Mixed Neural Style Transfer

Jianfeng Guo
New York University
jg6483@nyu.edu

Tianyi Li
New York University
tl3285@nyu.edu

Cheng Qian
New York University
cq2045@nyu.edu

Abstract

This project explores methods for artistic style transfer based on convolutional neural networks. Applying a different style to the semantic content of an image is difficult and a lot of researchers have published work related to it. But most works are about one content image and one style image. Here we render the semantic content of an image in multiple styles. We explore different ways to transfer multiple styles to the content image, including mixing multiple styles on the content at once and applying styles on the object and the background respectively.

1. Introduction

1.1. Neural style transfer

Neural style transfer is an optimization technique that takes two images including a content image and a style reference image (such as an artwork image) and blends them together so that the resulting image preserves the semantic content of the content image with style in the style reference image [2, 5]. It's one of the heated applications of convolution neural networks (CNN). Neural style transfer is fascinating because it creates a new image using existing content and a different artistic style, which can be Picasso's style or Vincent van Gogh's style. Theoretically, the problem is difficult but with CNN, we can deal with the problem and can even control how the final image is produced. Our project is to explore the CNN model of style transfer proposed by Gatys' paper [2] and improve the performance of the model.

1.2. Mixed Neural Style Transfer

Most of the existing work of style transfer including [2] is about one content image and one style image. But there are situations where multiple styles are applied to the content image. We decided to add additional feature layers so that our model can handle multiple styles. In the experiment, we will explore how to blend multiple styles and apply the mixed style transfer to the content image.

2. Related works

Gatys *et al.* [2] proposed a new style transfer algorithm based on CNN which extracts the semantic representations from the content image and then informs a texture transfer process to render the semantic content of the content images in the style of the style image because previous works only use low-level images features of the content image to inform the texture transfer.

Though [2] showed that the style and the content of an image can be disentangled and applied independently, their method was based on a slow optimization process and computationally expensive. Johnson *et al.* [4] talked about this issue and improved the performance by introducing perceptual losses and training a feed-forward network to replace the optimization-based method of Gatys. Furthermore, Huang *et al.* [3] proposed the network can learn a set of adaptive instance normalization (AdaIN) parameters representing a style. Chandran *et al.* [1] proposed Adaptive Convolutions (AdaConv), a generic extension of AdaIN, to allow for the simultaneous transfer of both statistical and structural styles in real-time. Besides, [1] experimented with style interpolation by generating the image with multiple styles.

3. Methods

3.1. Deep image representations

Gatys et al. [1] proposed deep image representations that independently model variations in the semantic image content and the style in which it is presented. Their approach was based on the VGG network [6]. The content representation of an image is encoded in each layer of CNN by the filter response to that image. Suppose a layer with N_l filters has N_l feature maps each of size M_l. So, the response in layer l are a matrix:

$$F_l \epsilon R^{N_l \times M_l} \quad (1)$$

Let p and x be the content image and the image that is generated, and P^l and F^l their respective feature representations in layer l. The content loss between the two feature representations at layer l is:

$$L_{content}(p, x, l) = \frac{1}{2}\sum_{i,j}\left(F_{ij}^l - P_{ij}^l\right)^2 \quad (2)$$

The style representation is obtained by using a feature space built on top of the filter responses, which consists of the correlations between the different filter responses. These feature correlations are given by the Gram matrix $G^l \epsilon R^{N_l \times N_l}$, where G_{ij}^l is the inner product between the vectorized feature maps i and j in layer l:

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \quad (3)$$

Let a and x be the style image and the image that is generated, and A^l and G^l their respective style representation in layer l. The contribution of layer l to the total loss is then:

$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}\left(G_{ij}^l - A_{ij}^l\right)^2 \quad (4)$$

and the total loss is:

$$L_{style}(a, x) = \sum_{l=0}^{L} w_l E_l \quad (5)$$

where w_l are weighting factors of the contributions of each layer to the total loss.

3.2. Neural Style Transfer

To transfer the style of an artwork a onto a photograph p, we generate a new image x that matches the content representation of p and the style representation of a. Thus, we minimize the distance of the feature representations of a white noise image from the content representation of the photograph in one layer and the style representation of painting defined on a number of layers of the CNN. The loss function we minimize is (6) where α and β are the weighting factors for content and style construction respectively.

