Study on Image Style Transfer

a refined neural algorithm based on CNN

Lei Yu
Math and Computer Science Department
Pennsyvalnia State University, Harrisburg
Harrisburg, PA, USA
funkyholic00@gmail.com

Sukmoon Chang
Math and Computer Science Department
Pennsyvalnia State University, Harrisburg
Harrisburg, PA, USA
sukmoon@psu.edu

Abstract

Traditional pixel-basesd methods of image style transfer usually focus on low-level style features and are computational expensive for ideal outputs. However as neural network become popular, a neural algorithm for style transfer has proved itself working well on this task. Here we propose and implement an refined algorithm based on the original neural algorithm for artistic style transfer, to cut down the number of convolutional layers used and simplify the model, but with ideally close performance.

Keywords—convolutional network; image processing; style transfer; image sythesis; nonphotorealisc; artistic style

I. INTRODUCTION

The emergence of CNN offers a completely new idea of image style transfer. It provides an end-to-end implementation which outputs ideal results without any preprocessing of the input. Related works of image style transfer using CNN have shown a key point that the content and the style of images can be separated[1] to some extent so that we can tune the weights to match two images, one of content, the other of style. The resulting image is perceptulally sound.

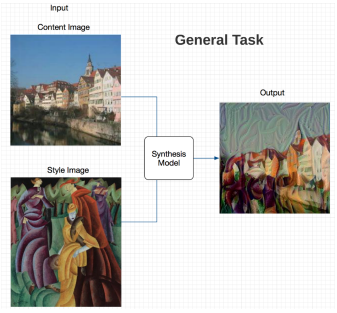

Figure 1: the general task to take input of two images and output a style-transferred image

However as both CNN layers and the image size go up, the computational cost becomes higher. In the original implementation using VGG-16 network, all 5 convoluational layers are used with an input image size of 224*224*3. With such a small resolution, there are still 10^5 pixel values to deal with for each filter in each layer. It will be a hugh cost for training, as every time it will compute trhough the whole network of CNN. Is there any way we can skip some layers to release the stress of computation? Here are two proposals:

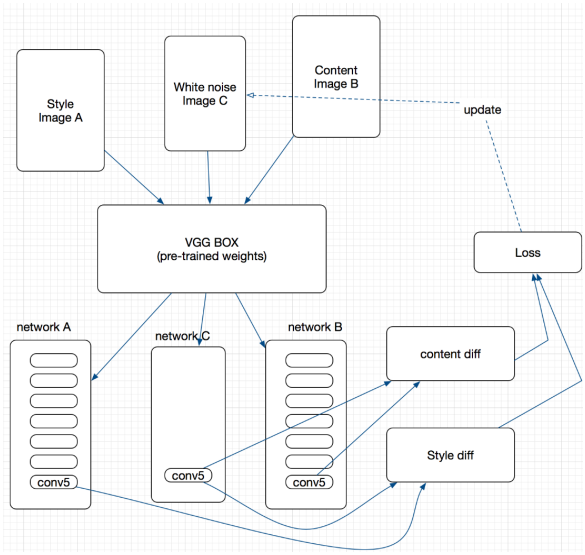

Figure 2: the neural algorithm in the original paper that goes through the whole network of CNN at each iteration. In order to update the content image, the image should go through the whole VGG(a pre-trained structured CNN) box.

As we go through the network(Figure 2), all activations(the resulting values in each convolutional layer are called activations), are computed. And the original paper used almost all these value. However as we know from the basic of CNN, different layers contains different details of the content and the style. So possible we could cut down the computations on some layers, just focusing on important layers to get similar results.

On the other hand the original paper starts the gradient descent on a white noise image in order to make unbiased

results. The algorithm will run slower with a smaller learning rate. Since we want an output image with both the style and the content, if we start updating from one of the input images, will it run faster to achieve the goal as we aleady have the content or style to start from?

This paper is meant to try an improved algorithm as a simplification upon the work of Gatys L A's to find a way of solving the two questions proposed above. Specifically the work in this paper will take only two input image, the content and the style image and try to update the content image based on the loss of some layers in CNN network.

