Neural Style Transfer with Self-Attention Module
UCSD COGS185 Final Project

Qiyu Chen

q5chen@ucsd.edu

Abstract

While transferring the style from one image to another is a hot topic as well as a challenging problem, numerous researchers try to utilize new architectures to improve upon the earliest neural style transfer method. Some of them try to combine another popular technique, the self-attention module, into the design. In this paper, we apply the self-attention module directly on the original style transfer algorithm in order to examine its efficacy in this problem. Our results demonstrate its ability to add stylized details in the generated image compared to the vanilla algorithm. However, further analysis indicates some existing limitation of this approach and conclude about its greater effectiveness in a model-optimization-based setting.

1 Introduction

Since the groundbreaking work of Gatys et al. [1] which proposed that the feature maps from a pre-trained deep neural network can capture the style pattern of an image and be utilized to transfer its artistic style onto another images, countless approaches has been come up to improve upon this method [2].

Among these new methods, many researchers have employed the model-optimization-based neural approaches, which trains on a feedforward network to speed up the slowness of the image-optimization-based methods like the vanilla neural style transfer algorithm [2]. By putting the waiting time to the training phase, these works largely reduce the amount of time required to process new incoming images, thereby achieving real-time video style transfer.

Besides speed, many networks try to incorporate new architectures between the usual encoders and decoders to ameliorate the quality of the synthesized stylized image, [3, 4]. The original loss function in the original Gatys' algorithm, which of the content loss and style loss, has a well known trade-off between the content image's local spacial arrangement and the style image's global style pattern [2]. To solve this limitation, researchers design better architectures to realize new loss functions which are capable of producing more balanced and detailed result. For example, AdaIN [3] modifies the content loss and changes the style representation using mean and variance statistics and SANet [4] adds a new identity loss into the loss function.

While developing new network architectures, some of them including the SANet bring the conception of self-attention into this field of style transfer. The self-attention module can refine the features at each position by putting higher attention to relevant and necessary parts over all locations. It is a popular technique in natural language processing like the Transformer [7] and nowadays also in computer vision such as the Non-local Neural Networks [5].

In this work, we examine the effectiveness of the self-attention in the problem of neural style transfer by apply the Gaussian-based self-attention module back to the original algorithm by Gatys [6]. In additional to the original loss function, we add a third component, an attention-based style loss. This new loss component is similar to the normal style loss, except that the target style feature map is a modification of the old feature map based on the spatial correspondence between the content image and the style image defined using the self-attention module. In our experiment, this new component can to some extent enable the generated image to retain more local appearance of the content image without breaking the global style consistency.

During the further analysis, this work compares the effect different hyper-parameters for the new loss component. In the end, we discuss the limitation of the self-attention module inside an image-optimization-based setting and conclude about its greater power in a model-optimization-based neural methods.

2 Method

The neural style transfer algorithm implemented by this paper is a modification of the original algorithm by Gatys et al., which contains a content loss and a style [6], along with another attention-based style loss. The goal of this new loss component is to preserve the local details from the content image with their corresponding style from similar patterns in the style image.

All input for these loss terms are the feature map output of a pre-trained deep convolutional neural network VGG19, which is suggested by Gatys and for better comparison. VGG19's small kernel size and deep structure render it capable of capturing more spatial features at multiple scales, which are important for neural style transfer problem.

The algorithm implemented contains large flexibility as it can realize the original loss function as well as the new loss function just by changing one hyper-parameter. To best preserve the result achieved by the vanilla algorithm, the implementation includes many common techniques in neural style transfers including filter output mean normalization and re-

placement of maximum pooling layers by average pooling layers as mentioned in [6], and some other ones like total variation denoising as described in [2],

2.1 Content Loss

The content loss is the weighted sum of mean squared distance between the feature maps of the content image C^l and the generated image F^l at each layer l. The feature map at each output layer is then transformed into a matrix with height N_l (the number of distinct filters) times H_l and width W_l (size). Accordingly the content loss at each interested layer l is as follows:

$$L_{content}(C, F, l) = \frac{1}{2N_l H_l W_l} \sum_{i,j} (F^l_{i,j} - C^l_{i,j})^2 \quad (1)$$

Based on [6], feature maps from lower layers in the VGG19 network try to capture the exact pixel values of the input content image, whereas those from higher layers contains high-level content information in terms of structure and arrangement. Since our goal is to transfer the style from one picture to another picture's structure instead of its pixel values, it is reasonable to also use a higher layer. Specifically, we stay with using the 'conv4_2' layer for content representation for better comparison.

