Final Report

Multiple Artistic Style Transfer

Zhihui Liu Zhengxiang Zhu

1 Introduction

1.1 Problem Description

The goal of our project is to learn the content and style representations, and then to combine them to generate new pictures. In the paper [1] and [2], Gatys et. al. performed style transfer using CNN which combined one content with one style image only. We are going to extended his approach by generating the style transfer network with multiple styles.

With current deep learning techniques, given two images, it is perfectly possible for a computer to recognize the specific content of the image. But the style of an image is a very abstract thing. The human eye can very effectively distinguish the styles of different painters of different genre paintings. For a computer, essentially, those are some pixels. The essence of a multi-layered network is actually to find out more complex and more intrinsic features, so the image style can theoretically extract some interesting features that may be contained in the image through a multi-layer network.

According to the method proposed in the [1] and [2], the network structure for style transfer consists of two parts. One is "Image Transform Net" and the other is "Loss Network". The Image Transform Net input layer receives an input image, and the final output layer output a picture. The overall model is divided into two phases, the training phase and the implementation phase.

1.2 Motivation

As we know, the art style is a very abstract thing. Sometimes artists absorb the ideas of others and then create a new paint style. From this point of view, we are thinking that we may create a new paint style with this method.

1.3 Open questions in the domain

Transferring the style from one image onto another can be thought as a texture transfer problem. The goal of texture transfer is to synthesis a texture from a source image and preserve the semantic content of a target image. Most previous texture transfer algorithms rely on non-parametric methods for texture synthesis in this domain. While, the limitation of these researches is that they use only low-level image features of the target image to inform the texture transfer.

1.4 Summary of approach

Based on the previous work [2, 6], we will combine multiple artistic style artworks onto the content image and use a pre-trained model (VGG-19) to capture low level features and transfer them to the content image.

2 Background

As we mentioned before, our project is mainly based on the approach in the [1] and [2]. Here's main idea of these two papers.

In simple terms, we are given two pictures, of which we want to extract the style picture for our style image, and another picture for which we want to extract the content is our content image.

We build three convolutional neural networks, one of which is used to extract the features of the style image, and the other is used to extract the features of the content image. The last neural network initializes a random noise image and continuously iteratively updates the image by making a gradient descent. Our image to generate our final result map.

The following figure shows the visual representation of the content image and style image given by the author in different levels of the neural network (VGG19) architecture. The results show that the visual reconstruction of the content image is almost indistinguishable from the original image in the low level of the neural network.

