Neural Style Transfer on Pixel Level and Brushstroke Level

Sixing Zhou Xiaoyun Zhi Han Wang
Department of Computer Science
New York University
{sz3704, xz3256, hw2725}@nye.edu

Abstract

Neural style transfer takes a content image and a style image as input and it generates a new image that applies the style to the first content image. Applying semantics of an image in different artistic styles is a difficult task. The common approach to this problem is to use a conventional neural network to find the content and style features from each image and output an image with mixed features. We will be exploring different pre-trained models, namely VGG-16 and VGG-19, each with different hyperparameters and optimizers in this task. We found that VGG-19 with Adam optimizer generates the best results among all experiments. Additionally, training models on parameterized brushstrokes yields better results than pixels. It takes 1.5 minutes inference time on average to generate an output image on a NVIDIA V100 GPU. Code for this project is available at https://github.com/HanNight/NYU-CV-Fall2022-Final-Project.

1. Introduction

The use of CNN technique in style transfer has been prevalent following the pioneer paper by Gatys et al. [4]. The advances in this field have been applied in real world projects, such as TikTok and Instagram, both of which allow users to make more creative contents. It is both an interesting and a challenging topic to explore because of the creativity of neural networks and its constraints on representing styles.

As of the approach proposed by Gatys et al. [4], we conducted experiments using both VGG-16 and VGG-19 backbones. On top of that, we tested different optimizers, including Adadelta, Adagrag, Adam, AdamW, RMSprop, SGD.

The above-mentioned experiments are conducted on pixel level input, which is unnatural in realistic works as paintings are drawn with brushes by artists. Following the guide by Kotovenko et al. [9], we experimented on optimizing parameterized brushstrokes instead of pixels.

2. Related Work

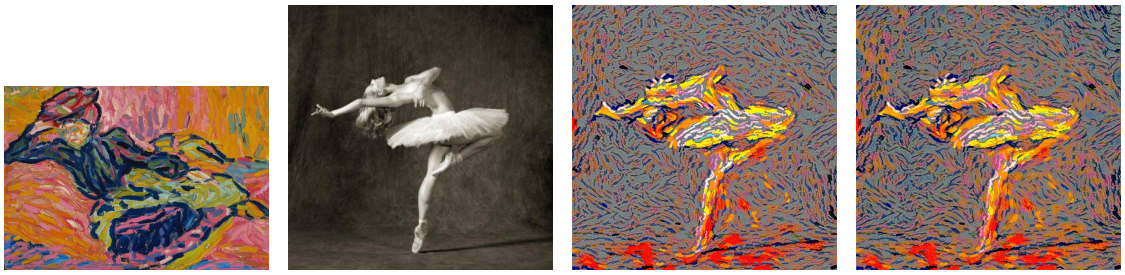
To generate a new content image with the style from another image while persisting the structure of the original image, this problem falls into the category of texture transfer. Previous research tackling this problem utilized non-parameterized approaches to perform texture synthesis. For instance, Efros and Freeman [3] designed a correspondence map that contains features of the target image. Wang et al. [16] introduced a method for synthesizing 2d directional texture.

After the advance of deep neural networks, Gatys et al. [4] showed that generic feature representations of content and style images can be learned by these CNNs. Their method jointly minimizes the content loss from pretrained models and style loss from the Gram metrics. Dumoulin et al. [2] designed a conditional instance normalization algorithm, which allows a single network to capture 32 different styles at the same time. Li et al. [11] proposed to treat neural style transfer as a domain adaptation problem, and they showed that Maximum Mean Discrepancy with the second order polynomial kernel is equivalent to the Gram metrics. Johnson et al. [7] proposed a feed-forward network which yields similar results but three orders of magnitude faster compared to the optimization approach by Gatys et al.

There have been a number of researches on stroke based rendering and brushstroke extraction. Stroke based rendering aims to synthesize an artwork by compositing marks such as lines and brushstrokes, which can be defined as a set of parameters. Hertzmann [6] proposed a method to generate a painting with a series of layers, each with progressively smaller brushstrokes. On the other hand, brushstroke extraction identifies brushstrokes in an image. Li et al. [10] utilized edge detection and clustering based segmentation to extract brushstrokes in an image.

3. Method

In the original style transfer formulation, Gatys et al. [4] propose an iterative method for combining the content of one image with the style of another by jointly minimizing



(a) Style (b) Content (c) VGG-16 (d) VGG-19

Figure 1. Girl on the Divan (style) + Dancing Girl (content) for Neural style transfer under VGG-16 and VGG-19.

content and style losses. The pixels are adjusted to match the brushstroke pattern and adjust each pixel to minimize the content and style losses. In contrast, Kotovenko et al. [9] choose to optimize directly on parameterized brushstrokes, using the same content and style losses.

