

Towards Improvement to the CLIPstyler

Chuanruo Ning¹, Zekai Zhang¹, Rundong Luo¹

¹School of EECS, Peking University

{chuanruo, justinzzk, rundong_luo}@stu.pku.edu.cn

Abstract

This is the final report for the course Multi Modal Learning (22 Fall). Our team chose the paper “CLIPstyler: Image Style Transfer with a Single Text Condition” as our focus.

Neural Style Transfer (NST) is a popular machine learning technique that could apply a given style to the content of an image. The result is a new image that blends the two images in an artistic and visually appealing way. Neural style transfer has many applications, including generating artwork, enhancing photographs, and creating personalized and visually appealing images. Most previous NST approaches are based on image conditions, where the target style is given as an image. In this report, we explore the text-conditional NST following the CLIPstyler and propose several improvements. Extensive experiments demonstrate that our method can significantly enhance the visual quality of the generated images. Codes are available at <https://github.com/TritiumR/CLIPstyler>.

1. Introduction

Neural Style Transfer (NST) is a popular technique for combining one image’s content with another’s style using deep learning. The process involves training a convolutional neural network to generate an output image that preserves the content of the input image while adopting the style of a reference image. The result is a new image that combines the input image’s content with the reference image’s style. Neural style transfer has been used to create a wide range of creative and artistic effects, including the ability to transfer the style of famous paintings onto photographs and other images.

Conventionally, NST approaches are based on image conditions, i.e., the input style is given by an image (e.g., Great Wave off Kanagawa). However, reference images are usually not readily available to users, thus limiting NST’s broader applicability. To address such a problem, recent literature [6, 8] has extended image conditions to text conditions, where texts could determine the target style. In this

report, we explore the text-conditional NST following the CLIPstyler and propose several improvements. Extensive experiments demonstrate that our method can significantly enhance the visual quality of the generated images.

The remainder of this report is organized as follows. In Sec. 2, we review the existing works. Then, in Sec 3.1 and 3.2, we briefly introduce the method of CLIPstyler [6]. Next, we present our improvement to the CLIPstyler in Sec 3.3. In Sec 4., we demonstrate the effectiveness of our method through experiments. Finally, in Sec. 5, we summarize the paper and discuss future directions.

2. Related Works

Image-conditioned NST. Gatys et al. [3] proposed an iterative pixel-level optimization by jointly minimizing content and style losses. Following this seminal work, Johnson et al [5]. proposed a feed-forward network for real-time style transfer, where each network only supports one style. To resolve such a dilemma, Chen et al. [1] proposed the style bank module that allows for multiple styles, and Huang et al. [4] further devised a network that works for the arbitrary style. Recent works [2, 7, 10] generally focus on the network architecture.

Although these methods have shown successful results, the methods require style images in order to make the content image to follow the texture of the target style.

Text-conditioned NST. Text-conditioned NST is a relatively new field of research. Early works rely on jointly-trained text encoders and image encoders, which suffer from the drawback of distinct embedding spaces. Recent works [6, 8] take advantage of the introduction of CLIP [9]. Specifically, CLIPstyler [6] proposed the directional CLIP loss and multiple auxiliary loss functions for stable image synthesis.

3. Method

In this section, we first briefly introduce the method proposed in CLIPstyler [6]. Then, we introduce our modification to their framework.

