A new method of style transfer based on AdaIN and Laplacian

王梓桐
北京大学数学科学学院
2300010750@stu.pku.edu.cn

黄天域
北京大学数学科学学院
2300010720@stu.pku.edu.cn

Abstract

Style transfer is a technique that combines the content of one image with the style of another, holding significant value in computer vision and artistic creation. This study proposes an improved method based on AdaIN and LapStyle algorithms, effectively leveraging the strengths of both to achieve significant enhancements in style transfer performance. Experimental results demonstrate that, compared to classic methods, our approach exhibits superior performance in terms of style-content fusion, detail processing, and overall stylization quality. This research not only accelerates convergence but also improves the quality of generated images, offering an effective solution for style transfer tasks.

1 Introduction

Style Transfer has always been a popoular and active field of computer vision, the core task of which is to generate a stylized image that resembles the content of one image and the style of another image. Conventional approaches mainly treat the style transfer problem as a texture synthesis problem. Since Gatys et al. (1) proposed a image-iteration method based on convolutional neural networks (CNN's), the research of neural style transfer (NST) has achieved great improvements and notable prospect. Improvements on this method has also been further researched, leading to the proposal of new tricks and insights such as LapStyle NST (2).

Despite the succesful of style transferring based on image iteration, the high computational cost and low genarating efficiency limits further application. To resolve these limitations, model-iteration based method, especially light-weight models such as AdaIN have been proposed(3). AdaIN aimes to realize real-time and high-efficiency style transfer through adaptive normalization of features. Nevertheless, singular normalization might not be expressive enough to capture complicated styles, and the model-based iteration proves to be difficult for users to fine-tune the generated images.

Preprint. Under review.

Therefore we propose a new method chaining the models together, combining the advantages of the two models, to generate high-quality stylized images with high efficiency.

2 Related works

Many has achieved great progress in the field of neural style transfer. Classical works of Gatys' et al (1) introduces CNNs pretrained on large image datasets to extract features. Further improvements on this image-iteration based method include deploying different priors, such as Markov Random Field (5) and Laplacian loss(8).

Since one can always treat the image optimization process as reconstructing images from its feature representation, this leads to GAN-based(11) and transpose convolution based structures(7). These approches can be classified generally as model-iteration based. Combined with the usage of masks, this enables finer control over the generation of stylized images. Further improvements on image quality are propelled by the trending of attention mechanisms and transformers, as shown in (10),(9).

3 Data usage

In our work, we used COCO2017 dataset(4) from Microsoft, which is composed of 80,000 images of various contents, as the training content images for the AdaIN part of our method. As for style images, we used the images from (6), the 20 images of which proved to be enough for our limitated computational resources. We resized the images to 512x512 and then randomly cropped the images to 256x256 size. We also deployed the classical normalization to suit VGG(citaion needed), which we used as feature extractor.

4 Methods

The methods we deploy are mainly Gatys' and Laplacian (1)(2), along with AdaIN (3) as an improvement to the classical approaches.

4.1 Gatys' style transfer

Gatys' et al. proposed to use VGG as a high-level feature extractor to capture the sematic informations of the images. Specifically, by passing the input image through multiple convolutional and pooling layers of the CNN, different levels of feature representations are obtained. In these layers, lower-level features retain more detailed information of the image, while higher-level features capture the high-level content, such as objects and their layout in the image. By reconstructing the image from a specific higher layer (e.g., "conv4_2"), an image that preserves the original content without being constrained by exact pixel values can be obtained. Thus we can

define the content loss $\mathcal{L}_{content}$ as the euclidean distance between high level features:

$$\mathcal{L}_{content} = \|\mathbf{x}_c - \mathbf{x}_{cs}\|^2 \tag{1}$$

To capture the style of an image, we compute the correlations between different feature maps in various convolutional layers, i.e. calculating Gram matrices, resulting in a multi-scale style representation. If we denote the feature as $\mathbf{x} \in \mathbb{R}^{C\times W\times H}$, then the Gram matrix$G \in \mathbb{R}^{C\times C}$is defined as:

$$G_{u,v} = \frac{1}{HW}\sum_{i,j} \mathbf{x}_{u,i,j} * \mathbf{x}_{v,i,j} \tag{2}$$

A total loss function is then defined, which is a linear combination of content loss and style loss:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{content} + \beta\mathcal{L}_{style} \tag{3}$$

where $\mathcal{L}_{style}$ is defined as:

$$\mathcal{L}_{style} = \sum \|G_s^k - G_{cs}^k\|^2 \tag{4}$$

By minimizing the mean squared error between the Gram matrices of the original and generated images, an image that matches the style of a given input image can be generated. Adjusting the weights (α and β) of content and style losses yields a method to control the balance between content and style in the generated image at will. Specifically, the minimization process is achieved by Limited-Memory BFGS methods, which is able to converge with acceptable amount of time.

The algorithm performs well on most cases, but the synthesized images are known to suffer from low-frequency noise. This is primarily shown as displaced structures from style images, affecting the realism of the synthesized image. Images generated by Gatys' method also suffer from distortion and artifacts, inspiring the following improvements.

