Neural Artistic Style Transfer

Lei Guo, Songnan Zhang

{$lg2681, sz1982$} @$nyu.edu$

Abstract. We consider artistic style transfer, where an image, most likely a photo, is transformed into an output image with a specific artistic style. We studied two popular implementations, *Gatysetal.* proposed a method in 2015 which is to optimize the combination of content image and style image by jointly minimizing the feature loss using $per - pixelloss$ and a style loss using $perceptual - loss$ extracted from a pretrained convolutional network; however, *Johnson*'s method is to train feed-forward neural networks purely using $perceptual - loss$ which is also based on high-level features extracted from pretrained networks. In this paper, we reproduced both methods, and since *Johnson*'s method produced models which are more generalized, we mostly experimented on this one. After each experimental step, we also compared the results with which produced by *Gatysetal.*'s optimization-based method.

Keywords: Artistic style transfer, image transformation, neural network

1 Introduction

In this report, we present two implementations of neural artistic style transfer.

In fine art field, human painters are able to create paintings with unique styles. The artistic style of a painter is like a signature dissolved in his/her works. With some knowledge of art history, one can easily tell whether a painting belongs to the works of Vincent van Gogh, Salvador Dali or Pablo Picasso. The style of a painting can be extracted from the original art work and combined with the content concept of other images. This process is called artistic style transfer.

Artistic style transfer problem is a classic image transformation task. The system will extract style information from the style input image and content information from the content input image. The style and content will be merged into one single output image as it is painted using the artistic style of the style input while carrying the same content of the content input. The abstract concept of style information and content information can be obtained using Deep Neural Network.

In this paper, we reproduced two popular methods. One of them is based on $per-pixel$ loss between output and ground-truth images, and most importantly, since this method does not train a $transformer - net$, so it does not produce a reusable model; however, another one is based on $perceptual-loss$ based on high-level features extracted from pretrained networks and the $transformer - net$

produces models which are reusable for any other content images. We experimented both methods and compared the results between the two after each experimental step. Fig. 1 shows stylized results of *Gatysetal.* and *Johnson*'s implementations after applying the south park style on a cat image.

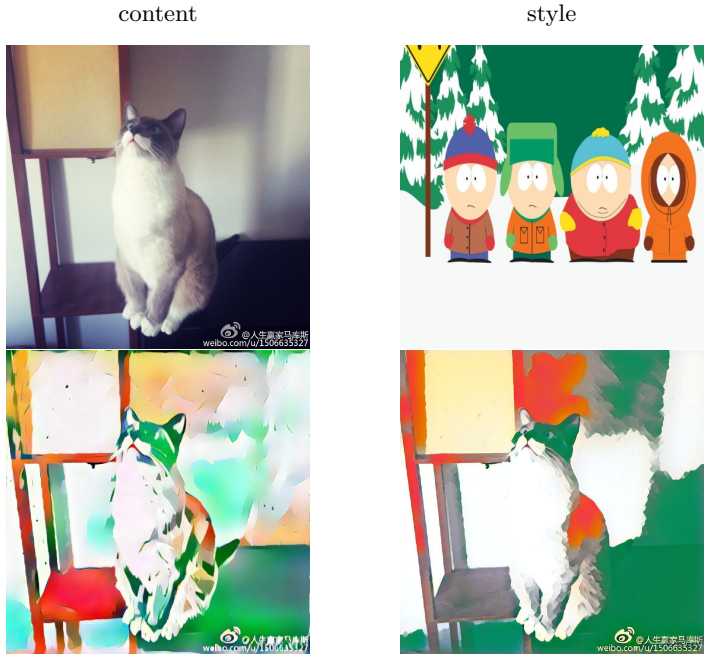

Fig. 1: Example results for style transfer using *Gatysetal.* and *Johnson*'s methods.

2 Related Work

Recently, a lot work has been done on neural style transfer.

The first neural style algorithm is developed by Leon A. Gatys, Alexander S. Ecker and Matthias Bethge in 2015[1]. Their method optimized the combination of the content of one image with the style of another by jointly minimizing the feature reconstruction loss using $per - pixelloss$ and a style reconstruction loss using *perceptualloss* extracted from a pretrained convolutional network for image classification. This algorithm generates desirable results with high perceptual quality.

In 2016, Johnson Justin, Alahi Alexandre, and Fei-Fei Li[2] published their method for real-time style transfer combining the idea of training feed-forward

convolutional neural networks[3, 4] with defining and optimizing perceptual loss functions based on high-level features extracted from pre-trained networks. Their method established feed-forward convolutional neural network for each style image. New images generated by the convolutional neural network have god perceptual quality.

Comparing to Gatys method, the feed-forward convolutional neural network from Johnsons method is reusable. As a consequence, once the training for that style is done, the on-line style transfer time is significantly reduced (around three orders of magnitude faster).