II. RELATED WORK

Style transfer has always been a popular topic in image processing. It is both perceptually and technically cool to apply the style of an image to another, especially the one looks so different.

Traditional image processing methods focus on pixel-based implementations, in which each pixel of the original images are somewhat modified to simulate the style. However, it is for sure easy to change the style of an image but hard to extract the style of another. Literally, the representation of style is always manually defined, such as the parameters controling the size of brush that tunes the pixels. We can not tell easily and strictly what the style parameters of a certain image are. So, using traditional methods it is very hard to transfer the style we want.

As neural network emerges, the situation of image style transfer has been brightened. With end-to-end neural network, that is the neural network whose input is what we have and whose output is what we want, we do not need to care about how the style of a certain image actually is as long as we mean to lower the loss of the network, neural network itself will figure out the ideal parameters.

Among those popular implementation of neural networks, CNN[3] has been proved effective for image transfer. CNN is originally designed to do the image recognition problem. During the process of recognition, CNN extracts multi-level features throughout its network hierachy. These multi-level features are shown to be the combination of both content and style features which can be separated to some extent.

Separating contents from styles of an image is then proved to be feasible. In the original implementation[1], the content and style representation of an image with respect to different levels are reconstructed upon VGG[4] network, a structured CNN network trained to be ideal for image recognition. And results can be seen clearly that the higher level the style representation is reconstructed, the more perceptually smooth and general style it will be. The content reconstruction is on the contrary, the lower reconstruction the more accurate. So when we go up to the higher level in the network, the local content details of the image are lost but the global shape can be still kept.

In order to do semantic style transfer from one image to another, we need to make a balance, to keep both the content of one image and the style of the other. If we use the higher level, the style transferred can be sound but we will lose the shape of the content. If we use lower level, the shape of content can be kept, but the style transferred will not be good. In this sense, the style and content can not be both satisfied intensively, we need to tune the weights for tradeoff.

It turns out that the neural algorithm for artistic style transfer works well, especially for view images with a style transfer from paintings. However it does not work well on human face or other complicated images with many tiny blocks of color. And the transfer is somewhat randomized as we can not control the degree or area of transfer with an end-to-end neural algorithm. Also, if the style image contains multiple styles, it will work as a noise and result in unsound outputs.

In order to solve the problem, Li C, Wand[6] use the neural algorithm in a context-sensitive way to constrain the model. Based on Li C' work, Champandard A J.[2] use a annotation-based method to preprocess the image first to manually constrain the area of style transfer. It first separates the content image into multiple areas in terms of the smoothness of color. Each area is called an annotation, then a neural algorithm of style transfer is applied respectively in each annotation, resulting in sound transferred style in each area. Because we manually define the semantic information of the image, so the algorithm will not blend noise on the border of each area heavily, making the output image look sound.

Although, the above works have great performance on artistic style transfer, it does not work well on photorealistic style transfer. There are mainly two difficulties for doing style transfer on photorealistic images: 1.the spacial structure can be twisted since high-level content feature does not take good care of local content 2.semantic transfer could be mismatched, for example the texture of grassland could be transferred to the area of sky.

A method[5] exclusively working for photorealistic style transfer has been proposed by Luan F, Paris S, Shechtman E, et al. It solves the above two difficulties well. To avoid spatial distortion, they found a trans- formation of the pixel space which have a significant influence on image colors, but little influence on spatial features. To match the style transfer accurately in the correct area, they label the sub areas of the content image with corresponding label of the style image, so the grassland will not be transferred to the sky.

As for now, the neural style transfer algorithm is sophisiticated. We can produce really great images which perceptually make sense after the process of style transfer. But there are still problems, such as whether we can control the degree of being stylized? Specifically, whether we can control the parameters of stylization? Advanced research with insights into the neural model has been on the way.

III. METHOD

The basic idea of CNN is the key of implementation. Each convolutonal layer in CNN in fact does several convolution operations (Fig1) followed by some activation function like ReLU.