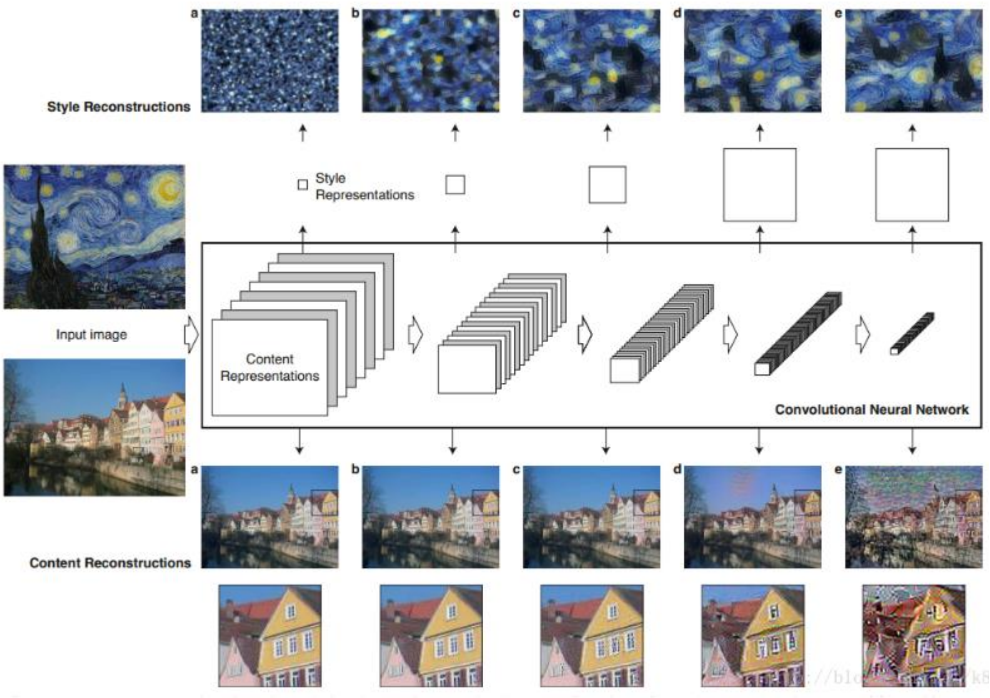

Figure 1 Image representations in a Convolutional Neural Network

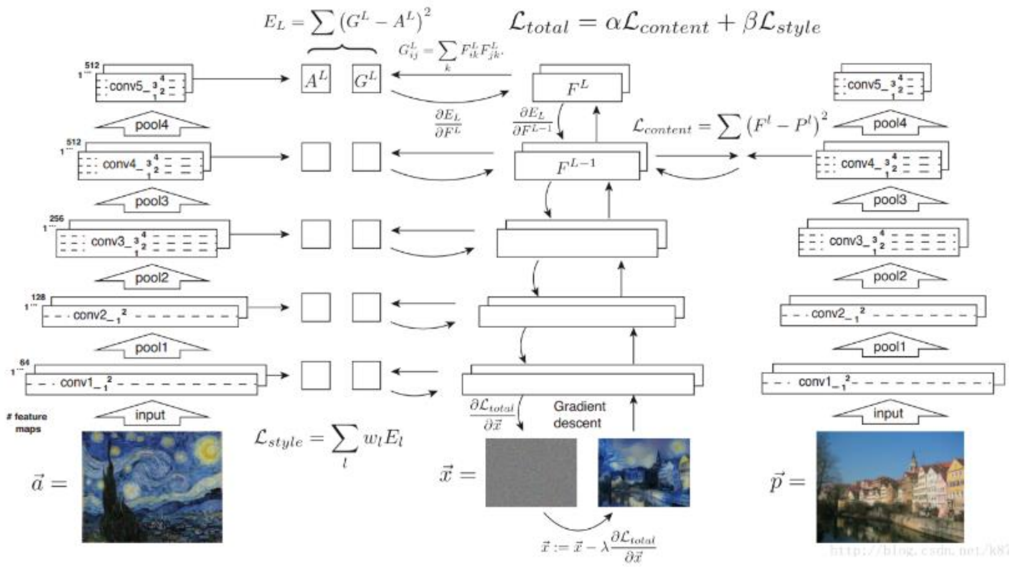

Figure 2 The main flow chart of the algorithm

Given a style image a and an ordinary image p, the style image will get many feature maps in each convolution layer when passing through VGG-19. These feature maps form a set A. Similarly, the ordinary image p passes VGG-19 will also get a lot of feature maps, these feature maps form a collection P, then generate a random noise image x, random noise image x through VGG-19 will also generate a lot of feature maps, these feature maps constitute a collection G and F correspond to sets A and P, respectively. The final optimization function is to adjust x so that the random noise image x finally looks both the content of the normal image p and the style of a certain style image a.

content representation

Before establishing the objective function, we need to give some definitions: In CNN, if a layer contains N_l filters, then N_l feature maps will be generated. Each feature map has dimension M_l and M_l is the feature map. The product of height and width. Therefore, the set of feature maps for each layer can be expressed as $F^l \in R^{N_l \times M_l}$, where F^l_{ij} represents the activation of the i-th filter at position j.

So, we can give the cost function of content:

$$L_{content}(p, x, l) = \frac{1}{2}\sum_{ij}(F^l_{ij} - P^l_{ij})$$

style representation

In order to create a style representation, we first use the Gram matrix to represent the relationship between each feature maps of each layer. $G^l \in R^{N_l \times N_l}$ and G^l_{ij} are the inner products of feature maps i,j:

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l$$

Using the Gram matrix, we can create a cost per style for each style:

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

Combine all layers to get the total cost

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

Finally, combining the content and style costs will eventually result in:

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

3 Method

In this project, we are inspired by paper from Gatys et al. [1, 2] and the open source [6]. Based on their outstanding work, we will provide our new idea that combined multiple style to generate a new artistic style. A pre-trained model (VGG-19) is used to capture low level features and transfer them to the content image. Figure 3 shows the structure of VGG-19 network which contains 16 convolutional and 5 pooling layers. This model is publicly available and can be download in [5].

