Font Style Translation in Scene Text Images with CLIPstyler

Honghui Yuan[0009−0001−4334−9363] and Keiji Yanai[0000−0002−0431−183X]

The University of Electro-Communications, Chofu, Tokyo, JAPAN
{yuan-h, yanai}@mm.inf.uec.ac.jp

Fig. 1. The proposed method can transfer the style of text regions in images with prompts.

Abstract. Scene text editing is widely used in various fields, such as poster design and correcting spelling mistakes in the image. Editing text in images is a challenging task that requires accurately and naturally integrating text within complex backgrounds. Existing methods have achieved changing the text content with the target text without altering the style of text and the background of the image. However, arbitrary style transformation of the text region in the image has not been achieved. To address this issue, we propose a new framework named FontCLIPstyler, which enables the style transformation of text in scene text images using prompts. The proposed method mainly comprises two networks: MaskNet, which extracts mask images of the text region in images, and StyleNet, which performs the generation of stylized images. In addition, we also propose a new loss function named Text-aware Loss, which can guide the StyleNet network in transferring style features to the text region without changing the background. Through extensive experiments and ablation studies, we have demonstrated the effectiveness of our method in scene text style transformation. The experimental results show that our approach can successfully transfer the semantic style from the input prompt to the text region of the image, and create naturally stylized scene text while keeping the readability of the text and the background invariant.

Keywords: Image style transfer · Font translation · Scene text images · Arbitrary style · CLIPstyler.

1 Introduction

In recent years, the development of deep learning has significantly enhanced the convenience of image editing. Many studies have achieved significant results on scene text editing using GAN(Generative Adversarial Networks) [9] and the diffusion model [26]. The previous methods of scene text editing focused on replacing the text content within the image while preserving the background and the text style (color, texture, font, etc.). However, these methods are limited to text content replacement and cannot transform text style arbitrarily. Therefore, as shown in Fig.1, we propose a task focused on scene text style transformation, aiming to translate the style of the text region without modifying the text content and the background in the image. In general, scene text editing is divided into three sub-tasks including background restoration, text conversion, and image re-synthesize. However, this process relies on using the original image as a style reference, ensuring that the generated image maintains the same text style and background as the original image. Recently, with the advance of the diffusion [26] model, many methods based on the diffusion model can generate natural and high-quality scene text images. However, when dealing with editing the text region of the image, these models could only transfer text style based on the style of other text in the same image, thus lacking arbitrary style transformation ability.

Image style transformation usually refers to generating new images by integrating the content of one image with the style of another image. A lot of methods [5], [38] have successfully generated attractive results with the GAN and transformer [32] models. However, sometimes it is hard to find the desired style images as the style reference image of style transfer. Recently, the CLIP [25] model, a multi-modal model of language and images, has been utilized to guide image generation through prompts. CLIPstyler [20] utilizes a lightweight CNN network and the CLIP model to achieve the image style transformation through the simple text description, without the need for style reference images. Also, the main content of the original image can be effectively maintained unchanged while applying the style transformation. Different from the style transformation of normal images, the style transformation of text images is more complex because it is necessary to ensure the readability of the text while transferring the style features to the text. Many studies [37], [34] have focused on the style transformation of text images, which usually utilize the network that uses text images to learn the text structure so that the content of the text will be maintained. However, these studies are usually limited to images of single characters, and cannot handle images of multiple characters such as words.

Therefore, we propose a new framework based on CLIPstyler [20] named FontCLIPstyler, to achieve the style transformation of text region in the scene text image without changing the image background and text content. The proposed method enables style transformation using prompts and enables arbitrary style transfer. Our main contributions are as follows:

1. We propose a style transfer framework for scene text images that can effectively transfer the style of the text region in the image while keeping the background and text content unchanged.
2. Our method uses prompts to control the style without the style reference image, achieving arbitrary style transformation of the scene text image.
3. The experiments have proved that our method can generate visually attractive styled scene text and successfully achieve the scene text style transfer task.

2 Related Work

2.1 Scene text editing

Scene text editing has made significant progress in replacing text in the original image with another text while preserving the style of the text. STEFANN [28] designed two networks for font structure transformation and color transformation. However, it is replacing one character of text at a time and is unable to vary the number of characters different from the original text. SRNet [36] achieved text replacement using three sub-networks: background restoration, text conversion, and image re-synthesis. SwapText [39] incorporates a TPS(Thin-Plate-Spline) module and utilizes spatial points to geometrically transform text based on SRNet. SimAN [22] implemented similarity-aware normalization and uses a self-supervised learning method to train the network. Based on StyleGAN [18], TextStyleBrush [19] integrates style vectors of the text image into the generator to guide the generation of final images. Mostel [24] added additional stroke-level information to the network and used synthetic and real-world data to significantly improve scene text editing performance. However, these methods require the style reference image to achieve style transformation of text. Recently, the diffusion model [26] had great success in image generation and editing. DiffSTE [16], DiffUTE [3], GlyphDraw [23] GlyphControl [42], TextDiffuser [4], and some other methods based on the diffusion model have achieved high-quality scene text generation and editing. However, they do not offer control over the style of the text. Different from the methods mentioned above, our method does not require the style reference image for style transfer and allows users to specify the scene text style by prompts.