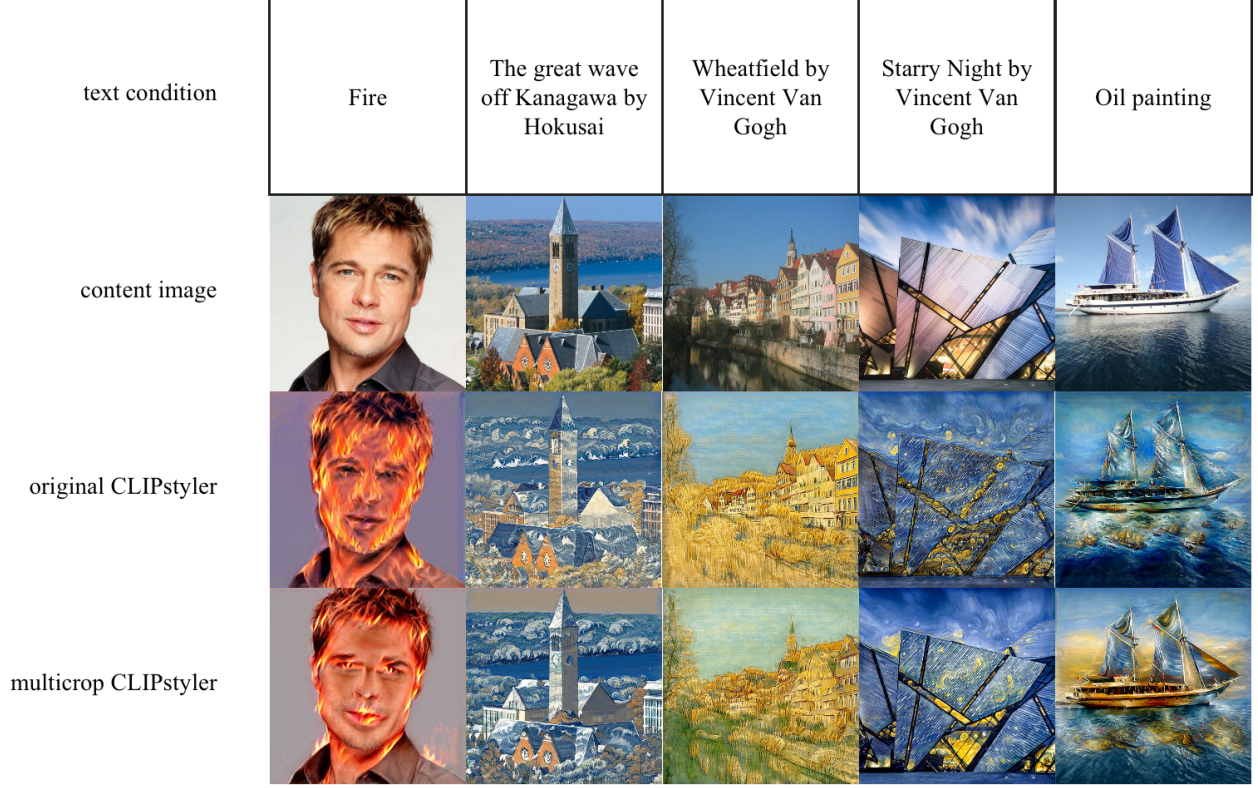


Figure 1. Comparison results between the original CLIPstyler and multicrop CLIPstyler.

3.1. Notations and Preliminaries

The purpose of text-conditioned style transfer is to apply the semantic style of target text t_{sty} to the content image I_c , thus obtaining the stylized image I_{cs} . The transformation is achieved by an encoder-decoder network, denoted by f . Different from image-conditioned style transfer, there is no style image to use as a reference. We denote the text encoder and image encoder of CLIP [9] by E_T and E_I , respectively. Our ultimate goal is to optimize the parameter of f given E_T and E_I so that $f(I_c)$ can preserve the content information of I_c while obtaining the texture information given in t_{sty} .

3.2. CLIPstyler

This section introduces the loss functions of CLIPstyler, which is the key to its empirical success.

CLIP Loss. The CLIP loss serves as the core of style fusion. It is formulated as:

$$\begin{aligned}
 I_{cs} &= f(I_c) \\
 \Delta T &= E_T(t_{sty}) - E_T(t_{src}), \\
 \Delta I &= E_I(I_{cs}) - E_I(I_c), \\
 \ell_{\text{CLIP}}(I_{cs}, I_c) &= 1 - \frac{\Delta I \cdot \Delta T}{|\Delta I| |\Delta T|}, \quad (1)
 \end{aligned}$$

where t_{src} is the prompt for input images. The authors set it as “Photo” for simplicity.

Patchwise CLIP Loss with Threshold Rejection To improve local stylization quality, the authors proposed the Patch-wise CLIP loss. Specifically, they randomly crop N patches $\mathcal{P} = \{I_1, \dots, I_n\}$ from I_{cs} , and then apply projective transformations before calculating the CLIP directional loss. They claim that using augmentations on each patch assists the network in representing more vivid and diverse textures. Moreover, to avoid over-stylization, they add a threshold to mute the loss of over-stylized patches. Finally, their patchwise CLIP loss is formulated as:

$$\mathcal{L}_{\text{patch}} = \frac{1}{N} \sum_{I_p \in \mathcal{P}} R(\ell_{\text{CLIP}}(I_p, I_c)), \quad (2)$$

where

$$R(x, \tau) = \mathbf{1}(x > \tau) \cdot x \quad (3)$$

is the threshold rejection function with threshold τ . Similarly, the global CLIP loss is defined as:

$$\mathcal{L}_{\text{global}} = \ell_{\text{CLIP}}(I_{cs}, I_c). \quad (4)$$

Total Loss. The overall loss function is a weighted combination of the global CLIP loss, the patch-wise CLIP loss,

the content loss [5] \mathcal{L}_c , and the TV loss \mathcal{L}_{tv} :

$$\mathcal{L}_{\text{total}} = \lambda_g \mathcal{L}_{\text{global}} + \lambda_p \mathcal{L}_{\text{patch}} + \lambda_c \mathcal{L}_c + \lambda_{tv} \mathcal{L}_{tv}. \quad (5)$$

3.3. Improved CLIPstyler

Our common problem of existing NST approaches is that the style is applied globally, i.e., the whole content image is affected. As shown in Figure xxx, the CLIPstyler is not an exception. In our work, we aim to control the image regions affected by the style.