4.2 Laplacian style transfer

Aimed at solving these problems, Shaohua Li et al. developed a new loss factor $\mathcal{L}_{lap}$ based on the Laplacian operator $\triangle$ To be more specific, we first perform a $p \times p$ average pooling first denoise, then use a convolutional Laplacian filter to calculate the laplacian, and compare it to the laplacian of the content image. We denote the mean squared error as $\mathcal{L}_{lap}$. After adding this loss term, we have:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{conent} + \beta\mathcal{L}_{style} + \gamma\mathcal{L}_{lap} \tag{5}$$

Through limiting the high-frequency feature differences, the introduction of Laplacian loss can ease the problems stated above.

4.3 AdaIN

Despite the image-iteration based methods mentioned above, we also take great interest in more recent model-based methods, namely AdaIN(3). In this method, we treat the features extracted

by VGG as latent representation of images, and mix them using Adaptive Instance Normalization defined as:

$$\text{AdaIN}(x, y) = \sigma(y) * \frac{x - \mu(x)}{\sigma(x)} + \mu(y) \tag{6}$$

where x denotes the feature of the content image and y denotes the feature of the style image, and $\sigma(\cdot)$ denotes the standard variance and $\mu(\cdot)$ denotes the mean.

Then we feed this mixed latent representation to a transpose convolution neural network, structured as the exact inverse of VGG.

Similar to the previous methods, we optimize the following loss function:

$$\mathcal{L} = \|t - t_{mixed}\|_2 + \sum_i \left(\left\| \mu\left(F_{\mathbf{s}}^i\right) - \mu\left(F_{mixed}^i\right) \right\|^2 + \left\| \sigma\left(F_s^i\right) - \sigma\left(F_{mixed}^i\right) \right\|^2 \right) \tag{7}$$

where t_{mixed} is the latent representation of the generated image extracted by VGG, and F_*^i denotes the output of "ReLU_i_1". We optimize this loss over 10 epochs, each epoch running over the COCO dataset and a random image from the style dataset, and obtain a AdaIN model.

4.4 Our improvements

In our work, we chained the AdaIN module we trained and the Laplacian style model together, using the AdaIN model's output as the initial image for optimization in Gatys' approach, as an attempt to enable better control and easier fine-tuning for model-based approches. This method also aim to achieve better and faster stylization results compared to simply deploying Gatys' or Laplacian by utlizing large amount of prior knowledge acquired from AdaIN traing process. The results can be seen in 5.2.

5 Experiments

5.1 Comparision between Gatys style and Lapstyle

We successfully implemented Gatys-style and LapStyle-based style transfer. Below, we will present our results and compare the effectiveness of the two algorithms. The second photo was taken at NingLang by a team member during a trip to Yunnan.

图 1: content(style) image

图 2: Gatys-style

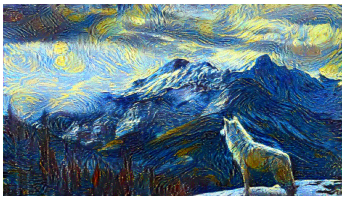

图 3: Lapstyle

Based on the comparison of the two sets of images above, it is very clear that the stylized images generated using LapStyle are noticeably smoother compared to those produced by Gatys-style.

图 4: content(style) image

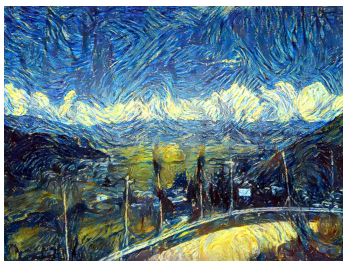

图 5: Gatys-style

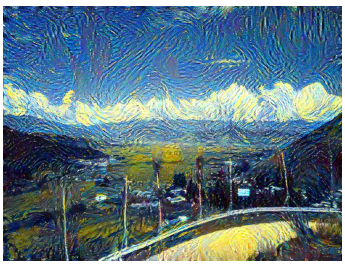

图 6: Lapstyle

For example, in the upper-left corner of the first set of images, the storm clouds in the Gatys-style result are visibly more chaotic, less smooth, and exhibit more abrupt color transitions influenced heavily by the style. This results in a less harmonious overall appearance. Beyond details like the storm clouds, in the second set of images, the overall composition produced by LapStyle appears softer. The visual elements of the starry sky from the style painting are more pronounced in the Gatys-style result, indicating that the fusion of style and content is less effective compared to LapStyle. This demonstrates that the Laplacian loss effectively suppresses unnatural and abrupt local changes in the stylized image relative to the content image, thereby significantly enhancing the quality of style transfer.

5.2 Experiments on our methods

5.2.1 Display of results

In this section, we will demonstrate that the quality of our algorithm's results significantly surpasses those of Gatys, LapStyle, and AdaIN. For the two examples in Section 5.1, I will separately present the results of LapStyle, AdaIN, and our algorithm.