2.2 Style transfer

Image style transformation aims to transfer the style from a reference image to a target image. Neural image style transfer [8] utilizes a CNN-based network to achieve image style transfer based on style images. AdaIN [13] using adaptive instance normalization to align the mean and variance of the content image's features with those of the style image, enabling arbitrary style transformations. In recent years, based on GAN [9] and Tansformer [32], methods such as StyleGAN [18], StyTr2 [5] have achieved significant success in generating high-quality

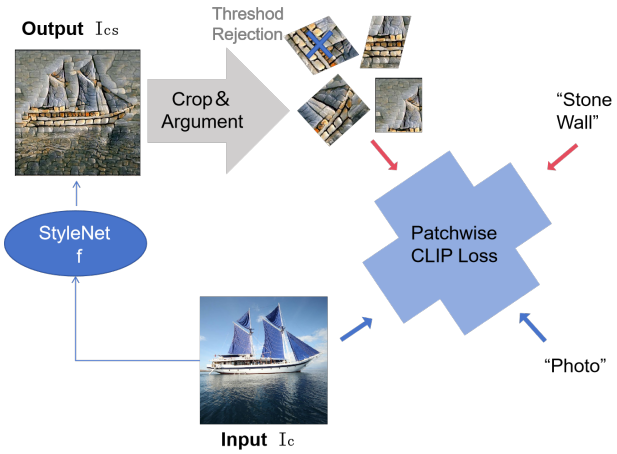

Fig. 2. Overview of CLIPstyler.

style images. However, these methods require the style reference image to get the style features. By learning models from a dataset consisting of 4 billion pairs of images and text, Recent research CLIP [25] can improve zero-shot performance for many downstream tasks. As shown in Fig.2, Based on the CLIP [25] model, CLIPstyler [20] has solved the problem that needs the reference style image to transfer the style of images and allowing arbitrary style transformations with prompts. Sem-CS [17] and Gen-Art [43] used semantic segmentation to solve the over-stylization problem of the foreground portion in CLIPstyler [20]. However, these methods perform style transformations on the entire image and are unable to transfer the style to specific parts of the image.

Text image style transformation methods have been widely researched due to the success of image style transfer. MCGAN [2] focused on the English alphabet, generating the remaining alphabet in the same style based on a few alphabet examples. TETGAN [40] enables style transfer from one character to others via a stylization subnetwork and a de-stylization subnetwork. FETGAN [21] enables the generation of new characters in the same style with only a few artistic characters, supporting both the English alphabet and more complex Chinese characters. Typography with Decor [35] proposed a novel deep-learning network to achieve the style transfer of characters that include decoration. Shape MatchingGAN [41] allows converting text styles by using just one style reference image. Multi-Style Shape Matching GAN [12] used a single model to achieve multiple styles generation of text images based on Shape MatchingGAN. However, these methods depend on style images or other text images as the style reference images. More recently, Word as Image [14], CLIPFont [29], DS-Fusion [31], Zero-shot Font Style Transfer [15] has achieved font style transfer using the prompt without the need for style images. However, these methods are limited to single-character images and struggle with generating satisfactory results for multiple characters, such as words. Our method is capable of transferring the style of text regions in scene text images through prompts and can preserve the readability of text unrestricted to the number of characters.

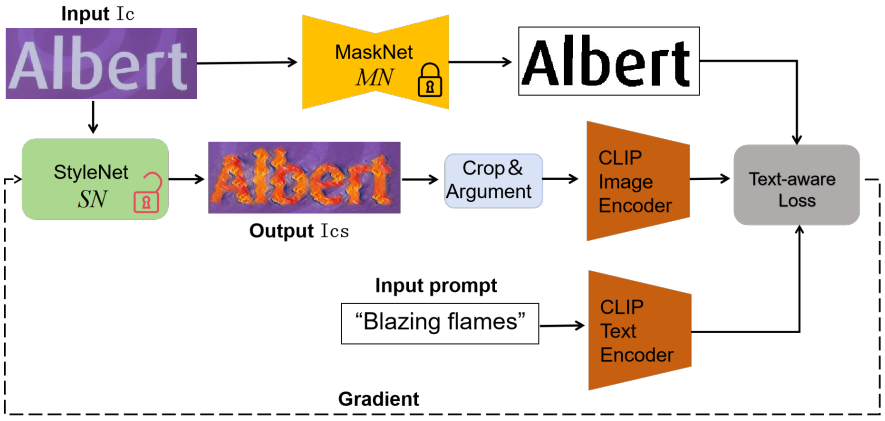

Fig. 3. The framework of our method FontCLIPstyler.