In this paper, we will first illustrate Gatys' method to separate and recombine the image content and style of natural images. What's more, take an insight into synthesized brushstroke patterns parameterized by Dmytro.

3.1. Pixel-level Neural Style Transfer

3.1.1 Deep image representations

Generally each layer in the network defines a non-linear filter bank whose complexity increases with the position of the layer in the network. Gatys assumes that perform gradient descent on a white noise image makes it possible to find another image that matches the feature responses of the original image, which is defined as content reconstructions [13]

Let $\overrightarrow{p}$ and $\overrightarrow{x}$ be the original image and the image that is generated, and P^l and F^l their respective feature representation in layer l. Gatys define the squared-error loss between the two feature representations as follows.

$$\mathcal{L}_{content}(\overrightarrow{p}, \overrightarrow{x}, l) = \frac{1}{2}\sum_{i,j}(F_{ij}^l - P_{ij}^l)^2 \tag{1}$$

The derivative of this loss with respect to the activations in layer l equal from which the gradient with respect to the image $\overrightarrow{x}$ can be computed using standard error back-propagation.

$$\frac{\partial \mathcal{L}_{content}}{\partial F_{ij}^l} = \begin{cases} (F_l - P_l)_{ij} & if\ F_{ij}^l > 0 \\ 0 & if\ F_{ij}^l < 0 \end{cases} \tag{2}$$

Style representation, which built on different layers of the network by constructing an image that matches the style representation of a given input image, shows good performance with stationary and multi-scale texture information.

Let $\overrightarrow{a}$ and $\overrightarrow{x}$ be the original 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

$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}(G_{ij}^l - A_{ij}^l)^2 \tag{3}$$

Let w^l the weighting factors of the contribution of each layer to the total loss, the total style loss can be calculated as:

$$L_{style}(\overrightarrow{a}, \overrightarrow{x}) = \sum_{l=0}^{L} w_l E_l \tag{4}$$

3.1.2 Style Transfer

The process of style transfer can be viewed as synthesising a new image that simultaneously matches the content representation of $\overrightarrow{p}$ and the style representation of $\overrightarrow{a}$. The loss function is defined as (5) where α and β are the weighting factors for content and style reconstruction, respectively.

$$L_{total}(\overrightarrow{p}, \overrightarrow{a}, \overrightarrow{x}) = \alpha L_{conetent}(\overrightarrow{p}, \overrightarrow{x}) + \beta L_{style}(\overrightarrow{a}, \overrightarrow{x}) \tag{5}$$

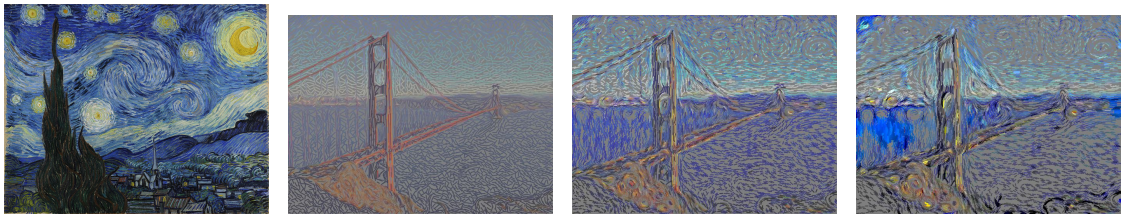
Gatys' method shows that the representations of content and style in the Convolutional Neural Network are well separable. Manipulate both representations independently to produce new, perceptually meaningful images seems to be possible.

3.2. Brushstroke-level Neural Style Transfer

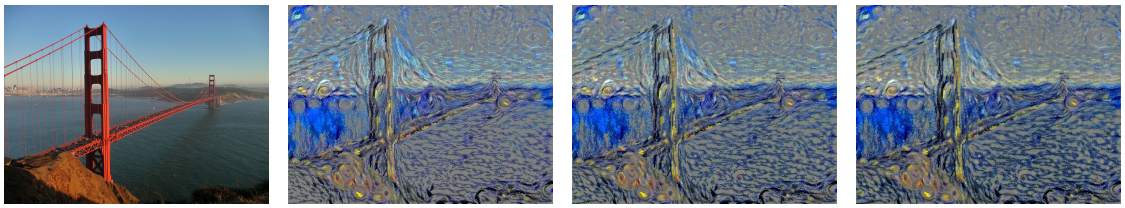
The separation of image content from style is not necessarily a well defined problem. Kotovenko assumes that paintings basically consist of brushstroke instead of color map. This may due to the vague definition of evaluation criterion if the style transfer is successful or not.