Our project is mainly focusing on reproducing, comparing and experimenting on both Gatys and Johnsons methods.

3 Method

3.1 Loss functions

The basic idea is to define two parts of loss functions: content loss L_c and style loss L_s. L_c measures how different the content is between input and arbitrary content image, while L_s measures how different the style is between input and arbitrary style image. Optimization is towards minimizing the weighted sum of these two loss functions Let X be an image. $Cnn(X)$ is the network fed by X. Let FXL in $Cnn(X)$ be the feature maps at depth layer L. By definition FXL is the content feature of X at layer L. Thus, the difference of content features at layer L between two images X and Y is defined as:

$$D_C^L(X, Y) = \|F_{XL} - F_{YL}\|^2 = \sum_i (F_{XL}(i) - F_{YL}(i))^2$$

Let FXL_k with $k <= K$ be the vectorized k_{th} of the K feature maps at layer L. Let GXL be the gram produce of all vectorized feature maps FXL_k with $k <= K$.

$$G_{XL}(k, l) = \langle F_{XL}^k, F_{XL}^l \rangle = \sum_i F_{XL}^k(i).\, F_{XL}^l(i)$$

Now we have $GXL(k, l)$, which is a measure of the correlation between feature maps k and l. That means GXL represents the correlation matrix of feature maps of X at layer L. Thus, difference of style feature at layer L between two images X and Y is defined as:

$$D_S^L(X, Y) = \|G_{XL} - G_{YL}\|^2 = \sum_{k,l} (G_{XL}(k, l) - G_{YL}(k, l))^2$$

3.2 Gatys method

A pre-trained VGG network with 19 layers ($VGG19$) is used in Gatys' method and the design diagram is shown in Fig. 2[1].

In order to minimize the combined content loss and style loss, the gradients of each distance at each wanted layer are computed and combined.

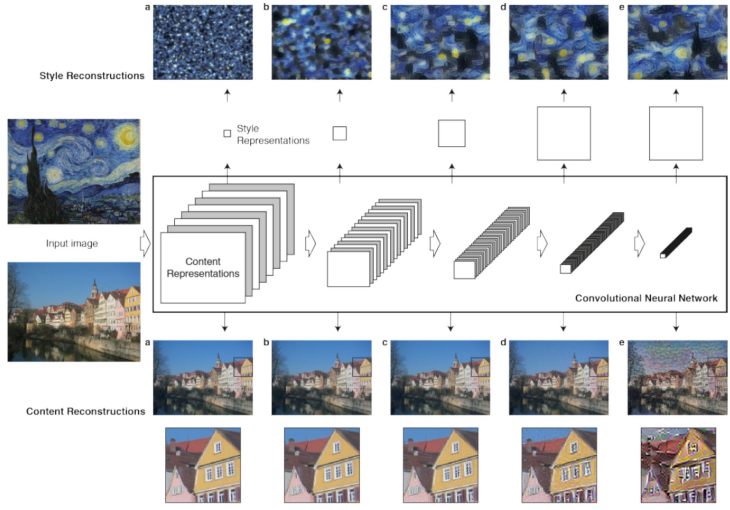

Fig. 2: Diagram of Gatys' method

$$\nabla_{extittotal}(X,S,C) = \sum_{L_C} w_{CL_C} \cdot \nabla^{L_C}_{extitcontent}(X,C) + \sum_{L_S} w_{SL_S} \cdot \nabla^{L_S}_{extitstyle}(X,S)$$

Where L_c and L_s are loss respect to the wanted layers (arbitrary stated) of content and style and w_{CL_c} and w_{SL_s} are the corresponding weights. Then, we run a gradient descent over X to achieve the optimal result:

$$X \leftarrow X - \alpha \nabla_{extittotal}(X,S,C)$$

3.3 Johnson's method

A pretrained $VGG16$ neural network is used to determine feature loss in this method. The design diagram is shown in Fig. 3[2]. In Johnsons method, a

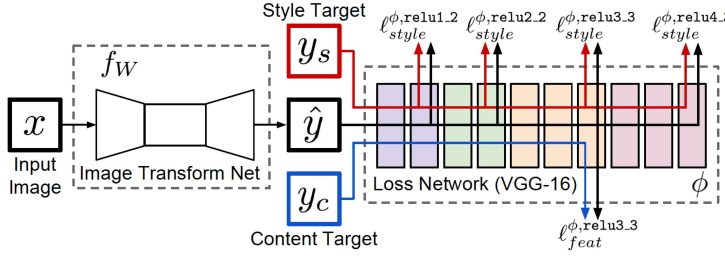

Fig. 3: Diagram of Johnson's method

feed-forward transformation networks for image transformation is trained using stochastic gradient descent to minimize a weighted combination of loss functions:

$$W^* = \arg\min_W \mathbf{E}_{x,\{y_i\}} \left[\sum_{i=1} \lambda_i \ell_i(f_W(x), y_i) \right]$$

4 Experiments

We reproduced both methods, and applied them on several different styles.
Framework: Pytorch.
Training set: training set of *CoCo*_2014.
GPU: P100 GPU on NYU HPC.