3 Methodology

In this study, we propose a new framework based on CLIPstyler [20] named FontCLIPstyler to achieve style transformation of the text region in scene images. An overview of our proposed method is shown in Fig.3. The network mainly consists of two networks, the MaskNet network(MN) that extracts the mask image of the text region in the scene text image, and the StyleNet network(SN) that conducts style transformation of the input image. By using the pre-trained text-image embedding model CLIP [25] and the Text-aware Loss proposed in this study, the parameters of the network SN are optimized to transfer the semantic style from input prompts to the generated image.

3.1 Basic Framework CLIPstyler

Firstly, we will start with an introduction to CLIPstyler, which has been used as the base model for our method. As shown in Fig.2, CLIPstyler aims to transfer the semantic style features from the input prompt to the input image. Since the style is expressed in the form of natural language, it does not require the style image as a reference to get style features. Without the constraints of reference style images, which are sometimes difficult to obtain, arbitrary style transformations can be realized through imaginative prompts. Specifically, the input image I_c is fed into StyleNet f which is the encoder-decoder CNN, and the parameters of f are optimized by Patchwise CLIPLoss to transfer style features from prompts to the image and generate the stylized image I_{cs}. When calculating the loss function, the generated images are randomly cropped and augmented to achieve more vivid and diverse textures. The loss function used in the CLIPstyler can be formed as follows.

$$L_{total} = \lambda_d L_{dir} + \lambda_p L_{patch} + \lambda_c L_c + \lambda_{tv} L_{tv} \tag{1}$$

The Directional CLIPLoss(L_{dir}) proposed by StyleGAN-NADA [7] be applied to guide the output image rendering with the semantic style of the target prompt. As follows, Directional CLIPLoss encodes the input image I_c, input

prompt t_{style}, output image $f(I_c)$ and content prompt t_{src} by CLIP Encoder. By aligning the direction between these features, the generated images have the same semantic style as the input prompts.

$$\begin{aligned} \Delta T &= E_T(t_{style}) - E_T(t_{src}), \\ \Delta I &= E_I(f(I_c)) - E_I(I_c), \\ L_{dir} &= 1 - \frac{\Delta I \cdot \Delta T}{|\Delta I|\,|\Delta T|} \end{aligned} \tag{2}$$

Patch CLIPLoss(L_{patch}) proposed by CLIPstyler [20] using the randomly cropped patches $\hat{I}^i_{cs}$ to calculate Directional CLIPLoss instead of the entire generated image to get more high-quality images. In addition, random geometrical augmentation was applied to the cropped patches before calculating the Directional CLIPLoss and achieved more vivid and diverse textures. Furthermore, to prevent over-stylization of the image, a specific threshold τ is set to reject patches that are below this threshold. To retain the content of the original image in the generated image, use the VGG-19 network to calculate the content loss L_c. Furthermore, CLIPstyler also utilized total variation regularization loss (L_{tv}), which reduces the side artifacts caused by irregular pixels in the image. The calculation of Patch CLIPLoss is as follows.

$$\begin{aligned} \Delta T &= E_T(t_{style}) - E_T(t_{src}), \\ \Delta I &= E_I(aug(\hat{I}^i_{cs})) - E_I(I_c), \\ L^i_{patch} &= 1 - \frac{\Delta I \cdot \Delta T}{|\Delta I|\,|\Delta T|}, \\ L_{patch} &= \frac{1}{N}\sum_{i}^{n} R(l^i_{patch}, \tau) \\ where\ R(s,\tau) &= \begin{cases} 0, & \text{if } s \le \tau \\ s, & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

The calculation of the total variation regularization loss L_{tv} is as follows. This function calculates the sum of squared gradient differences of the pixel values in horizontal, vertical, and two diagonal directions for the input image x of size $H * W$ across the three channels c.

$$\begin{aligned} L_{tv} = &\sum_{c=1}^{3}\sum_{i=1}^{H}\sum_{j=1}^{W-1}(x_{i,j,c} - x_{i,j+1,c})^2 + \sum_{c=1}^{3}\sum_{i=1}^{H-1}\sum_{j=1}^{W}(x_{i,j,c} - x_{i+1,j,c})^2. \\ &+ \sum_{c=1}^{3}\sum_{i=1}^{H-1}\sum_{j=1}^{W-1}(x_{i+1,j,c} - x_{i,j+1,c})^2 + \sum_{c=1}^{3}\sum_{i=1}^{H-1}\sum_{j=1}^{W-1}(x_{i,j,c} - x_{i+1,j+1,c})^2 \end{aligned} \tag{4}$$

3.2 FontCLIPstyler

CLIPstyler [20] addresses the problem that needs style images as references in image style transformation, making the process more convenient. By using prompts

as guidance, the model has enabled arbitrary style transformations. However, CLIPstyler applies style transformations to the entire image and cannot target specific regions within the image for style transformation. To overcome this limitation, we propose the FontCLIPstyler framework based on CLIPstyler, which focuses on the style transformation of text region in the scene text image. In the following, we will provide a detailed description of the proposed framework and the loss function.