3.2.1 Evaluation Criterion for style transfer

Deception rate, which proposed by Sanakoyeu et al[18], is used to evaluate the quality of the stylization. It is defined



(a) Style (b) 10 steps (brushstroke) (c) 50 steps (brushstroke) (d) 100 steps (brushstroke)



(e) Content (f) 100 steps (pixel) (g) 500 steps (pixel) (h) 1000 steps (pixel)

Figure 2. Starry Night (style) + Golden Gate Bridge (content): Intermediate images during brushstroke and pixel optimization. Zoom in on details.

as the fraction of stylized images that the network has assigned to the artist, whose artwork has been used for stylization. Higher deception score a stylized image reach, the more plausible it is.

The method of style image grouping gives a quantized thoughts to find a set Y of related style images where a given single style image y_0 be classified $y_0 \in Y$.

Let $\Phi(y)$ be the activations of the fc6 layer of the VGG16 network C for input image y. Retrieve all nearest neighbors of y_0 based on the cosine distance δ of the activations $\Phi(.)$, i.e.

$$Y = \{y \mid y \in \wp, \delta(\Phi(y), \Phi(y_0)) < t\} \tag{6}$$

3.2.2 Presentation of parameterized brushstrokes

The definition of content and style loss function in Gatys et al's method is still in use in Kotovenko's experiment. However, Kotovenko optimize directly on parameterized brushstrokes instead of pixels. The brushstrokes are parameterized by location, color, width and shape. The shape of a brushstroke is modelled as a quadratic Bezier curve, which can be viewed in equation 7.

$$B(t) = (1-t)^2 * P_0 + 2(1-t)t * P_1 + t^2 P_2, 0 \leq t \leq 1 \tag{7}$$

To find an efficient and differentiable mapping from the brushstroke parameter space into the pixel domain. Kotovenko put forward a brand new rendering mechanism. The process can be decomposed into three parts.

First, compute a matrix of distances to the curve DB and mask points that are closer than the brushstroke width. It is worth noting that in specific experiment, Kotovenko samples equidistant points along the curve and computing the minimum pairwise distance between the point and curve. It is claimed that there exists an analytical solution of this distance for a quadratic Bezier curve.

Next, compute the individual renderings of brushstrokes and the assignment matrix, which defined as equation 8. The equation indicates which object is the nearest to the coordinate (i, j).

$$A_{i,j,n} = \begin{cases} 1 & if\ D_n(i,j) < D_k(i,j)\ \ \forall k \neq n \\ 0 & otherwise \end{cases} \tag{8}$$

The final image can be computed by the weighted sum of renderings weighted according to the assignment matrix A. It corresponds to equation 9.

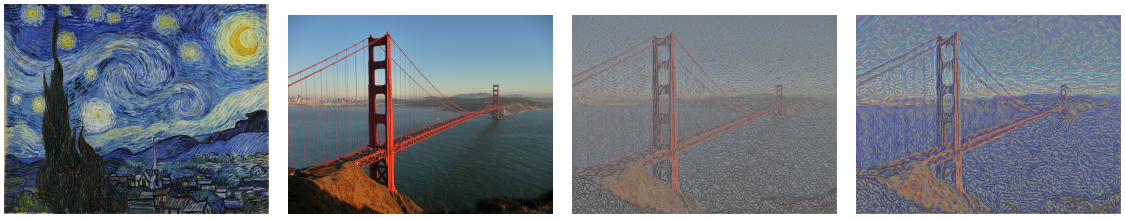
$$I(i,j) = \sum_{n=1}^{N} I_n(i,j) * A(i,j,n) \tag{9}$$

Kotovenko's method shows good performance where brushstrokes are clearly visible and the representation is more natural for artistic style transfer. This can be viewed in the following experiment.

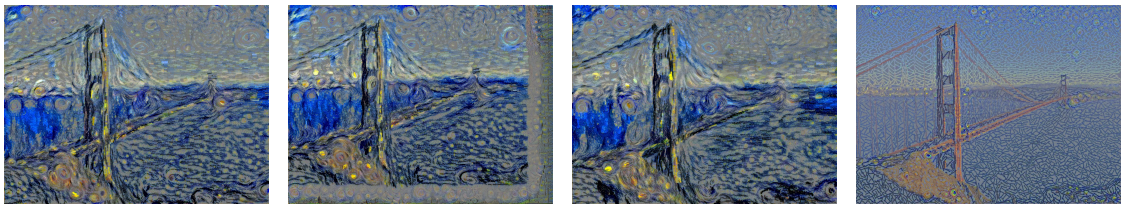
4. Experiment

4.1. Experimental Setting

We closely follow the setup in Kotovenko et al. [9]. We adopt VGG-16 [14] as our backbone network to extract the features of the input image. Our stylization process consists of two stages. At the first stage we optimize brushstroke parameters, at the second stage we optimize individual pixels.