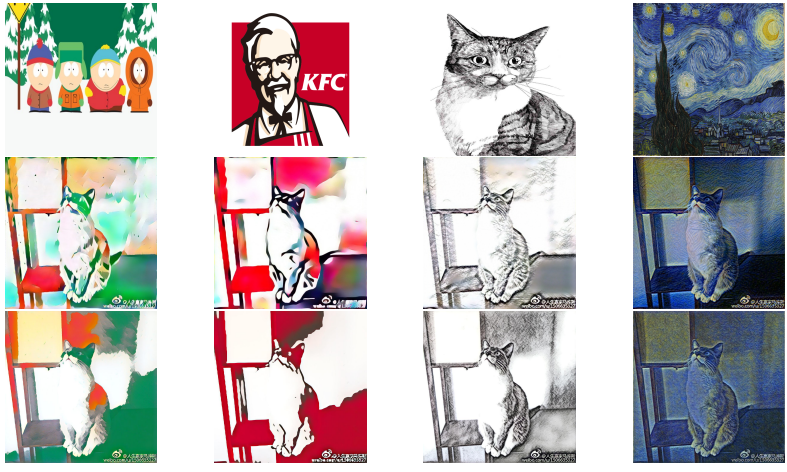

Fig. 4: Different styles transfered using both methods. First row are style images. Second row are results of *Gatys*' method. Third row are results of *Johnson*'s method.

Since *Johnson*'s method produced models which are reusable, we mostly experimented on this one. After each experimental step, we also compared the results with which produced by *Gatysetal*.'s optimization-based method.

4.1 *Gatys*' method

We tried using both white noise and arbitrary content images as input for the loss network. Fig. 5 shows the results.

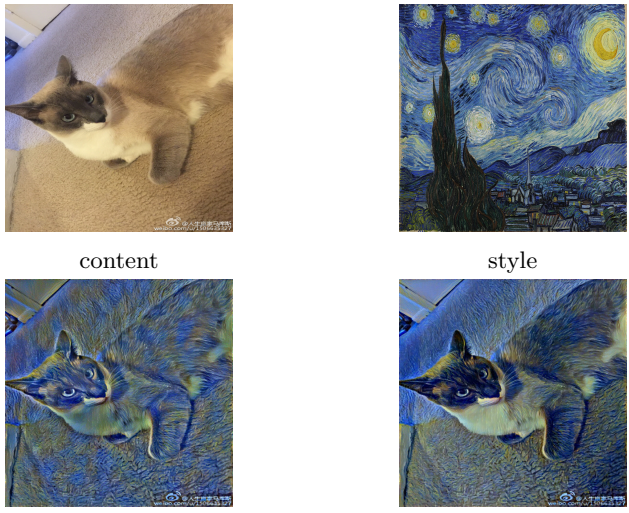

Fig. 5: Results using white noise or content as input

4.2 *Johnson*'s method

First, we tried to figure out the optimal layer to choose for content loss calculation. The result is shown in Fig. 6. We found that ReLu2_2 is the optimal layer to choose. Using layer ReLu3_3 cause unexpected color lumps. And this phenomena becomes severer when ReLu4_3 is applied.

Fig. 6: Results using different layer from loss network for content loss calculation.

We also modified the batch size for training. However, there is no significant difference on either training time or evaluation results. Fig. 7 shows the results for the cat image.

batch-size = 4

batch-size = 6

batch-size = 8

Fig. 7: Results using different batch size for training.

5 Conclusions

In this paper, we reproduced two popular methods for fast artistic style transfer. Both methods produce perceptually good results.

The first method is more efficient when it comes to a single style transfer. In case of one style is repeatedly combined with different content images, the second method provide a reusable feed-forward neural network which dramatically decreased the on-line transferring time.

Several experiments are carried out on both methods. We found out that for the first method, when using content instead of white noise as input image, the content information is better delivered into the output. We also found that $ReLu2\_2$ is the optimal layer to choose for content loss calculation.

References

1. Gatys, L.A., E.A.B.M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)
2. J. Johnson, A.A., Li, F.: Perceptual losses for real-time style transfer and super-resolution. European Conference on Computer Vision (ECCV) (2016)
3. Dosovitskiy, A., B.T.: Inverting visual representations with convolutional net- works. CVPR (2016)
4. Radford, A., M.L.C.S.: Unsupervised representation learning with deep convolutional generative adversarial networks. ICLR (2016)