Multi-crop Patch CLIP Loss. In the original form of $\mathcal{L}_{\text{CLIP}}$, ΔI is calculated between patches of I_{cs} and the original image I , which may induce a semantic discrepancy. Therefore, we propose to calculate $\mathcal{L}_{\text{patch}}$ between patches of both I and I_{cs} , i.e.,

$$\mathcal{L}_{\text{patch}} = \frac{1}{N} \sum_{I_p \in \mathcal{P}} R(\ell_{\text{CLIP}}(I_p, I_{cp})), \quad (6)$$

where I_{cp} is the corresponding patch of I_p in content image I_c . We call this improved version ‘‘multicrop CLIPstyler’’.

Localized Style Transfer. The original CLIPstyler applies target style to the whole image, which may induce undesirable artifacts, as shown in Figure 2. To address such a problem, we propose localized style transfer by exploiting the semantics similarity given by the CLIP model. By giving different source descriptions of images, we could control the part of image we want to stylize.

Specifically, similarity weight is added to specific parts of the image, while the weight of other parts are kept unchanged. We formulate the similarity weight w_i as:

$$w_i = \frac{S(I_{cp}^i, t_{src}) \mathbb{I}(S(I_{cp}^i, t_{src}) > \tau)}{\sum_{I_{cp} \in \mathcal{P}} S(I_{cp}, t_{src})} \quad (7)$$

and the $\mathcal{L}_{\text{patch}}$ is modified to

$$\mathcal{L}_{\text{patch}} = \sum_{I_p \in \mathcal{P}} w_i R(\ell_{\text{CLIP}}(I_p, I_{cp})), \quad (8)$$

where I_{cp}^i denotes the corresponding patch of I_p in content image I_c , S is the similarity metric given by CLIP model and \mathbb{I} is an indicator function (1 if the expression is true and 0 otherwise). We set τ as 0.25 in our implementation.

This modification intuitively means putting different weights to the patchwise loss by measuring the similarity between the cropped content images and the source description.

We also develop a training method for localized style transfer. We first train the model with only content loss \mathcal{L}_c and TV loss \mathcal{L}_{TV} to force the output to be close to the original image. Then we apply weighted patchwise loss $\mathcal{L}_{\text{patch}}$ and global CLIP loss $\mathcal{L}_{\text{global}}$ to stylize the part assigned by

source text. This pipeline ensures the stylization of specific part, while leaving other parts unchanged.

Style-aware Selective Sampling. Authors of CLIPstyler [6] claims that CLIPstyler suffers from over-stylization problems. It is caused by the indiscriminate updating of all sampled patches, including those well-stylized ones that have been updated in previous processes. To address this issue, we introduce Style-aware selective sampling, a method that considers individual patches’ style intensity. By setting a style intensity threshold α beforehand and only updating patches with style intensity below the threshold, we can control the overall stylization level and prevent over-stylization.

Given a patch I_p and a style text t_{sty} , we compute the dot product between the patch and the style text as a measure of style intensity, i.e.,

$$SI(I_p) = E_I(I_p) \cdot E_T(t_{sty}) \quad (9)$$

where SI refers to the style intensity of a given patch.

To effectively control the stylization of the whole image, we only update a subset of the sampled patches, namely $\mathcal{P}_{US} \subseteq \mathcal{P}$, that have style intensities below a threshold value α and are therefore under stylized. Thus, the $\mathcal{L}_{\text{patch}}$ is modified to

$$\mathcal{L}_{\text{patch}} = \sum_{I_p \in \mathcal{P}_{US}} R(\ell_{\text{CLIP}}(I_p, I_{cp})), \quad (10)$$

$$\mathcal{P}_{US} = \{I_p | I_p \in \mathcal{P}, SI(I_p) < \alpha\} \quad (11)$$

By adjusting the value of α , we can control the maximum style intensity of individual patches. This enables us to exert control over the whole image and achieve the desired level of artistic expression.

4. Experiments

We reproduce all experiments in the paper, including single-image style transfer, fast style transfer and video style transfer. Results could be find in the slide we submit.

4.1. Multicrop CLIPstyler

Figure 1 demonstrates the stylization result of our multicrop CLIPstyler. As shown in the figure, our improved version can significantly reduce the artifact and thus make the generated image more realistic.

4.2. Localized Style Transfer.

We conduct experiments on diverse content images and source prompts to test model’s ability to stylize specific part of the image.