2.2 Style Loss

Similarly, this work sticks with Gram matrix for style representation. A gram matrix is the result of multiplying a given matrix by its transposed matrix. It can capture the correlations between different filter responses at each position over the compared images.

$$G^l_{i,j} =< F^l_i, F^l_j > \quad (2)$$

The style loss will be a weighed sum of the mean square error between the Gram matrix $G^l \in \mathbb{R}^{N_l \times N_l}$ of the generated image and the Gram matrix S^l of the style image using feature maps at layer l. Each of the feature map will have height N_l (number of distinct filters) and width $H_l \times W_l$ (size) and the gram matrix will have both height and width of $H_l \times W_l$. So the style loss is then:

$$L_{style}(S, G, l) = \frac{1}{4N_l^2 (H_l W_l)^2} \sum_{i,j} (G^l_{i,j} - S^l_{i,j})^2 \quad (3)$$

Again based on Gatys' experiments [6], the correlations of feature maps from multiple layers provide a "stationary, multi-scale representation" of the image artistic style. Using feature maps up to a higher layer discards more global arrangement of the scene and captures the style at larger scale. Therefore, we likewise decide to include up to layer 5 ('conv1_1', 'conv2_1', 'conv3_1', 'conv4_1' and 'conv5_1') to obtain a comprehensive representation of the style.

On the other hand, the way that Gram matrices are calculated determines this style loss will compress local details for global arrangement. This defect is why in output images by vanilla neural style transfer algorithm, some local areas possess the right style but not the same appearance. The problems inspires us to find a way to better retain the spatial details from the input content image.

2.3 Attention-based Style Loss

We start out by envisioning that each small area on the content image can know what the style should be on the most similar areas on the style image. The idea of this 'knowing' corresponds with the essence of the self-attention module, and can use this technique to accomplish the desired results. This is also the idea behind the design of the SANet [4].

However, we are focusing on image-optimization-based approaches, which differs from model-optimization-based methods like SANet in that it cannot train parameters for a fixed model or, more importantly, an embedded space. Thus, we decide to choose a simpler design for the self-attention module which uses only the Gaussian function without any embedding space as described in [5].

$$f(x_i, x_j) = e^{x_i^T x_j} \quad (4)$$

Fortunately, [5] also discovers that the choice of the function does not have significant result on its output. However, different from its non-local model, our structure is closer to the design of the SANet model [4] which also computes the similarity between two inputs instead of one. We create a new style target feature maps FA where each position becomes a linear combination of the style image responses A^l with weights based on the dot-product similarity to the content image responses C^l at this location through the softmax function for each interested layer l:

$$FA^l_i = \frac{1}{\mathbf{C}(FA)} \sum_{\forall j} f(\overline{C^l_i}, \overline{A^l_j}) A^l_j \quad (5)$$

$$\mathbf{C}(FA) = \sum_{\forall j} f(\overline{C^l_i}, \overline{A^l_j}) \quad (6)$$

where i denotes a pair of coordinates on the output matrix, and j enumerates all positions on the style image feature map in matrix form, $\mathbf{C}$ represents the softmax normalization factor, and the bar indicates the matrix is mean-variance channel-wise normalized, which is the same for inputs in SANet. Comparing the Gram matrix of the new target style feature map FA and the generated image feature map F, our attention-based style loss is defined as:

$$L_{attention\_style}(FA, F, l) = L_{style}(SA, G, l) \quad (7)$$

where SA is the Gram matrix style representation from the mean normalized version of FA, and G is that using F. Here we do not choose to normalize variance for FA because our experiments show that the synthesized image owns more detail and texture in this way.