3.3. Mixed Neural Style Transfer (MNST)

The previous approach is restricted to only one style. We want to expand the situations in that multiple styles are applied to the content images. In the previous function, α and β are the weights for the content loss and the style loss, respectively. Therefore, we introduce γ as a new style weight. So, the loss function is (7) where γ is weighting style loss 1 and style loss 2 individually, which influences the overall style loss. By shifting the value of γ in the range between 0 and 1, the result for two style images would change the dominator for the styles.

$$L_{total}\ (p,a,x) = \ \alpha L_{content}(p,x) + \ \beta L_{style}\ (a,x) \tag{6}$$

$$L_{MNST}\ (p,a_1,a_2,x) = \alpha L_{content}(p,x) + \ \beta\left(\gamma L_{style}\ (a_1,x) + \ (1-\gamma)L_{style}\ (a_2,x)\right) \tag{7}$$

3.4. Localized style transfer

During the experiments, we found that style transfer was applied to the entire image which cannot distinguish the background and people. So, we performed object detection on approaches for achieving partial-image style transfer. This would keep the people in the original image, while the mixed style would be applied to the background image.

To perform localized style transfer, we use Deeplab [7] to identify the objects which are classified as human. Deeplab is a deep learning model for semantic image segmentation, whose mission is to assign semantic labels like people in our task. The goal of semantic segmentation is to label each pixel with a given class [8]. After we finish the semantic segmentation processing, each pixel of the content image is labeled by a model classifier.

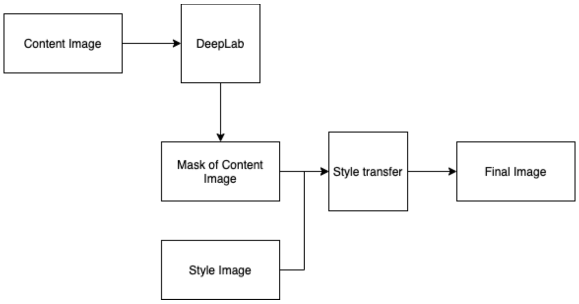

Figure 1. The architecture of localized style transfer work

4. Experiments

In this section, we explore how the parameters in the loss function affect the results of style transfer. First, we evaluate the effectiveness of ratio α/β in single-style transfer. Then, we change the style weight γ to obtain the different changes. At the last, we apply Deeplab model to do semantic segmentation work by keeping the portion of people from the content image and transforming the background.

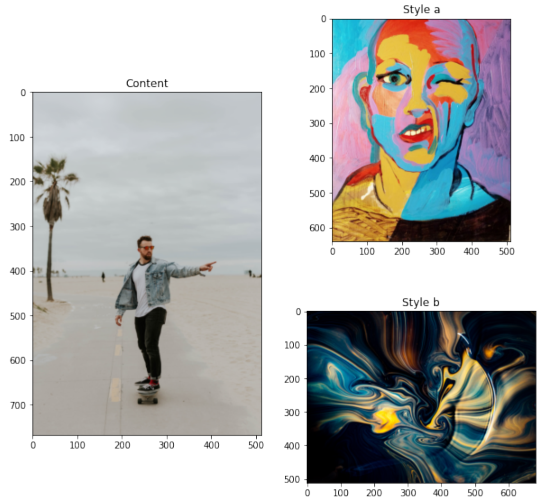

Figure 2. Content image and two styles images

4.1. Experiment with different content weight and style weight ratio

When blending the content image and the styles image, the generated image usually can’t perfectly match every constraint at one time. But since the loss function is a linear combination between the content loss and the style loss, we can alternate the emphasis on the content or the style. A strong emphasis on the style will produce images that match the appearance of the style image but shows hardly the content image. If the emphasis is placed on the content, the content is well-preserved but the style is not as well-matched. Below are the results with the different relative weighting of matching content and style.