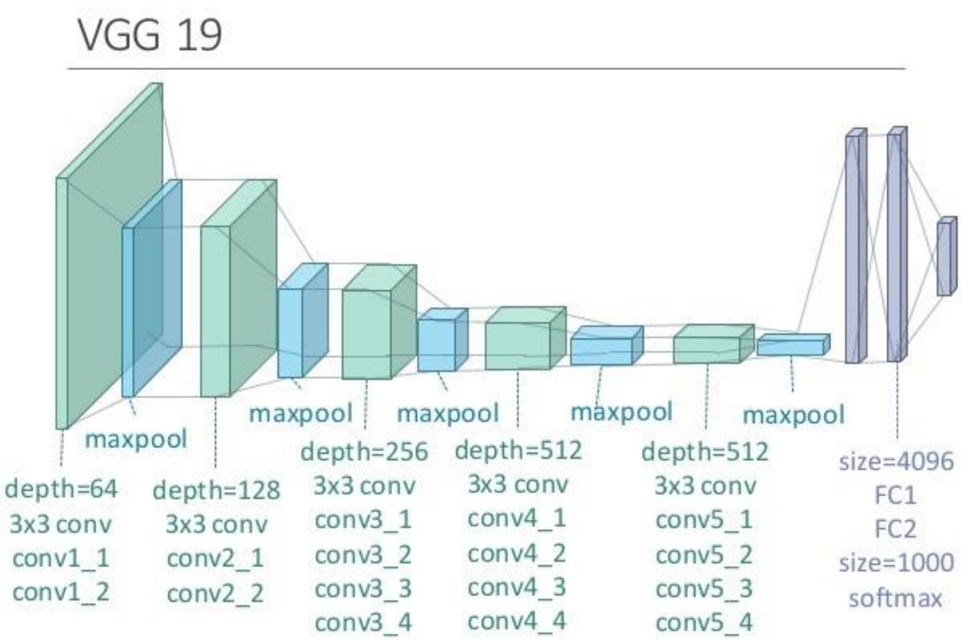

Figure 3 The structure of VGG 19 network

In the section 2, we have introduced the algorithm in [2] that used to obtain a content and a style representation of an input image. In order to transfer the style of an artwork onto a photograph, Gatys et al. [2] 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 the painting defined on a number of layers of the Convolutional Neural Network. The loss function as below:

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

To add one more style of artwork onto the photograph and generate a new artistic style, we have new loss function here:

$$L_{total}(p,a,x) = \alpha L_{content}(p,x) + \beta [L_{style1}(a,x) + L_{style2}(a,x)]$$

4 Experiment

The framework is built in python using TensorFlow with GPU support. TensorFlow run significantly faster on a GPU than on a CPU. The image processing is often time consuming. Therefore, we will implement the framework on the computer with NVIDIA GTX 980 graphics card, i7-6700K CPU and 32G RAM. We choose a picture of the Japanese actress, Ryōko Hirosue as the content image and one style of artwork: Girl before a Mirror from Pablo Picasso, one style of artwork: Cafe Terrace at Night from Vincent van Gogh as the style image in the experiment.

Figure 4 shows the result of the output image when combine the content image with the style image (Girl before a Mirror, Pablo Picasso).

(a) (b) (c)

Figure 4 (a) Girl before a Mirror, Pablo Picasso, 1932. (b) The original photograph. (c) The output image.

(a) (b) (c)

Figure 5 (a) Cafe Terrace at Night, Vincent van Gogh, 1888. (b) The original photograph. (c) The output image.

Figure 5 shows another example we transfer the style of a photograph (Cafe Terrace at Night, Vincent van Gogh) onto the content image. The synthesized image resembles the colors and painting style of the style image and displays a special style of the content image. We set 4000 steps to train the model. A output image is saved every 200 steps. From Figure 6, we can find that the output image becomes change little after training 2000 steps.

