Multiple Image Style Transfer

Yicheng Li
a1698732

Abstract

Extracting the style of image is a difficult image processing task, the previous research provided a method that can generate a new image that the style from one image and the content from another image. However, if we reuse this method to combine more than one style, it is hard to found all style feature in the output image and the transfer process cost too much time, therefore, in this paper, I modified input processing function and style loss calculation function to provide a new method, it performs better when combining multiple image style, especially, there are lower style loss and time cost.

1. Introduction

In the late 1960s, computer vision began at universities, and it is a part of artificial intelligence. Then, after many years development, the importance of computer vision technique has been increasing, especially, now this technique is widely used in daily life, for example, face detection and image classification. In comparison with human, sometimes computer can better understand the content of the image. Moreover, recent research suggests the computer could comprehend image style better than human as well, using computer vision and neural network can abstract the style from an image, and transfer this style into another image.

This paper will focus on the multiple image style transfer, in detail, I will extract multiple image style from different images by using Convolutional Neural Network, then transfer these style into a new image and keep the content of the new image, only style will be changed.

2. Background

2.1. VGG

The Convolutional Neural Network is the fundamental of the image analysis, therefore, a reasonable network structure can benefit to analysis result. Simonyan and Zisserman (2015) [2] provide a deep convolutional neural network structure which aim to improve the image classification accuracy. As figure 1 illustrated, the depth of these structures are from 11 layers to 19 layers. Then these network structures are implemented in the classification problem, the test result from Simonyan and Zisserman (2015) [2] suggests that the classification error rate is decreasing while the depth is increasing as figure 2 showed.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 1. VGG Test Result[2]

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

Figure 2. VGG Structure[2]

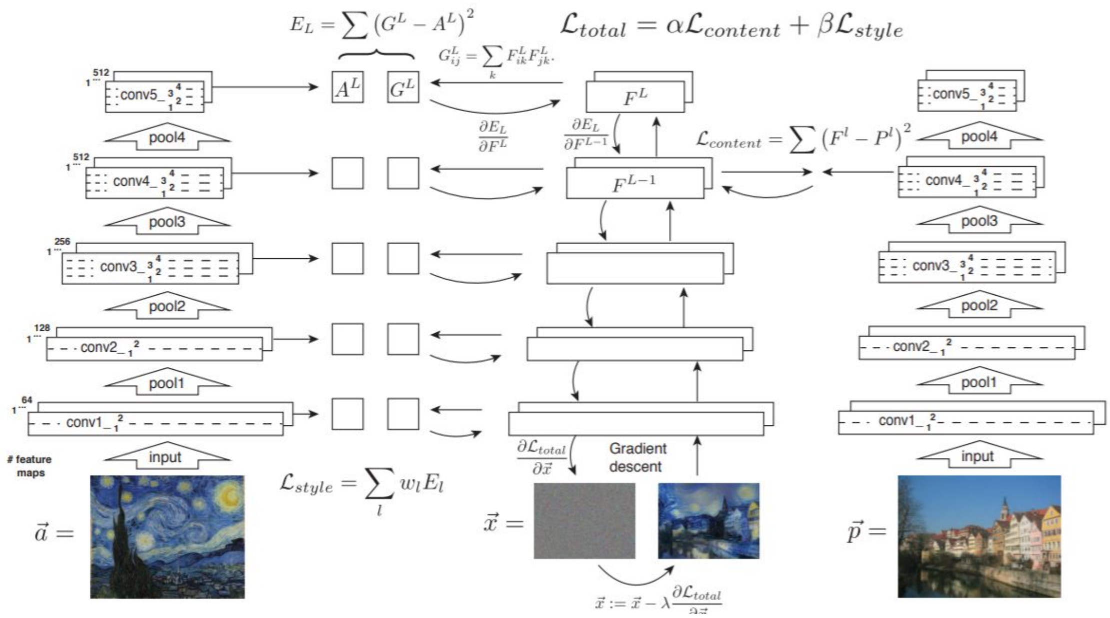

Figure 3. Style Transfer [1]

2.2. Style Transfer

With the development of computer vision, some people found the VGG19 structure is not only can be used in image classification, but also it can be used in image style transfer. Gatys (2016) [1] suggests an algorithm based on VGG19 that can transfer image style. In order to achieve style transfer, the image style and content should be extracted at first, after a list of tests, the result shows that when we input an image into VGG19 network, in higher layers of this network, the high-level content information of the image is preserved and other detail information is lost. In addition, in terms of image style, each convolutional layer will generate feature maps, then the Gram matrix can be used to calculate the correlation between these feature maps. the result can represent the style of this image. The following is the definition of the Gram matrix.

$$G^l = F^l (F^l)^T.$$

Then, the style transfer can be achieved image can as we can extract the content and style. As figured 3 showed, if we input 2 images, a content image and a style image into VGG19 network, the content information can be extracted from high-level convolutioanl layer output, such as the conv4 layer. Then, the style information can be extracted by using Gram matrix. Finally, the program will generate a new white noise image, then, using gradient iteratively update this white noise image, try to decrease the content loss between this image and content image, decrease the style loss between this image and style image. at last, the white noise image will be changed and it will have the content from content image and the style from style image.