(a) Style (b) Content (c) Adadelta (d) Adagrad



(e) Adam (f) AdamW (g) RMSprop (h) SGD

Figure 3. Starry Night (style) + Golden Gate Bridge (content) for neural style transfer with the six optimizers under VGG-16. Zoom in on details.

Brushstroke parameters optimization. We use layers "conv4_2" and "conv5_2" for the content loss and layers "conv1_1", "conv2_1", "conv3_1", "conv4_1", and "conv5_1" for the style loss. We optimize our render for 100 steps using Adam Optimizer with a learning rate of 0.1.

Pixel optimization. The layers "conv1_2_pool", "conv2_2_pool", "conv3_3_pool", "conv4_3_pool", and "conv5_3_pool" are used for the content loss and the layers "conv1_1", "conv2_1", "conv3_1", "conv4_1", and "conv5_1" are used for the style loss. We upsample the canvas with fitted strokes to have the smallest image side of 1024px and keep the input content image aspect ratio. This image with fitted brushstrokes is used as both content image and initialization for the standard Gatys et al. [4] stylization routine. We optimize for 1000 steps using the Adam optimizer. All the experiments are conducted on one NVIDIA Tesla V100 with 16GB.

4.2. Intermediate Images During Optimization

Figure 2 shows the effect of the brushstroke optimization and the pixel optimization. The brushstroke optimization is gradually adding stroke details to the image, from simple lines to oil brush lines. The pixel optimization makes brushstrokes blend together and adds texture.

4.3. VGG-16 vs VGG-19

When given the same content and style images, the VGG-19 using Adam optimizer generates visually better output, as shown in Figure 1.

4.4. Optimizer Experiments

We attempted six commonly used optimizers in our VGG-16 model:

- Adadelta [17]
- Adagrad [1]
- Adam [8]
- AdamW [12]
- RMSprop [5]
- SGD [15]

Figure 3 shows the final output images using different optimizers. In terms of visual effect, Adam and RMSprop obviously outperform other optimizers. The three optimizers Adadelta, Adagrad, and SGD only make the brushstrokes appear on the image, but they do not perform well in style learning at the pixel level. AdamW optimizer makes the gray areas appear on the left and bottom of the image, which damages the overall effect.

4.5. Brushstroke vs Pixel Optimization

We conduct experiments with different update steps for brushstroke optimization and pixel optimization to explore the relationship between the output image and the number of update steps for brushstroke and pixel optimization. We set the number of brushstroke and pixel optimization steps to $0, 250, 500, 750, 1000$ respectively, so that we can get $5 \times 5 = 25$ combinations and output images. As shown in Figure 4, we find that as the number of brushstroke optimization steps increases, the content of the picture will become more and more blurred. For example, the ivory

Figure 4. Picasso's Painting (style) + Elephant (content) for neural style transfer with different brushstroke optimization steps and pixel optimization steps. Zoom in on details.

in the figure becomes more and more unclear from left to right. In addition, as the number of pixel optimization steps increases, the style of the style image becomes more prominent on the output image. Therefore, we suggest to choose less brushstroke optimization steps and more pixel optimization steps to get the better style transfer image.

5. Discussion

In this report, we explore the influence of a number of factors on the representation of style transfer at the pixel level and brushstroke level, such as the type of optimizer, and the optimization steps. We find a limitation of this approach is that it performs best for artistic styles where brushstrokes are clearly visible. As shown in Figure 5, this approach performs well for Van Gogh's oil painting style while performing terribly for the Chinese ink painting style. This may potentially be alleviated with more sophisticated brushstroke blending procedures and should be investigated in future endeavors.

References

[1] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.

[2] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

[3] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001.

[4] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[6] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 453–460, 1998.

(a) Content (b) Style 1 (c) Output 1 (d) Style 2 (e) Output 2

Figure 5. Road (content) + Van Gogh's oil painting (style) or Chinese ink painting (style) for neural style transfer.

[7] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[9] D. Kotovenko, M. Wright, A. Heimbrecht, and B. Ommer. Rethinking style transfer: From pixels to parameterized brushstrokes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[10] J. Li, L. Yao, E. Hendriks, and J. Z. Wang. Rhythmic brushstrokes distinguish van gogh from his contemporaries: findings via automated brushstroke extraction. *IEEE transactions on pattern analysis and machine intelligence*, 34(6):1159–1176, 2011.

[11] Y. Li, N. Wang, J. Liu, and X. Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017.

[12] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[13] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[15] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2013.

[16] B. Wang, W. Wang, H. Yang, and J. Sun. Efficient example-based painting and synthesis of 2d directional texture. *IEEE Transactions on Visualization and Computer Graphics*, 10(3):266–277, 2004.

[17] M. D. Zeiler. Adadelta: An adaptive learning rate method, 2012.