MaskNet and StyleNet Our framework is mainly composed of two networks: MaskNet and StyleNet. For editing an image in a specific area without impacting other regions, a straightforward and effective approach is to employ the mask image to delineate the targeted area. Thus, we introduce the MaskNet network designed to extract the mask image of the text region in scene images. The mask image serves as guidance for the method, enabling the style transfer to the text region of a scene text image while keeping the image background and text content unchanged. Specifically, because of the high computational efficiency and real-time image processing capability of U-Net [27], it is used as the backbone of the proposed MaskNet network. We train the MaskNet with real-world scene text images, and during the style transfer process, it is frozen to generate the mask image of the specified text region. In addition, with reference to CLIPStyler, we employ a CNN encoder-decoder network StyleNet for the style transfer. During training, the parameters of StyleNet are optimized with our proposed loss function Text-aware Loss, allowing for the generation of styled scene text images with the input prompt.

Text-aware Loss We propose a new loss function named Text-aware Loss that could control the StyleNet network to realize style transformation into the text region in the image while maintaining the background invariably and the readability of the text. Our proposed loss is mainly composed of three parts: Distance loss, TextPatch CLIPLoss, and Background reconstruction loss. Distance Transform loss [1] allows style transformation within a limited region based on distance transformation of the input image. Therefore, to transform the style in the text region of the scene text image, we utilize Distance Transform loss in this study. The loss function can be defined as follows. Specifically, a distance transform map I_d is created using the mask image obtained from the proposed MaskNet network. The distance transform assigns to each pixel $I_d^{(i,j)}$ a value that represents the distance to the nearest target region pixel. Using the Euclidean distance metric, this transformation calculates the distance between each pixel of the distance transform map $p_{i,j}$ and the pixel of mask image $p_{x,y}$. Pixels that are part of the target region are assigned a value of zero. As the pixels get further from the target, their assigned values, which represent their distance from the nearest target pixel, increase. Then, the input image I_c and stylized output image I_{sty} from StyleNet will multiplied by I_d respectively, and the mean squared error is calculated.

$$I_d^{(i,j)} = \min_{x,y \in mask} ||p_{i,j} - p_{x,y}||_2 \tag{5}$$

$$L_{distance} = \frac{1}{2}\sum_{i,j}(I_c^{(i,j)} \cdot I_d^{(i,j)} - I_{sty}^{(i,j)} \cdot I_d^{(i,j)})^2 \tag{6}$$

When utilizing prompts to control the style transfer, it is necessary to use Patch CLIPLoss to reflect the semantic style of the input prompt into the image. However, in CLIPstyler, Patch CLIPLoss randomly crops patches from the image to be stylized and tends to include background areas that do not require style transformation for our task. As a result, style features are reflected in the background resulting in background changes. To solve this problem, we propose a new loss function, TextPatch CLIPLoss. Specifically, the patches of the background region $I^i_{b\_patch}$ for the input image I_c are cropped using the mask image($Mask$) obtained from MaskNet(MN). The cosine similarity (sim) with the patches of the generated image and background region image $I^i_{b\_patch}$ is calculated, as a standard for determining the regions. Specifically, we utilize the image encoder of CLIP(E_I) to encode the generated image $\hat{I}^i_{sty}$ and the background region image $I^i_{b\_patch}$ of N patches. Then, we set a threshold μ to determine whether the generated image belongs to the background or text region. Patches with significantly different similarities are determined to belong to the text region $\hat{I}^i_{sty\_t}$. Then Patch CLIPLoss is calculated as CLIPstyler for patches belonging to the text region to render the style features using CLIP text encoder(E_T) and CLIP image encoder(E_I). In addition, when calculating Patch CLIPLoss, CLIPstyler sets a threshold τ to prevent the image from being over-stylization and get a better result in preserving the main content of images. However, it is more difficult to render style features into text regions in scene text images than usual images because the font typically consists of elongated structures, and showing visible and attractive style features is more difficult. Therefore, the threshold used to avoid over-stylization makes it difficult to reflect style features into the text region in scene images. To better render the style features into the text, we used all the patches in our loss function. L_{patch} is calculated as follows:

$$\begin{aligned}
&Mask = MN(I_c),\\
&I^i_{b_patch} = crop((1 - Mask) \odot I_c),\\
&sim = 1 - \frac{E_I(\hat{I}^i_{sty}) \cdot (\frac{1}{N}\sum_i^n E_I(I^i_{b_patch}))}{\left|E_I(\hat{I}^i_{sty})\right| \left|\frac{1}{N}\sum_i^n E_I(I^i_{b_patch})\right|},\\
&\hat{I}^i_{sty} = \begin{cases} \hat{I}^i_{sty_b} & if \quad sim < \mu \\ \hat{I}^i_{sty_t} & otherwise \end{cases}
\end{aligned} \tag{7}$$