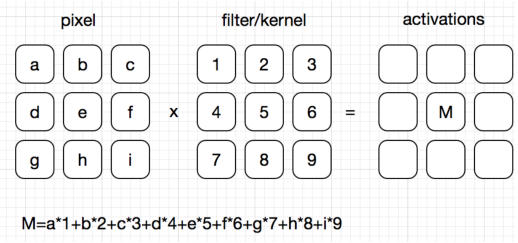

Figure 3: the convolution/correlation operation, note the convolution here has different definition from mathematical counterpart. The resulting values is the sum of the multiplication of all corresponding values. Usually a filter defines the weights, such as gaussian filter. The resulting values are call activations.

For a specific convolutional layer in CNN, it applies multiple filters on the input, and for each filter it will result an activation map. So for an input size of H*W*N, and k filters, and zero padding(a technique to maintain the size form input to output) the output size will be H*W*k (Figure 4).

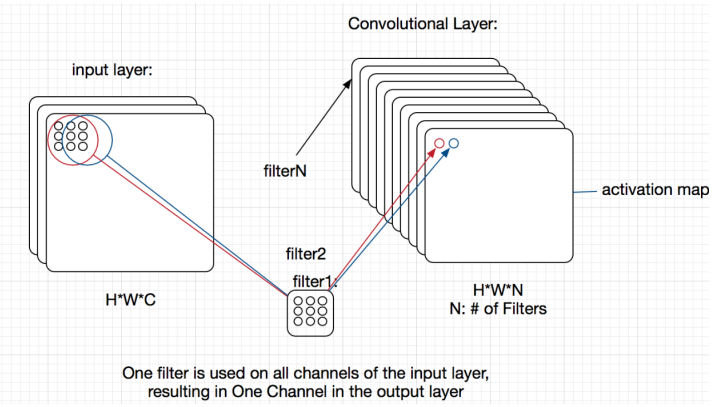

Figure 4: the explanation of the operation in one convolutional layer. For one filter, it slides the widget of filter on the input to get all values for one activation map.

Filters can be taken as a feature extractor, for example, a laplacian filter will detect the edges of the original image. So for one convolutional layer, there is a collection of extracted features. And as we go through the CNN network, the following convolutional layer extracts the feature of the previous covolutional layer. So the higher the level, the basic feature(feature of the original image) are more correlated.

Upon the above basis of CNN, we hereby define the content loss(Figure 5) and the style loss:

Denote A as the content image and B as the style image. And B_i is the ith convolutional layer of B, similar for B_i.

Content Loss: the difference between the corresponding convolutional layers (Figure 5).

The content loss CL of the ith covolutional layer is

$CL_i = \sum(A_i - B_i)$

Here $\sum$ means the sum of all difference of corresponing values.

So the total content loss for the whole network is

$CL = \sum(w_i * CL_i)$, where w_i is the weighting factor.

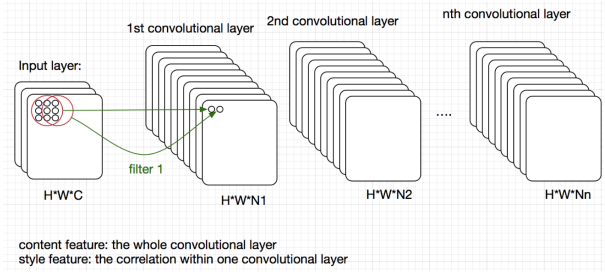

Figure 5: the representation of content loss, here the convolutional layer itself represents the content. So the loss measures the difference between the content of content image A and the content of style image B.

Style Loss: the difference between the gram matrix of the corresponding convolutional layers (Figure 6).

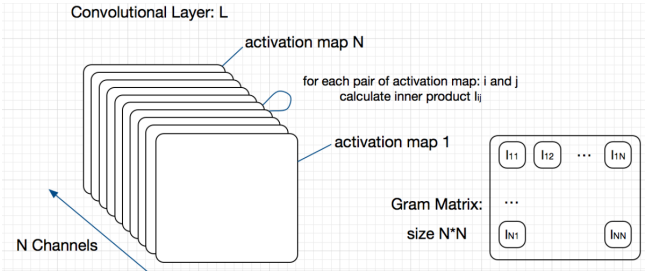

Figure 6: the representation of style loss, here for the ith convolutional layer, it's gram matrix Gi is the matrix of inner production of each pair of activation maps. Gram matrix is a useful representation of image styles as it reserves the correlations among different features.