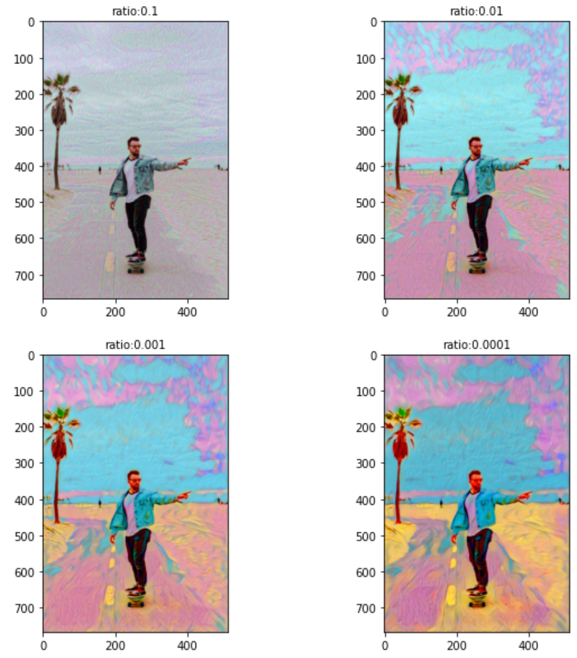

Figure 3. The effect of different ratio α/β

From our observations, we found that when the ratio α/β is 1×10^{-2}, the generated image has a well-balanced effect of content and style. Therefore, we choose the ratio of content weight and style weight equals to 1×10^{-2} as the style transfer parameters for all other experiments.

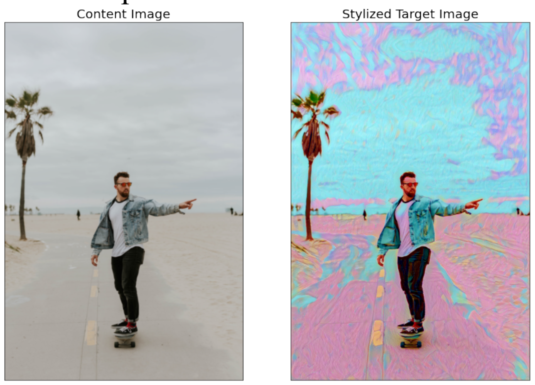

Figure 4. Content image and Stylized image

4.2. Experiment with weights of different styles

Since we want to apply multiple styles to the content, we want to explore how different style weights γ affect the results. So, we trained with different values of γ from 0 to 1. And below are the results.

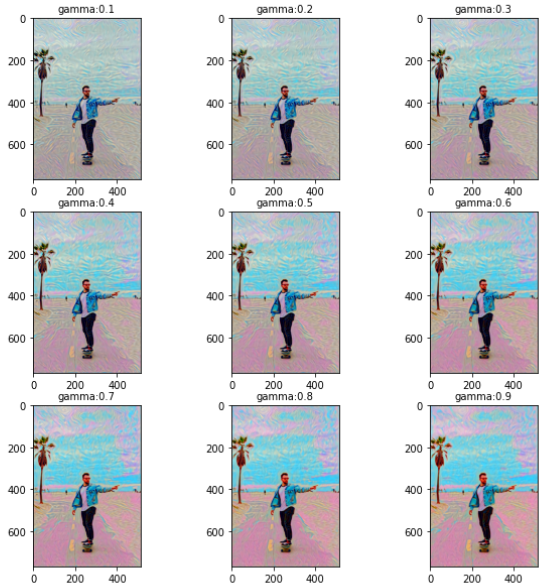

Figure 5. The result of the values of γ, based on the α and β ratio is 1×10^{-2}

When comparing the figures with different values of γ, it can clearly see how the weight of the first style image increasingly impacts the whole-content image.

We use RI to denote the relative improvement for style loss 1 and 2 during the optimization process. It compares the initial loss of the first iteration against the loss after the last iteration. While c stands for the content image, x is the generated image after the last iteration.

$$RI_1 = L_{style}(s_1, x)/L_{style}(s_1, x) \quad (8)$$

$$RI_2 = L_{style}(s_2, x)/L_{style}(s_2, x) \quad (9)$$

And we explore how the RI1 and RI2 develop for different values of γ by calculating the relative improvements for values from γ=0.1 to γ=0.9 in steps of 0.1. The ratio between RI_1 and RI_2 was used to find out how much more style loss 1 was improved compared to style loss 2.

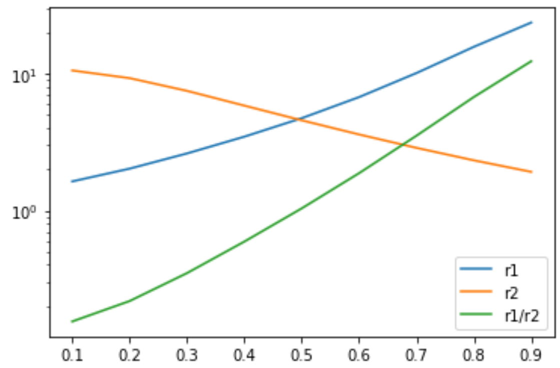

Figure 6. Observation of r1 and r2 for different values of γ

4.3. Experiment with Localized Style Transfer

The ratio α/β in this experiment is 1×10^{-2} and γ *is* 0.2. First, we perform object detection to detect people on the content image. Then, a mask of the content image is generated using semantic segmentation. The pixels of the mask correspond to the people of the content image.

