining markov random fields and convolutional neural networks for image synthesis. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2479–2486.

Li, C., and Wand, M. 2016b. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proc. European Conf. Computer Vision*, 702–716.

Li, Y.; Wang, N.; Liu, J.; and Hou, X. 2017a. Demystifying neural style transfer. In *Int'l Joint Conf. Artificial Intelligence*, 2230–2236.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M. H. 2017b. Diversified texture synthesis with feed-forward networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017c. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, 386–396.

Lian, Z.; Zhao, B.; and Xiao, J. 2016. Automatic generation of large-scale handwriting fonts via style learning. In *SIGGRAPH ASIA 2016 Technical Briefs*, 12:1–12:4. ACM.

Liu, M. Y.; Breuel, T.; and Kautz, J. 2017. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, 700–708.

Murez, Z.; Kolouri, S.; Kriegman, D.; Ramamoorthi, R.; and Kim, K. 2018. Image to image translation for domain adaptation. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 4500–4509.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proc. Int'l Conf. Medical Image Computing and Computer-Assisted Intervention*, 234–241.

Sun, D.; Ren, T.; Li, C.; Su, H.; and Zhu, J. 2018. Learning to write stylized chinese characters by reading a handful of examples. In *Int'l Joint Conf. Artificial Intelligence*.

Ulyanov, D.; Lebedev, V.; Vedaldi, A.; and Lempitsky, V. 2016. Texture networks: feed-forward synthesis of textures and stylized images. In *Proc. IEEE Int'l Conf. Machine Learning*, 1349–1357.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2017. Instance normalization: The missing ingredient for fast stylization. arXiv:1704.00028.

Wang, X.; Oxholm, G.; Zhang, D.; and Wang, Y. F. 2017. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*.

Yang, S.; Liu, J.; Lian, Z.; and Guo, Z. 2017. Awesome typography: Statistics-based text effects transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 7464–7473.

Zhang, Y.; Zhang, Y.; and Cai, W. 2018. Separating style and content for generalized style transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*.

Zhu, J. Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. Int'l Conf. Computer Vision*, 2242–2251.