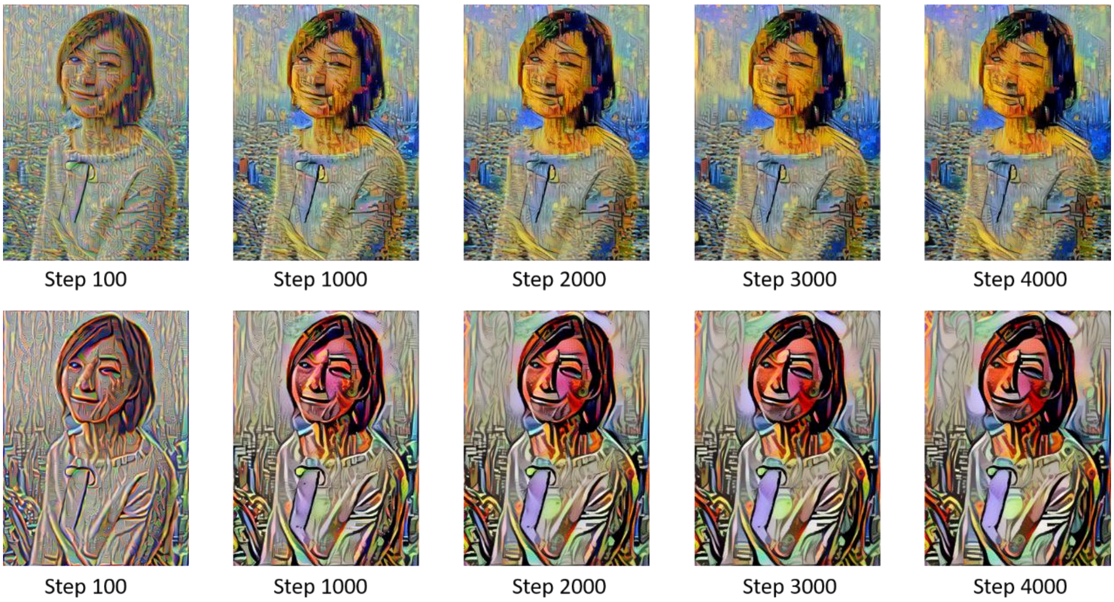

Figure 6 The output image in different steps of iteration

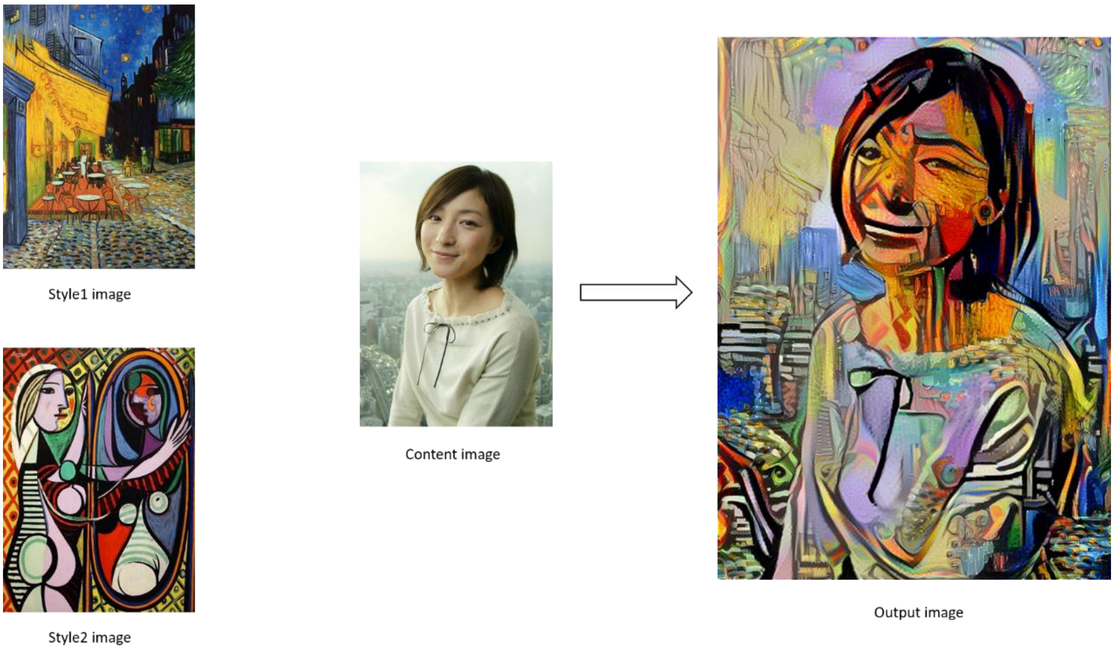

Figure 7 Combine the content image with two styles of artworks

The focus of this project is on multiple artistic style transfer. Thus, we combine two styles of artworks that have been used to generate output image individually in the previous experiments. The new artistic style image is shown in Figure 7. In Figure 8, we list three artistic styles images and the new artistic style image (in the middle) shows that it can capture the features from two artistic styles images very well.

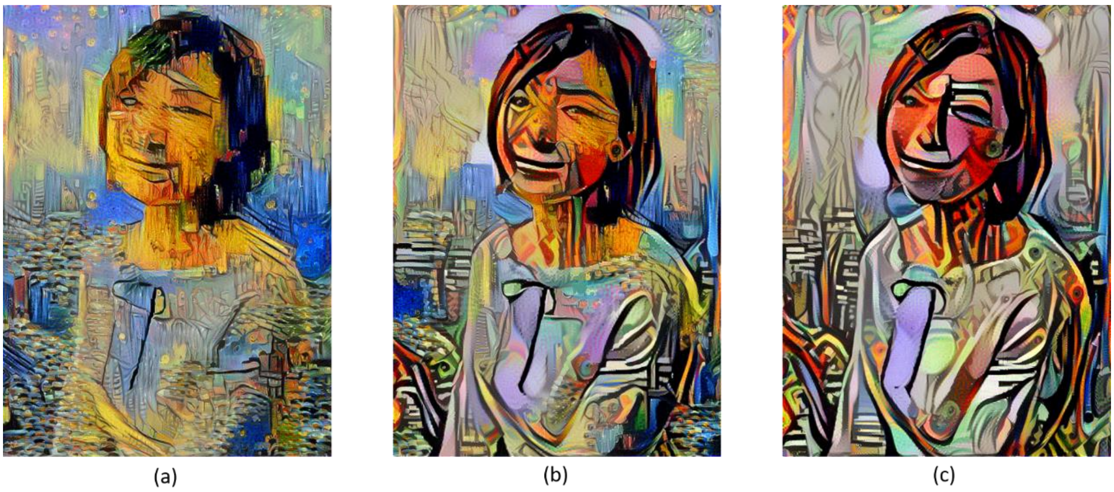

Figure 8 (a) Output image with Cafe Terrace at Night. (b) Output image with two styles of artworks. (c) Output image with Girl before a Mirror.

5 Conclusions

Transferring the style from one image onto another is a texture transfer problem. In our project, we follow the work from Gatys et al. [1, 2] and the open source [6]. We provide a new idea that combined multiple style to generate a new artistic style. The result shows that the new artistic style image can capture the features from two artistic styles images very well and displays a special painting style. However, as mentioned in Gatys et al. [2], we only consider style transfer to be successful when the output image 'looks like' the style image and shows the objects of the content image. The evaluation criterion is not mathematically precise and maybe cannot be agreed universally. Therefore, it is hard to say our multiple artistic style image beats the single artistic style image. Anyway, we have learned how to use machine learning method to solve the texture transfer problem and reviewed the implementation of Convolutional Neural Networks and TensorFlow in the project. And the last but not the least, we have a lot of fun in this procedure.

6 Future works

Future works may focus on different style transfers on different part of the image, such as transferring the face to Monet, while changing her cloth to Picasso. We are thinking to do it with separating the image into several parts like cloth and facial regions, and applying an algorithm on those two regions separately. It is interesting to extend the application of this work to other stages of building a fresh new artistic style.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, A Neural Algorithm of Artistic Style. Computer Vision and Pattern Recognition, Aug 2015.

[2] 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, Jun 2016.

[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

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

[5] VGG-19 model, https://github.com/BVLC/caffe/wiki/Model-Zoo

[6] Implementing of a Neural Algorithm of Artistic Style, https://github.com/log0/neural-style-painting/tree/ff05c7f3c1bd058d3dc10158f023e18182e9a497