So the Style loss is computed in this way:

$SL_i = \sum(G_i(A) - G_i(B))$, where $G_i(A)$ means the gram matrix for the ith convolutional layer of image A.

Specifically, if there are k activation maps in the ith layer of image A, and each activation map is of size N, that is N=H*W, the number of pixels, then

$G_i(A)_{(p,q)} = \sum(M_p * M_q)$, where p and q are indices, and M denotes the activation map.

And the total style loss should be summed all over the network:

$SL = \sum(z_i * SL_i)$, where z_i is the weighting factor

Upon the style loss and the content loss, we derive the final total loss.

Total Loss = a*CL+b*SL

Here a and b are the tuning weight for balancing the match of content and style. If we feed a large b, then style loss will become more inportant, so the image will pay less attention to keep its local content, thus become more abstract with the style transferred.

Restate the learning task:

The general task(Figure1) of implementation in this paper is to take the input of two images, one of content image, the other of style image, and tries to update each pixel in the content image to make it output closer activations in the CNN as the style image does.

We hereby propose a sythesis model(Figure 7) which does the end-to-end job, that is, without any extra manual preprocessing, the algorithm will take input of the original images and output ideal results.

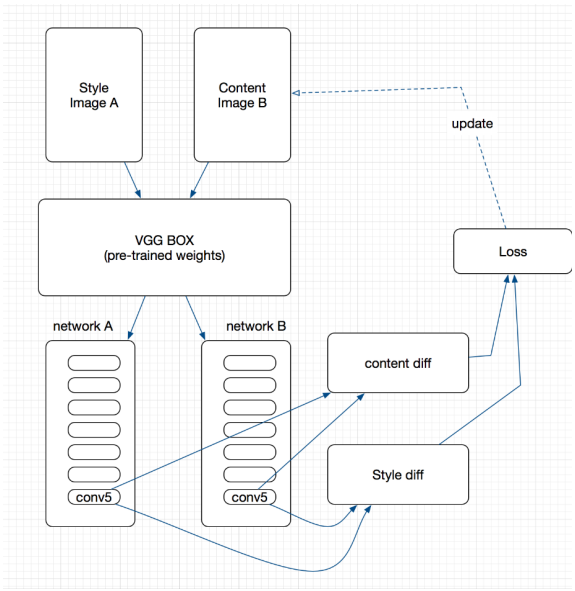

Figure 7: the model proposed by this paper, instead of having 3 input images, one more white noise for updating, this model takes two inputs and update from the content image. And instend of using all convolutional layers to compute the loss, we just use some layers like conv5.

Specifically, when we skip some layers i, the corresponding weight w_i and z_i are set to zero, so they no longer contribute to the total loss.

A clear hierarchy of computation can be seen below(Figure 8).

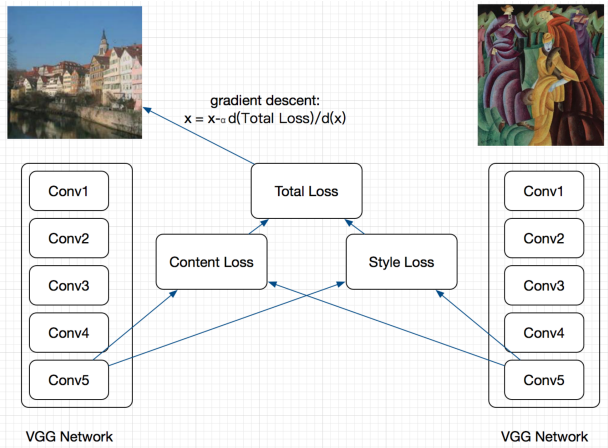

Figure 8: a clear look of the computation hierarchy.

By calculating the gradient of total loss with respect to x, the pixel value of content image A, the algorithm will update the actual pixel value.

The style image is also called the reference image which should remain unchanged. So lterally, the algorithm is trying to modify the content image until it becomes more similar to the reference image iterms of style.