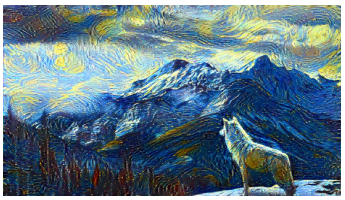

图 7: Lapstyle

图 8: AdaIN

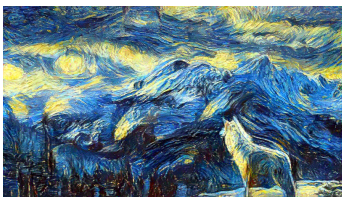

图 9: Ours

It is evident from the results that our algorithm produces the highest-quality images. In the first set of images, both AdaIN and our method result in smooth transitions in the upper-left corner's clouds, avoiding chaotic blotches. However, compared to AdaIN, our results exhibit superior stylization. This is particularly noticeable in the depiction of the clouds, where our results clearly reflect the swirling patterns reminiscent of the Milky Way in Van Gogh's Starry Night. In contrast, AdaIN's results only capture similar colors but fall short in embodying the distinctive style of the starry sky. In the second set of images, the advantages of both AdaIN and our method over LapStyle are even more apparent. Not only do both methods achieve more

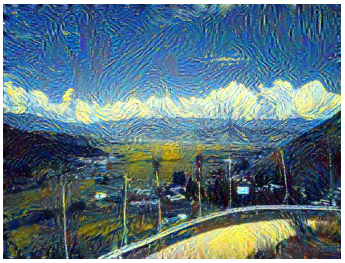

图 10: Lapstyle

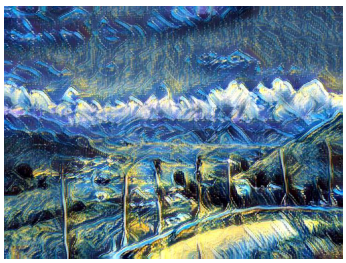

图 11: AdaIN

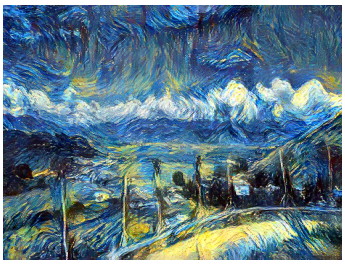

图 12: Ours

harmonious style-content fusion, but the stylized results also align more closely with Van Gogh's Starry Night. Regarding the details of the clouds, our results stand out further. The swirling patterns in the upper part of the clouds in our images are more pronounced and better capture the essence of the starry sky style.

5.2.2 Analysis of the results

Next, we will provide a rational analysis to explain why our algorithm performs better. We believe the key lies in the differences in the loss functions. AdaIN employs a relatively simple style loss, relying only on basic statistical measures such as mean and standard deviation. LapStyle and similar approaches adopt a more complex style loss, which can be seen as the norm of the difference between covariance matrices. In our view, achieving similarity in more complex statistics results in greater alignment with the original style's details but often introduces distortions in the finer aspects, deviating from the content image. Conversely, focusing on simpler statistics may lead to insufficient adherence to the style details but has the benefit of reducing distortions and disharmony in the finer aspects.

Our algorithm effectively combines the advantages of both approaches. The content image and style image are first transformed through a pre-trained neural network into an image that maintains good details, achieves harmonious style-content fusion, and closely resembles the overall style of the style image. This intermediate result is then refined using the LapStyle algorithm to add the finer stylistic details of the style image. This two-step process not only significantly accelerates convergence but also improves the quality of the final output.

6 Conclusion

In this project, we implemented and enhanced style transfer methods, integrating AdaIN and LapStyle to achieve superior style-content fusion, detail preservation, and stylistic quality compared to classical approaches. Our method effectively balances content fidelity and stylistic detail by combining the strengths of simple and complex statistical measures in style loss functions. From this project, we learned to effectively combine different approches and combine its advantages to maximize results. Future application of this method can be used to for more user friendly image styling, especially when combining with state-of-the-art tricks.

References

[1] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks."Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[2] Li, Shaohua, et al. "Laplacian-steered neural style transfer."Proceedings of the 25th ACM international conference on Multimedia. 2017.

[3] Huang, Xun, and Serge Belongie. "Arbitrary style transfer in real-time with adaptive instance normalization."Proceedings of the IEEE international conference on computer vision. 2017.

[4] `https://cocodataset.org/#download`

[5] Li, Chuan and Michael Wand. "Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 2479-2486.

[6] `https://github.com/titu1994/Neural-Style-Transfe`

[7] Tian Qi Chen, Mark Schmidt, "Fast Patch-based Style Transfer of Arbitrary Style" `https://doi.org/10.48550/arXiv.1612.04337`

[8] Shaohua Li et al. "Laplacian-Steered Neural Style Transfer" `https://dl.acm.org/doi/abs/10.1145/3123266.3123425`

[9] Yingying Deng et al. "StyTr2: Image Style Transfer With Transformers" Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 11326-11336

[10] Songhua Liu et al."AdaAttN: Revisit Attention Mechanism in Arbitrary Neural Style Transfer" Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 6649-6658

[11] Xinyuan Chen et al. "Gated-GAN: Adversarial Gated Networks for Multi-Collection Style Transfer" in IEEE Transactions on Image Processing, vol. 28, no. 2, pp. 546-560, Feb. 2019, `doi:10.1109`