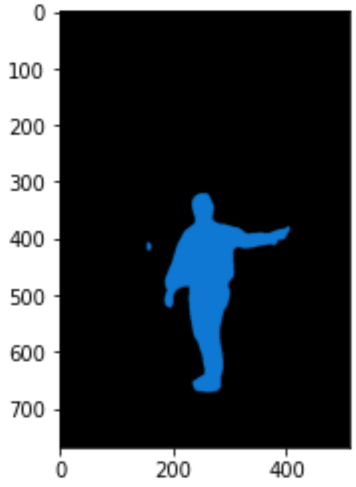

Figure 7. The Deeplab-generated mask

After we get the mask, we can recognize the pixels of the content image which is labeled by different classifications. The model does an excellent job of correctly classifying the pixels of people and also includes the passerby in the background, which is not the main object in our content image.

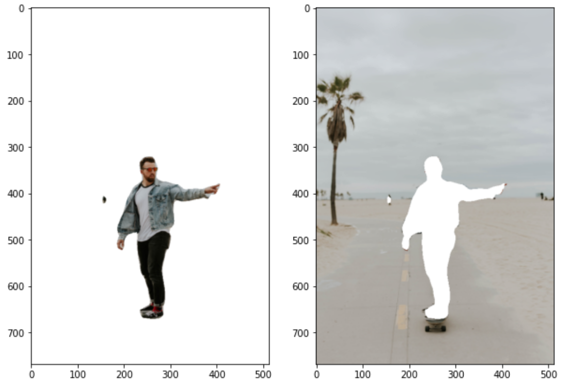

Figure 8. The selected portion of content image in Figure 2

The result of the Localized Style Transfer is shown in Figure 9. To get this result, we performed the mixing style transfer over the content image (Figure 2) with Nicola Powys' painting (Figure 2)

and Dan-Cristian's work (Figure 2). Using a mask generated by Deeplab, we then replaced pixels in the style-transferred image with pixels from the content image. This enabled us to recover the people from the content image while performing the mixing style transfer on the background. We can see that the person in the resulting image () has not performed the style transfer, but the background is transferred.

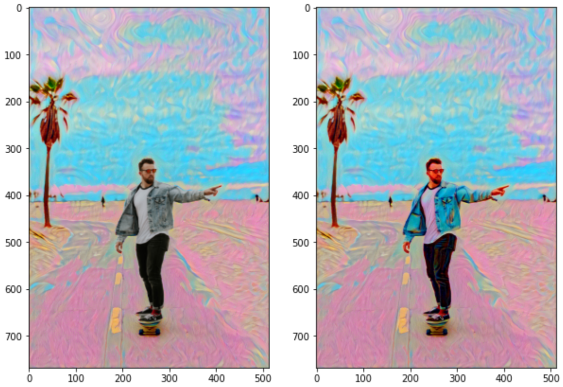

Figure 9. The final result image of using both mixing style transfer and localized style transfer

5. Conclusion

Although it is difficult to evaluate the quality of mixing style transfer objectively because the value of artistic work is a subjective opinion, we think style transfer could be applied in portrait effect - Localized style transfer. We use semantic segmentation in the content image so that the person is well into the style-transferred background.

6. References

[1] P. Chandran, G. Zoss, P. Gotardo, M. Gross and D. Bradley, "Adaptive Convolutions for Structure-Aware Style Transfer," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 1,2

[2] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer(CVPR), June 2016. 1

[3] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE International Conference on Computer Vision(ICCV), Oct 2017. 1

[4] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision, 2016. 1

[5] "Neural style transfer, Tensorflow Core" TensorFlow. [Online]. Available: https://www.tensorflow.org/tutorials/generative/style_transfer. 1

[6] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for r using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Large-Scale Image Recognition. arXiv:1409.1556 [cs], Sept. 2014. arXiv: 1409.1556. 2

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2017. 3

[8] "Stanford University CS231N: Deep Learning for Computer Vision." [Online]. Available: http://cs231n.stanford.edu/reports/2017/pdfs/416.pdf. 8