Test Cases:

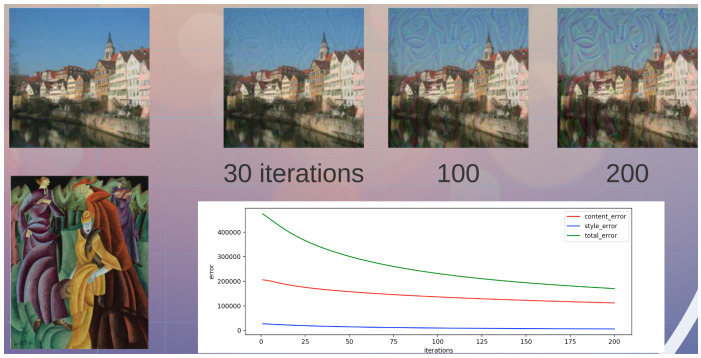

Figure 9: using conv1-2, conv3-3, conv5-3, with learning rate=0.00001, a=1, b=10. Running gradient descent under 30,100,200 iterations.

As we can see in the plot (Figure 9), the loss goes down steadily. With more iterations, the content image tends to be more similar to the style image below in terms of the style. Since we set a=1, b=10, with the ratio a/b=0.1, the content loss still affects significantly.

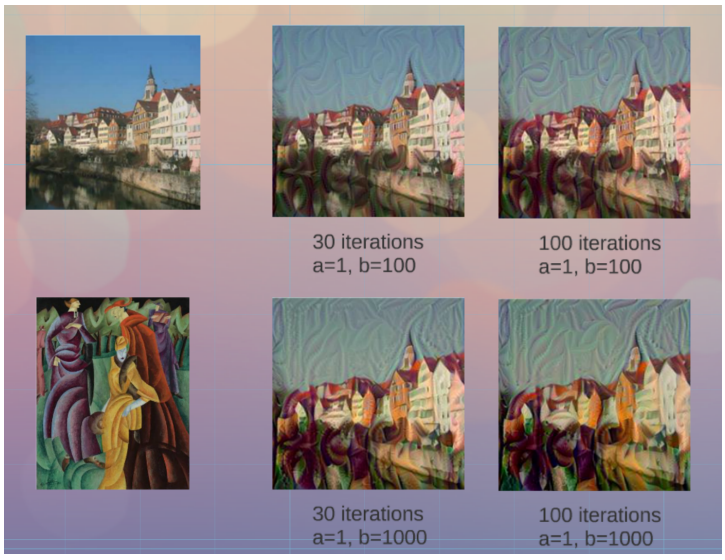

Figure 10: using conv1-2, conv3-3, conv5-3. with learning rate a=0.00001, and a=1, b=10 or b=100, or b=1000

As shown in Figure 10, the ratio of a/b affects greatly, when b is larger, which means more contribution of style loss, the

output image becomes more stylized to the extent that the shape of local area of the content image is distorted. But the global shape is kept, since we can still figure out for sure that it is the house.

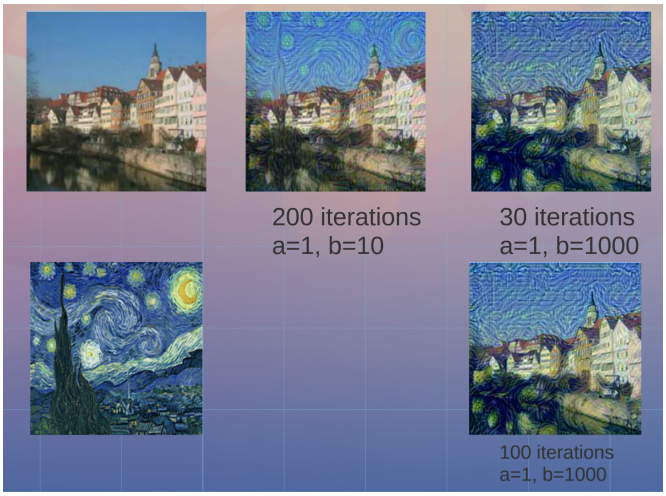

Figure 11: more test cases using conv1-2, conv3-3, conv5-3with learning rate = 0.00001

It is clear to see that if we use larger a, more contribution of content loss, the content image will try to match the content of style image. As we can see in the plot(Figure 11), the output image in the middle top has a shaow of the content of the style image. The content of the style image is transferred.