3. Multiple Style Transfer

In this paper, I mainly focus on multiple image style transfer, in detail, multiple style will be extracted from more than one image, and these style will be transfered into one image.

3.1. Previous Style Transfer Function

Firstly, there is a simple way to achieve multiple style transfer by using the Gatys (2016) [1] method, a content image and the first style image will be input, then a combination image will be generated and this image will be inputted into model again with the second style image, as a result, the second combination image will be generated which will have both styles from the first and second style images. In addition, the style weight will be set to 0.5 if we try to transfer 2 style, and style weight will be set to 0.3 when we combine 3 style. As figure 4 illustrated, we input an Adelaide university image as the content image and a thunder image as the first style image, so the first combination image will preserve the building information, but the style is transferred into thunder style, Then, the combination image is inputted into network again with a fire style image, after the second combination, the third style-water style will be added.

The image that includes multiple style can be generated by using this simple method, however, the result could not meet the requirement, as figure 4 showed, the output image that only combines with 1 style perform better than combines with 2 or 3 style. For example, the first combination can clearly display the thunder feature, however, after adding the fire style, most of the thunder feature is replaced by fire feature, only few thunder feature can be found in the second combination image, then adding the third style, the thunder feature is almost disappeared.

In order to clearly display how the thunder feature change, the style loss calculation that introduced in Gatys (2016) paper [1] will be reused, and the style loss between thunder image and the combination image will be set as the a indicator to evaluate the output image, As figure 5 showed, when a new style is added the style loss between output image and thunder image significantly increased, when adding 3 styles the style loss slightly decrease, it might because the style weight is set to 0.3 as we have to combine 3 style. But in general, it could be considered that the thunder feature quickly disappeared when there is new style added, in addition, the style of combination image is mainly related to the last input style image. If users try to keep all styles, they have to carefully modify the style weight before input every style image, but it might be difficult.

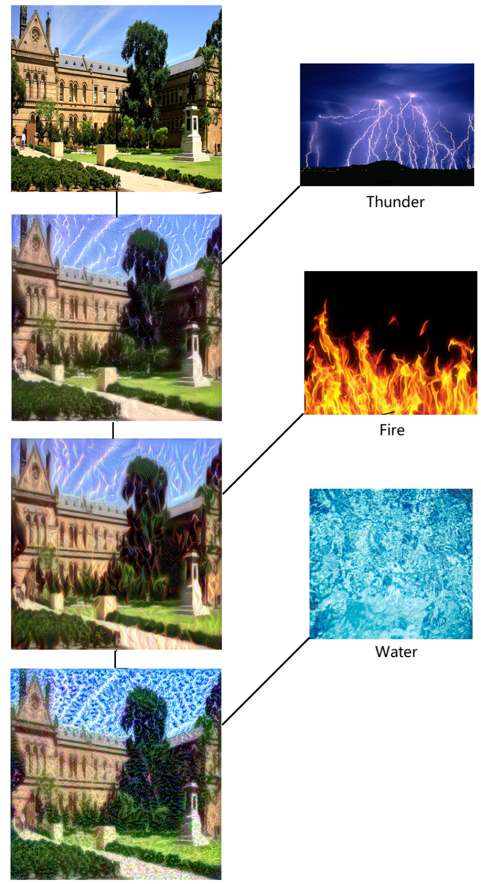

Figure 4. Previous Style Transfer

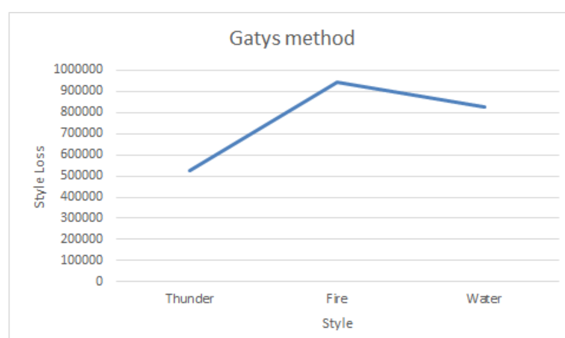

Figure 5. Thunder Style Loss

3.2. New Multiple Style Transfer Function

As using the previous function could not generate an good quality image with multiple style, I modified some parts of the previous function, the new multiple image style transfer function is still based on Gatys (2016) [1] method and VGG19 model, especially, the calculation of content loss will be reused. In terms of style part, I modified the input image function and loss calculation function. Firstly, in order to extract style from different images, all style images should be combined before input into the VGG19 model. I use the TensorFlow as a backend, when loading an image, at first, the image tensor shape will be set to (Height, Width, 3) as the image colour is formed with RGB, thus there should be 3 channel to store RGB. Then, the tensor shape should be expanded to (index, Height, Width, 3), as a result, it is easy to concatenate all style images as figure 6 displayed, I set the content image as the base image and set the index to 0, then, generate a white image as the combination image and set the index to 1, the index of style images are set from 3 to n.