$$\begin{aligned}
\Delta T &= E_T(t_{style}) - E_T(t_{src}),\\
\Delta I &= E_I(aug(\hat{I}^i_{sty_t})) - E_I(I_c),\\
L^i_{patch} &= 1 - \frac{\Delta I \cdot \Delta T}{|\Delta I|\,|\Delta T|},\\
L_{patch} &= \frac{1}{N}\sum_i^n l^i_{patch}
\end{aligned} \tag{8}$$

The content loss L_c is used in CLIPStyler to retain the main contents of the generated image unchanged from the original image. However, it will lead to the text style of the original image being reflected in the generated image. Therefore, to reduce the influence of the text style from the original image and make sure the content of the background is unchanged, instead of using the content loss L_c, we utilized a background reconstruction loss L_{recon}. Specifically, VGG19 loss is calculated as follows between the N patches of the generated image belonging to the background region $\hat{I}^i_{sty\_b}$ and the cropped patches from the input image I^i_c. Here F_{4_2} and F_{5_2} represent the convolution layer conv4_2 and conv5_2 of the VGG19 network, respectively.

$$\begin{aligned} L^i_{recon} &= \left\| F_{4\_2}(\hat{I}^i_{sty\_b}) - F_{4\_2}(I^i_c) \right\|^2_2 \\ &\quad + \left\| F_{5\_2}(\hat{I}^i_{sty\_b}) - F_{5\_2}(I^i_c) \right\|^2_2 \\ L_{recon} &= \frac{1}{N} \sum_i^n l^i_{recon} \end{aligned} \tag{9}$$

The loss function of the proposed Text-aware Loss L_{ta} is as follows:

$$L_{ta} = \lambda_d L_{distance} + \lambda_p L_{patch} + \lambda_r L_{recon} \tag{10}$$

where λ represents the weight of each loss function.

We also utilized the total variation regularization loss L_{tv} to reduce the side artifacts caused by irregular pixels in the image as CLIPstyler. Thus, the total loss function is as follows:

$$L_{total} = L_{ta} + \lambda_{tv} L_{tv} \tag{11}$$

Fig. 4. The results of different style transformations on the same scene text images using the proposed method.

4 Experiment

Implementation Details The proposed MaskNet network was trained using 2000 real-world scene text images collected from Mostel [24] and we randomly

selected 200 other synthetic and real-world images to test the MaskNet. The network of MaskNet consists of four downsample and upsample layers, with 64, 128, 256, and 512 channels respectively, followed by max-pooling. And two bottleneck layers with 1024 channels, the final layer is a sigmoid function. When conducting style transformation, MaskNet is frozen and the StyleNet network is optimized with the loss function without training. We conducted experiments with real scene text images that were randomly selected from the COCOText V2.0 dataset [33]. As for the network of StyleNet, We utilize a lightweight U-net [27] architecture featuring three down-sampling and three up-sampling layers. The channel sizes for each down-sampling layer are 16, 32, and 64, and the sigmoid function at the last layer. The input scene text images are converted to 512×512 size and the final output result will be resized to the original size. We set λ_d, λ_p, λ_r and λ_{tv} to 1×10^2, 9×10^3, 150 and 2×10^{-3}. The model is trained using a learning rate of 5×10^{-4} and Adam optimizer. Training iteration is set to 500 and the learning rate is halved every 100 iterations. We used a single NVIDIA TITAN RTX to train the model, and the training time per image was approximately 90 to 120 seconds.

Fig. 5. The results of the proposed method with various prompts and different scene text images.

4.1 Evaluation

We conducted experiments under various conditions to verify the effectiveness of the proposed method in generating stylized scene text images. Specifically, we have experimented with our method in various prompts and different scene text images. As shown in Fig.4, we confirm the ability of our method in scene text style transfer by using the same image with different prompts. The proposed method successfully reflects the semantic style specified by the prompt to the text region, without changing the background and maintaining the readability of the original text content. Additionally, Fig.5 shows more results from our

Fig. 6. The results of style transformation using the proposed method for synthesized scene text images.

Fig. 7. The results of mask images using the proposed MaskNet.

experiments. Regardless of the length as well as the size of the text, our method effectively rendered the style features into the text. These results demonstrate the effectiveness of our proposed method in arbitrary style transformation of scene text images using prompts. To confirm that the proposed method can achieve style transformation of text even on complex backgrounds, experiments were conducted using high-resolution synthetic scene text images. As shown in Fig.6, our method generated high-quality stylized text while preserving the fine texture of the background. Fig.7 shows the test results of our MaskNet. The results demonstrate that our network can effectively extract the masks of the text parts in various scenes.