If using smaller a relatively, say smaller a/b, then the style loss has more influence. So it is clear in the right top image, that the style is transferred.

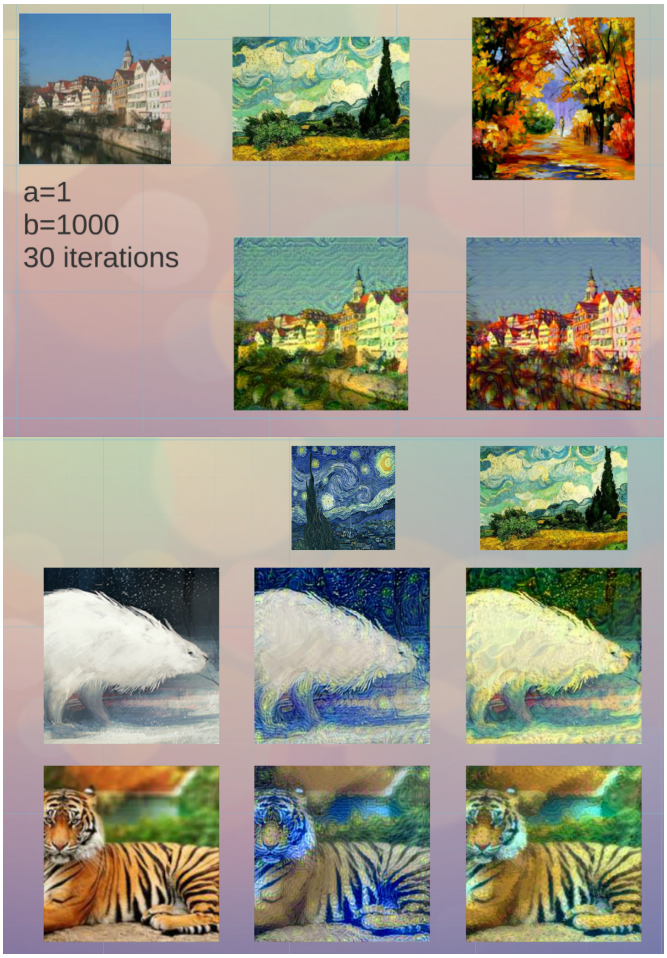

Figure 12&13: more test cases

CONCLUSION

By cutting down some convolutional layers we still successfully generate some perceptually meaningful images that have both the content of content image and the style of style image.

The refined algorithm implemented in this paper has achieved following conclusions:

1.we can still transfer the style by computing the loss over half of the convolutional layers.

2.the algorithm works well on artistic style transfer

As for artistic style transfer, the actual pixels are strictly to be accurate, so there could some glitches as long as the global style matches.

There are also some drawbacks which should be focused in the future:

1.The image generated is not smooth.

This happens especially when we use use high a/b, the style loss will have a great effect on the total loss, resulting in more stylized images, but the style transfer is not smooth. So probaly more iterations should be run to see if it helps. Moreover, we should also tune the weights of a and b to get a better performance.

2.It is still subject to the input images.

As said before, the algorithm for photorealistic style transfer works poorly. So there is still much space for improvements.

ACKNOWLEDGMENT

Thanks to Dr.Jeremy Blum, Dr.Sukmoon Chang and all others that offered advice and help.

REFERENCES

1. Gatys L A, Ecker A S, Bethge M. A neural algorithm of artistic style[J]. arXiv preprint arXiv:1508.06576, 2015.
2. Champandard A J. Semantic style transfer and turning two-bit doodles into fine artworks[J]. arXiv preprint arXiv:1603.01768, 2016.
3. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.
4. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
5. Luan F, Paris S, Shechtman E, et al. Deep Photo Style Transfer[J]. arXiv preprint arXiv:1703.07511, 2017.

6. Li C, Wand M. Combining markov random fields and convolutional neural networks for image synthesis[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2479-248