During the experiments, we also compare the effect of selecting feature maps from the third, the fourth, and the fifth

layers. In addition to these layers, we try to generate the weight matrix from the output of softmax operation for the second layer. Nevertheless, since the size of the weight matrix is $(N_l W_l)^2$, which means for an image of 512 $\times$ 512 resolution, it contains over 4 billion entries, such matrix creates an overwhelming burden on the memory and consequently we disregard the option below the third layer. Furthermore, we downsample the weight matrix to mimic a style representation from the sixth layer. However, since for an image of 512 $\times$ 512 pixels, its width and height is only 16 and the oversized receiptive field renders it incapable of capturing detailed information, the result of using it is almost neglegible.

2.4 Style transfer: total Loss

Combining the three aforementioned components together, we have our total loss function designed to not only jointly minimize the content feature difference and the Gram matrix style representation difference between an white noise and the content image and the style image, respectively, but also preserve details in their corresponding style. So the total loss is as follows:

$$L_{total}(C, S, FA, F) = \alpha \sum_l w^{cl} L_{content}(C, F, l) + \beta \sum_l w^{sl} L_{style}(S, F, l) + \gamma \sum_l w^{al} L_{attention\_style}(FA, F, l) \tag{8}$$

where w is the weight for each loss term at each layer, and α, β, γ are the weighting factor for the content loss, style loss, and attention-based style loss, respectively. Here if γ is set to 0, then the loss function is exactly the one used by Gatys in [6]. Even though we eventually decide to only include one content layer, here we leave the full expression of our loss function and implement it in similar way to preserve the maximum flexibility in for our experiments. We also notice that the effect of the tuning of w is trivial compared to the choice of layers. Therefore, we determine to apply average layer weight for each loss, which is likewise adopted by Gatys.

Besides, like Gatys, we will L-BFGS algorithm for optimization since it proves to produce the most sophisticated result, and resize the input content and style image to the same size for adoption of style at similar scales.

3 Experiment

3.1 Comparison with vanilla neural style transfer

The main goal of this paper is to examine the effectiveness of self-attention module on neural style transfer problem. To do this, we disregard many new techniques and models appearing in recent years and directly add an attention-based style loss using self-attention module to the original algorithm. To make the comparison more reasonable, we first tune the weighting factor α and β with γ being zero to accomplish attractive output images with the vanilla algorithm, respectively. Subsequently, we fix α and β and then change γ to the empirically best value.

All of the images are generated under 500 iterations, and both methods take around 21 seconds on Google Colab Pro with GPU environment.