4.2 Ablation Studies

We conducted ablation studies to verify the effectiveness of each component of the proposed Text-aware Loss. As shown in Fig.8, without Distance Transform loss, the style features render over the entire image, and the text content is unidentifiable. The result has failed in the style transformation to the text area. If TextPatch CLIPLoss is not applied, the semantic style from the input prompts is not reflected well in the image. When background reconstruction loss is not utilized, the features of the background will affect the text causing the boundary between text and background regions to become unclear and making text content difficult to identify. The model with all loss functions was able to transfer style features in the text region while preserving the background and text content. The quantitative evaluation results are shown in Tab 1. Without the $L_{distance}$ and L_{recon} results in a decrease in each score, and without L_{patch} leads to poor

qualitative results. Our full model achieves the highest scores in both NIMA [30] and CLIP scores [10], with the remaining scores being in second place. This demonstrates the effectiveness of each component of our model in generating aesthetic images and images that are consistent with the given prompts.

Table 1. Quantitative evaluation results for ablation studies.

	DISTS↓	NIMA↑	LPIPS↓	FID↓	CLIP SCORE↑
w/o Ldistance	0.4997	4.5776	0.6082	910.50	0.2428
w/o Lpatch	**0.3132**	4.4650	**0.5489**	**318.72**	0.2149
w/o Lrecon	0.4196	<u>4.7785</u>	0.5950	713.20	<u>0.2536</u>
Full model	<u>0.4075</u>	**4.9550**	<u>0.5846</u>	<u>485.00</u>	**0.2583**

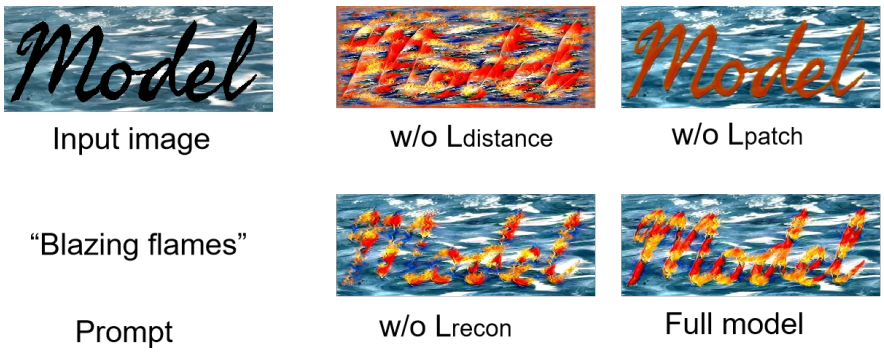

Fig. 8. Results of ablation studies.

Table 2. Quantitative evaluation between our method and previous methods.

	DISTS↓	NIMA↑	LPIPS↓	FID↓	CLIP SCORE↑
CLIPstyler	0.3901	4.6776	0.7171	**372.40**	0.1848
Sem-CS	<u>0.3838</u>	<u>4.7322</u>	<u>0.6984</u>	460.79	<u>0.2065</u>
Ours	**0.3324**	**4.8632**	**0.6667**	<u>445.55</u>	**0.2101**

4.3 Comparisons with Previous Methods

Existing methods of scene text editing address the conversion of text content and do not allow arbitrary style transformations of text. Therefore, in addition to the base method CLIPstyler, we compare our method with the style transfer methods that are closer to the purpose of our study. Sem-CS [17] and ControlNet [44] achieved image style transformation using prompts and more focus on the main part of the image rather than the whole image. The results of the comparison are shown in Fig.9. The results of CLIPstyler show that the style features were reflected throughout the entire image, resulting in the whole image change. Even though the results of Sem-CS reflect the style mainly on the text parts, style features tend to be around the text not inside the text, and make the style features not noticeable. The results of ControlNet reflected a noticeable style transformation, but the background changed significantly and the readability of the text declined. All these methods changed the background while achieving style transformation. Compared to previous methods, the proposed method