Figure 2 demonstrates the localized stylization result of our localized style transfer. Compared with original styler, our method could distinguish the part we want to stylize and leaving other parts unmodified, removing unwanted effect.







content image	source prompt	target prompt	styler	localize styler
	boat	fire		
	water	stars		
	sky	Starry Night by Vincent van gogh		
	forest	The great wave off Kanagawa by Hokusai		

Figure 2. Comparison results between the original CLIPstyler and the localized CLIPstyler.

4.3. Style-aware Selective Sampling

Figure 3 demonstrates the style controllability of our style-aware selective sampling method. By increasing α , the generated images exhibit a smooth transition in style attributes such as colors, textures, and stylized areas, as well as a steady increase in style intensity.

Our method also addresses the problem of over-stylization. As shown in Figure 4, while CLIPstyler ($\alpha=1.0$) produces an incorrect application of “scream” textures to Lena’s hat, our selective sampling method can avoid this issue by setting a relatively low style intensity threshold (e.g., $\alpha=0.1$) while maintaining a high level of stylization in the overall image.

5. Conclusions

In this report, we explored the text-conditioned neural style transfer and proposed multiple practical approaches to minimize the existence of artifacts. Going beyond the simple patch-wise global stylization scheme, our proposed multicrop loss and region-aware loss can exert regional control to the stylization process. Extensive experiments have demonstrated the superiority of our method.

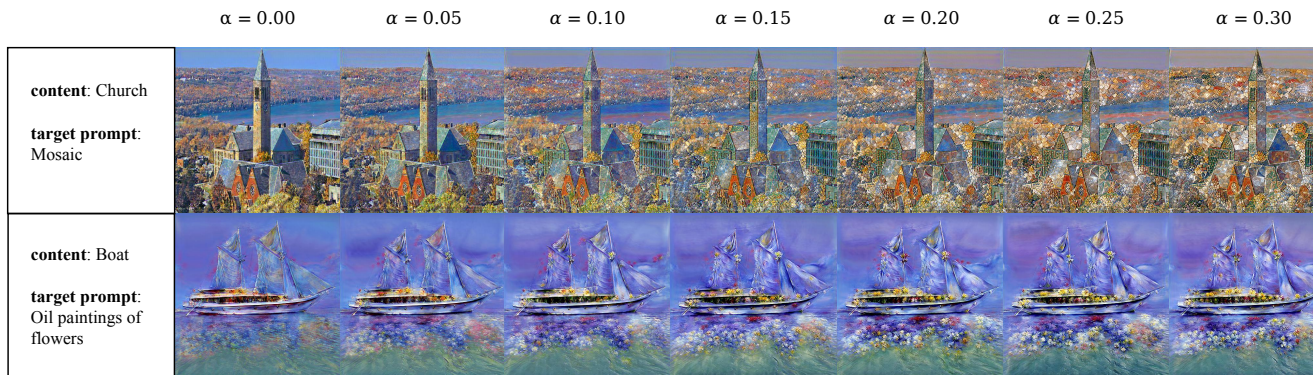


Figure 3. Experimental results of Style-aware Selective Sampling on style controllability.

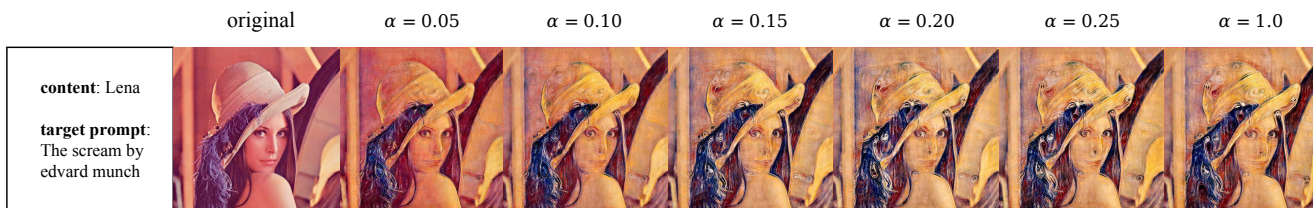


Figure 4. Experimental results of Style-aware Selective Sampling on over-stylization.

References

- [1] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *CVPR*, 2017. 1
- [2] Yingying Deng, Fan Tang, Weiming Dong, Wen Sun, Feiyue Huang, and Changsheng Xu. Arbitrary style transfer via multi-adaptation network. In *ACM MM*, 2020. 1
- [3] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 1
- [4] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 1
- [5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 1, 3
- [6] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *CVPR*, 2022. 1, 3
- [7] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *CVPR*, 2019. 1
- [8] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 1
- [9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2
- [10] Wenju Xu, Chengjiang Long, Ruisheng Wang, and Guanghui Wang. Drb-gan: A dynamic resblock generative adversarial network for artistic style transfer. In *ICCV*, 2021. 1