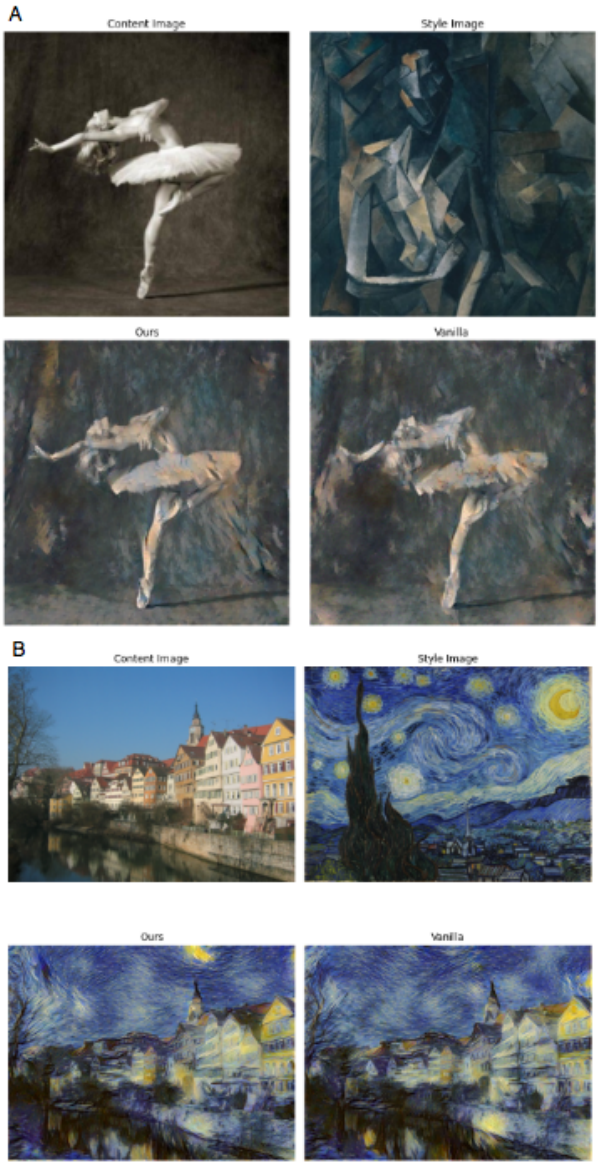

Figure 1. Comparison of our method and the vanilla method. Both **A** and **B** use α, β, γ of 1, 1e7, and 6e6 respectively

Our experiments demonstrate that with some fine-tuning on the hyper parameters, the new attention-based style loss term can contributes to better preservation of the detail with the expected style. For example, in Fig. 1A, the ankle of the dancer contains more abundant shading, and her fingers are more visible; similarly in Fig. 1B, lots of details of the buildings are preserved compared to the vanilla methods. One thing worth noting is that in Fig. 1B, the reflection of the yellow house in the river obtains more styles from the yellow moon.

3.2 Effect of different layers

As mentioned previously, the choice of which layer as the source of responses is critical to effect of the attention-based style loss, because their different receiptive ranges decides the size and type of the local spatial patterns the new loss enforces on the white noise image. Throughout the same depth, the response from the first convolutional layer seems to capture the most details and is the stable one, we compare the synthesized images using feature maps from layers at different depth to demonstrate their difference.

Figure 2. Synthesized images using different layers as the input for the attention-based style loss.

The result shown in Fig. 2 is produced with $\gamma = 5e7$ which is slightly larger than it should be to underpin their difference. Using lower layers adds many small patches onto the original images due to the smaller receptive field in the convolutional layer, and the resultant effect is conspicuous. But the problem that on such small scale, the feature map may not contain enough information to find places in the style image with similar semantic meaning. On the other hand, feature maps from higher layers bring effects in larger local area, but the effect is less visible under the same γ and sometimes such large area is not helpful in improving small details like the fingers in this case. Therefore, to align with our purpose of enriching the detail of the synthesized image, we eventually select only 'Conv4_1'as the source of feature maps. The comparison here also shows the inevitable need for hyper-tuning γ for each individual choice of the layer to use.

3.3 Comparison of optimizers

We also compare the effect of different optimizers on the resultant images. As shown in Fig. 3, LBFGS algorithm is more suitable for our method in contrast with Adam, since it provides faster result and more interesting details

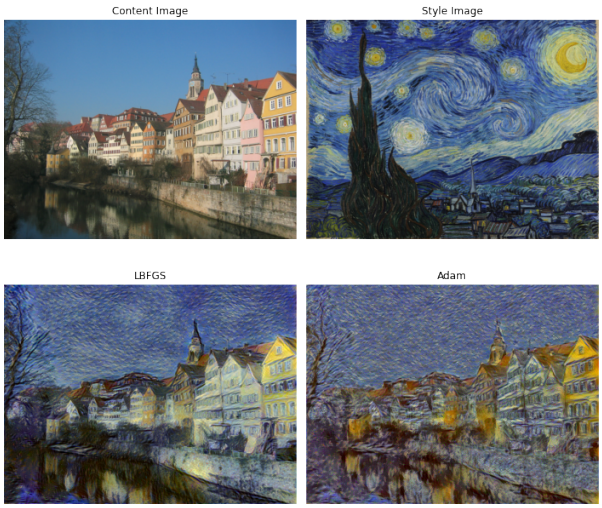

Figure 3. Comparison of LBFGS (left) and Adam optimizers (right, with learning rate of 0.01). Both use α, β, γ of 1, 1e6, and 5e6 respectively

3.4 Explore the weight matrix under softmax

During our results, sometimes the generated image does not learn the expected styles from corresponding places with similar semantic meaning. This phenomenon spurs us to visualize the weight matrix calculated while producing FS in order to check if the self-attention module maps the position between the content image and the style image correctly.