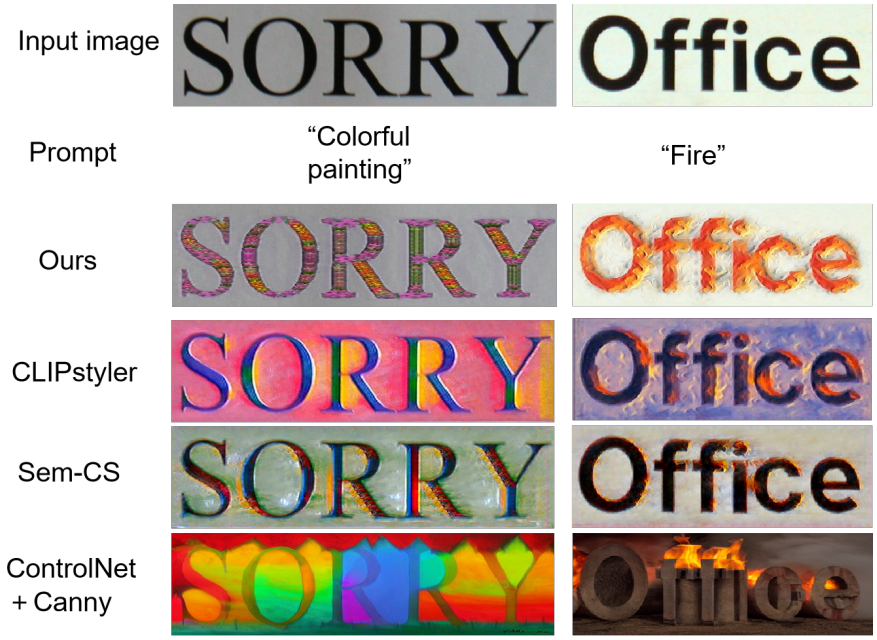

Fig. 9. The comparison results with other methods.

achieved the best results and is successful in transforming the style into the text region without changing the background and readability of the text.

Regarding qualitative evaluation, we used DISTS [6] and NIMA [30] scores to evaluate our method and compare it with previous methods. DISTS utilizes a texture resampling full-reference image quality model that matches human evaluations of image quality. NIMA proposes a deep CNN that can predict the distribution of image evaluation on human opinions from a direct view (technical perspective) and attractiveness (aesthetic perspective), thereby evaluating the images in terms of human perception. We evaluated 100 images of scene text style transformations obtained from our model and other methods. The results are shown in Tab 2. We achieved the best score in DISTS and demonstrated that our method realized style transformation in the text region while the generated image maintained the consistency of structure and texture information with the original image. We also obtained a better score in NIMA. Previous methods modified the entire image which could ensure the integrity of the image, under the condition of only changing a part of the image, our approach is still capable of generating naturally stylized scene text images with an aesthetic perspective. We also evaluate the LPIPS and FID [11] to value the similarity of the generated stylized image with the style image obtained by stable-diffusion-2-1-base with the same prompt. Moreover, We utilize the CLIP score to measure the similarity between generated images and prompts. Although our method only stylized the text region, while the other methods stylized the entire image, we achieved the best score in LPIPS, second only to CLIPstyler in FID. We obtained the best score in CLIP score, proving that our results were closest to the prompts.

5 Conclusion

In this paper, we proposed a new framework FontCLIPstyler to achieve the style transformation of scene text images. The proposed method does not require the style reference images and achieves arbitrary style transformation of text regions in the scene image using prompts. With the Text-aware Loss and MaskNet network, the proposed method solved the problem of CLIPstyler's inability to transform the style to a specific region in the image. The experimental results confirmed that our method could generate visually attractive stylized scene text while preserving the image background and text content. Our work could offer assistance in editing images like posters and designing more attractive artwork. Although this research realized the scene text style transformation, it is currently applicable only to the English alphabet, and can not transfer with more complicated text like Chinese characters or Japanese Kanji letters, since the MaskNet network only supports alphabet characters. In the future, we will continue to work on the realization of scene text style transformation in other languages.

References

1. Atarsaikhan, G., Iwana, B.K., Uchida, S.: Contained neural style transfer for decorated logo generation. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 317–322 (2018)
2. Azadi, S., Fisher, M., Kim, V.G., Wang, Z., Shechtman, E., Darrell, T.: Multi-content gan for few-shot font style transfer. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 7564–7573 (2018)
3. Chen, H., Xu, Z., Gu, Z., Li, Y., Meng, C., Zhu, H., Wang, W., et al.: Diffute: Universal text editing diffusion model. Advances in Neural Information Processing Systems **36** (2024)
4. Chen, J., Huang, Y., Lv, T., Cui, L., Chen, Q., Wei, F.: Textdiffuser: Diffusion models as text painters. Advances in Neural Information Processing Systems **36** (2024)
5. Deng, Y., Tang, F., Dong, W., Ma, C., Pan, X., Wang, L., Xu, C.: Stytr2: Image style transfer with transformers. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 11326–11336 (2022)
6. Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Image quality assessment: Unifying structure and texture similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(5), 2567–2581 (2020)
7. Gal, R., Patashnik, O., Maron, H., Bermano, A.H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators. ACM Transactions on Graphics (TOG) **41**(4), 1–13 (2022)
8. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 2414–2423 (2016)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in Neural Information Processing Systems **27** (2014)
10. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. arXiv preprint arXiv:2104.08718 (2021)