Then, the style loss calculation function will be modified in order to calculate style loss based on all style images, the calculation process is :

$$\sum_{j=styleImage1}^{styleImageN} \frac{\sum_{i=conv1}^{conv5} \frac{(G_c^i - G_j^i)^2 * styleWeight}{(width*height)^2*layerSize}}{N}$$

G is the Gram matrix of the image, N is the number of style images. In order to found the whole style loss, firstly, we will calculate the sum of each layers style loss from conv1 to the conv5 for different style images, then find the average style loss of all style images, finally, we will still use a gradient to update the combination image to decrease the loss.

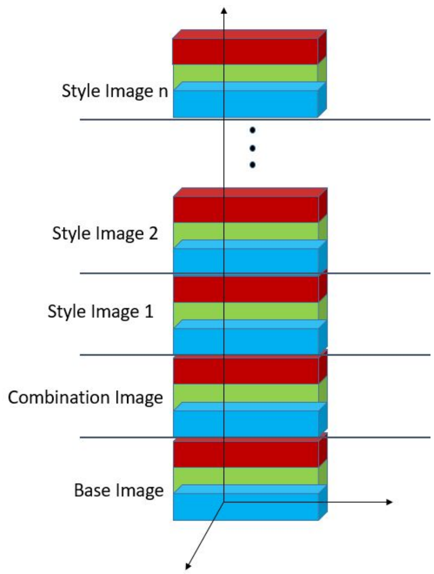

Figure 6. Image Concatenate

4. Experiment

Figure 7. Multiple Style Transfer

In this section, I will setup a experiment to compare whether the new method can generate a better output image. The previous test has combined 2 and 3 styles, and result shows that the thunder style is decreasing while inputting new style images, therefore, we will use new multiple style transfer method to do transfer style and generate 2 images, one combines with thunder style and fire style, the other combine with thunder style, fire style and water style. Then, the result will be evaluated by using style loss between the thunder image and the combination image, if the style loss is much less when we implementing new method, it could be considered the new method can better preserve style than the previous method in terms of multiple style transfer.

Figure 7 demonstrates the multiple style transfer result, the new method might be better than simply using Gatys (2016) method [1], even there are 2 or 3 different styles combined, each style also can be found in the final output image. For instance, the thunder style still can be found in the combination image no matter how many image style added. figure 8 also proved this, the thunder style loss is much less than the previous result when adding a new style. Therefore, it could be considered that the new style transfer function performs better in terms of multiple style transfer problem.

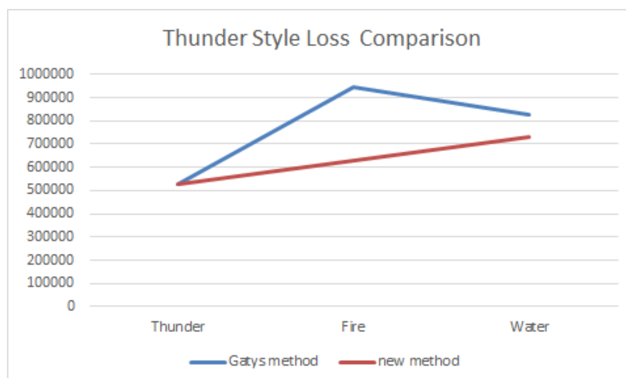

Figure 8. Style Loss

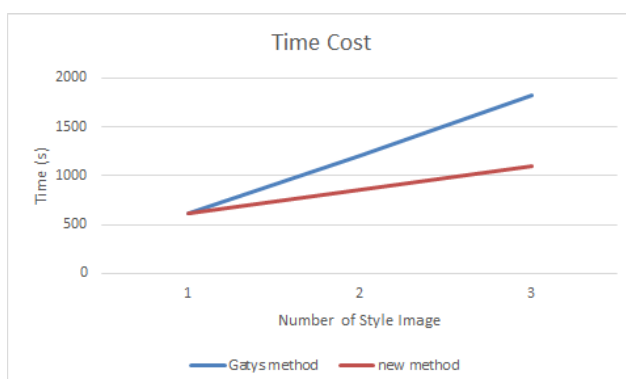

Figure 9. Time Cost

In addition, the new method also can improve the transfer efficiency because all style images will be inputted and calculated at once, when we use the previous method, the computer will calculate style loss every time inputting a new style image. As figure 9 showed the time cost, the new method spends less time than the previous method when doing 2 or 3 style transfer.

5. Conclusion

In conclusion, concatenating style images before input into the neural network and simultaneously calculating all style loss can generate a better quality image that includes multiple style in comparison with the previous method, especially, there is a significant decrease in style loss and time cost. However, so far, the transfer process still cost too much time, and there are some recent research focus on improving the transfer efficiency by normalizing input data, therefore, in future, it could be used to keep decreas time cost.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2414–2423. IEEE, 2016.

[2] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.