To get some insight into the accuracy of the self-attention module, we examine some positions in the content images and plot the weights for correspondence between this position and the whole the style feature using feature maps from different layers, with the position of the highest weight marked as a red dot in the middle and the right images.

The varying sizes of the right images are expected as we discussed above that they correspond to the size of the receiptive field of the convolutional layer. However, one unexpected discovery is that although on all A, B, and C in Fig. 4, we select a point in the sky, which is visually similar, the result of the highest matching positions in the drawing are completely different using distinct layer responses. Furthermore, even for the largest receptive field in Fig. 4B and C, the positions of top weights does not largely resemble the semantics of the red dot in Fig. 4A. Therefore, it causes us to doubt about the accuracy and consequently the effectiveness of self-attention module.

4 Discuss

Not only does the previous section implies some limitation of our method, previous sections also point out the need for more hyper-tuning in our approach, which is inconvenient and time-consuming to use. On top of these, another significant disadvantage can be seen from Fig. 2A that though the new style

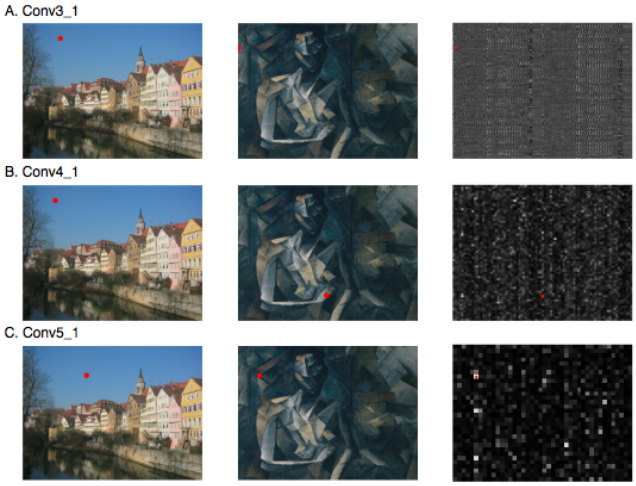

Figure 4. Visualization of the weight for one point. The red marker inside left images are roughly the point we are comparing. The red dot inside middle images are the location of the style image with highest correspondence by our self-attention module, and the right images provide visualization of weights over all indices, with white being high value and the red marker being the highest

feature map contains the anticipated style based on correspondences, it may not produce constructive influence when directly combining features together linearly and can even cause huge noise when γ is set to be too large. This is also the reason that we often set γ to be at least smaller than β.

Therefore, it can be concluded that without an embedding space from training, the spatial self-attention module is not so effective in our image-optimization-based setting for neural style transfer compared to the efficacy in a model-optimization-based method [4]. Nonetheless, it still proves to be able to improve the synthesized image quality if used with care.

5 Conclusion

In this paper, we examine the effectiveness of the self-attention module in the field of neural style transfer by adding another attention-based loss term to the original loss function. Our experiment indicates its ability to enrich some stylized details in the generated image with some fine-tuning compared to the vanilla neural style transfer algorithm. Nevertheless, with further analysis, we demonstrate and discuss some limitations of directly applying self-attention technique in an image-optimization-based setting, and conclude its greater efficacy when we can make use of it with an embedded space through training.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in Advances in Neural Information Processing Systems, 2015, pp. 262–270.

[2] Jing, Y., Yang, Y., Feng, Z., Ye, J., Song, M., "Neural style transfer: A review," 2017, arXiv preprint arXiv:1705.04058.

[3] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1501–1510.

[4] Park, D. Y., and Lee, K. H, "Arbitrary style transfer with style-attentional networks," 2019, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5880–5888.

[5] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. "Non-local neural networks," In CVPR, 2018.

[6] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2414–2423.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, In Advances in Neural Information Processing Systems, pages 5998–6008.