11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
12. Honghui, Y., Keiji, Y.: Multi-style shape matching gan for text images. IEICE TRANSACTIONS on Information and Systems **E107-D**, 505–514 (Apr 2024)
13. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 1501–1510 (2017)
14. Iluz, S., Vinker, Y., Hertz, A., Berio, D., Cohen-Or, D., Shamir, A.: Word-as-image for semantic typography. ACM Transactions on Graphics (TOG) **42**(4), 1–11 (2023)
15. Izumi, K., Yanai, K.: Zero-shot font style transfer with a differentiable renderer. In: Proceedings of the 4th ACM International Conference on Multimedia in Asia. pp. 1–5 (2022)
16. Ji, J., Zhang, G., Wang, Z., Hou, B., Zhang, Z., Price, B., Chang, S.: Improving diffusion models for scene text editing with dual encoders. arXiv preprint arXiv:2304.05568 (2023)
17. Kamra, C.G., Mastan, I.D., Gupta, D.: Sem-cs: Semantic clipstyler for text-based image style transfer. In: IEEE International Conference on Image Processing (ICIP). pp. 395–399 (2023)
18. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)
19. Krishnan, P., Kovvuri, R., Pang, G., Vassilev, B., Hassner, T.: Textstylebrush: Transfer of text aesthetics from a single example. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
20. Kwon, G., Ye, J.C.: Clipstyler: Image style transfer with a single text condition. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 18062–18071 (2022)
21. Li, W., He, Y., Qi, Y., Li, Z., Tang, Y.: Fet-gan: Font and effect transfer via k-shot adaptive instance normalization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 1717–1724 (2020)
22. Luo, C., Jin, L., Chen, J.: Siman: exploring self-supervised representation learning of scene text via similarity-aware normalization. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 1039–1048 (2022)
23. Ma, J., Zhao, M., Chen, C., Wang, R., Niu, D., Lu, H., Lin, X.: Glyphdraw: Learning to draw chinese characters in image synthesis models coherently. arXiv preprint arXiv:2303.17870 (2023)
24. Qu, Y., Tan, Q., Xie, H., Xu, J., Wang, Y., Zhang, Y.: Exploring stroke-level modifications for scene text editing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2119–2127 (2023)
25. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763 (2021)
26. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
27. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 234–241 (2015)

28. Roy, P., Bhattacharya, S., Ghosh, S., Pal, U.: Stefann: scene text editor using font adaptive neural network. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 13228–13237 (2020)
29. Song, Y., Zhang, Y.: Clipfont: Text guided vector wordart generation. In: British Machine Vision Conference. BMVA Press (2022), https://bmvc2022.mpi-inf.mpg.de/0543.pdf
30. Talebi, H., Milanfar, P.: Nima: Neural image assessment. IEEE Transactions on Image Processing **27**(8), 3998–4011 (2018)
31. Tanveer, M., Wang, Y., Mahdavi-Amiri, A., Zhang, H.: Ds-fusion: Artistic typography via discriminated and stylized diffusion. In: Proc. of IEEE International Conference on Computer Vision. pp. 374–384 (2023)
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)
33. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140 (2016)
34. Wang, C., Zhou, M., Ge, T., Jiang, Y., Bao, H., Xu, W.: Cf-font: Content fusion for few-shot font generation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 1858–1867 (2023)
35. Wang, W., Liu, J., Yang, S., Guo, Z.: Typography with decor: Intelligent text style transfer. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 5889–5897 (2019)
36. Wu, L., Zhang, C., Liu, J., Han, J., Liu, J., Ding, E., Bai, X.: Editing text in the wild. In: Proc. of ACM International Conference Multimedia. pp. 1500–1508 (2019)
37. Xie, Y., Chen, X., Sun, L., Lu, Y.: Dg-font: Deformable generative networks for unsupervised font generation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 5130–5140 (2021)
38. Xu, W., Long, C., Wang, R., Wang, G.: Drb-gan: A dynamic resblock generative adversarial network for artistic style transfer. In: Proc. of IEEE International Conference on Computer Vision. pp. 6383–6392 (2021)
39. Yang, Q., Huang, J., Lin, W.: Swaptext: Image based texts transfer in scenes. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 14700–14709 (2020)
40. Yang, S., Liu, J., Wang, W., Guo, Z.: Tet-gan: Text effects transfer via stylization and destylization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 1238–1245 (2019)
41. Yang, S., Wang, Z., Wang, Z., Xu, N., Liu, J., Guo, Z.: Controllable artistic text style transfer via shape-matching GAN. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 4442–4451 (2019)
42. Yang, Y., Gui, D., Yuan, Y., Liang, W., Ding, H., Hu, H., Chen, K.: Glyphcontrol: Glyph conditional control for visual text generation. Advances in Neural Information Processing Systems **36** (2024)
43. Yang, Z., Song, H., Wu, Q.: Generative artisan: A semantic-aware and controllable clipstyler. arXiv preprint arXiv:2207.11598 (2022)
44. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proc. of IEEE International Conference on Computer Vision. pp. 3836–3847 (2023)