Style Transformation Basing on Convolutional Neural Network

Yu Shun Lin
Northeastern University
Electrical and Computer Engineering
Boston, USA
lin.yus@husky.neu.edu

Sicheng Ke
Northeastern University
Electrical and Computer Engineering
Boston, USA
ke.s@husky.neu.edu

Tianchu Li
Northeastern University
Electrical and Computer Engineering
Boston, USA
li.tianch@husky.neu.edu

***Abstract*—This document introduces the style changing of drawing piece depends on Convolutional Neural Network (CNN). Consider an image transfer problem that the original image is transferred into desired image. Feed forward Convolutional Neural Network is proposed to solve this problem via training. The desired style of image could be generated by defining and optimizing perceptual loss function which based on high level features extracted from pretrained network. Combination of two approaches for our method of style transformation. The result shows that it is similar optimization-based method. The simple-image super-resolution also gives visually results. Finally, we desire that the weight of the style could be adjust for the mixture style image.**

Keywords—Convolutional Neural Network (CNN), Image style transformation, Feed-Forward Neural Network, VGG network, Deep Learning, Super-Resolution

I. INTRODUCTION

There are a lot of existing problem for image transformation. A system received an image and transfer to another style image. Super-resolution, and colorization provide methods with inputting a noisy image, but outputting in a high-quality image. With image segmentation and image depth estimation, the former methods implement the transformed output scene.

One of the approaches to solve the transformation problem is to train a feed-forward convolutional neural network in a supervised manner. Calculate the loss function for each pixel for measuring the difference between base-image and the output. This has been used by Dong *et al* for super-resolution [1], by Cheng *et al* for colorization [2], by Long *et al* for segmentation [3], and by Eigen *et al* for the depth and surface prediction [4]. This combination of method only needs a forward pass through the trained network because approaches are that efficient. Though it only passes the trained network, however, the losses function used in the method so not get the perceptual difference between output and the base-image. Take the two identical image coordinates from each image, no meter how they are similarly, their perceptual measurement would be very different if measure by each pixel's losses calculate by loss function.

Recent work has shown that the high-quality image can be generate by *perceptual loss function* via the pretrained convolutional neural networks. They also use the minimizing loss function to approach the generation. This strategy has been used in feature inversion [5] by Mahendran *et al*. On the other hands, the visualization is provided by Simonyan *et al* [6] and Yosinski *et al* [7], and texture synthesis and style transfer by Gatys *et al* [9,10]. The method and implementation help to produce the high-quality image, but the speed is slow for calculation because the solving an optimization problem.

Our goal is to combine the benefits of these approaches. And create our own model and compare the different with the existing method. We train a feed forward *transformation* depending only on low-level pixel information because the training time would be long since solving an optimization problem. During the training, perceptual losses measure similarity more robustly than each-pixel losses, and the time measurement is in real-time.

The other goals are to Implement the style transformation by the former discussion. Compare with single-image super resolution. For the transformation, there is no single correct output.

We suppose we could input different style of image that the transformation could be alternative not only just for one style transferring. The result would be like the mixture of the style in different style.

II. Related Theory

A. CNN nerual network

The Convolutional neural network is the one of the main categories to do on image recognition. CNN neural network takes an image for 5-dimension in coordinate, red green m blue value for doing the recognition of objects. (Like cat, dog, etc.) Each input will pass through the convolution layers, with filters, pooling, fully connected layers and fitting the softmax activation function in the perceptron to classify an object probability in 0 to 1.

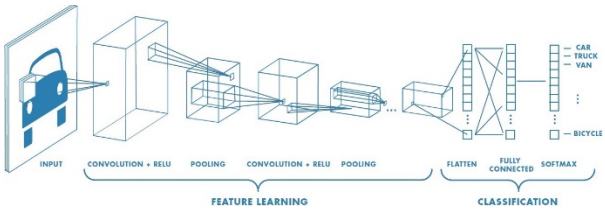

Figure 1. example of neural network including multiple convolutional layers.

There are a lot of method to generating the feature map. Such like SSD, NSSD function. CNN neural network took the filter multiplicate in matrix of center in each pixel to generate the feature map.

Strides is the number of pixels shifts over the input image matrix. Setting strides value 1 to move the filter to 1 pixel over each time, stride 2 for moving filter to 2 pixels for each time. Padding method is for the situation that the filter does not fit perfectly to the input image. Two option below could be selected.

- Pad the picture with zeros
- Drop the part of the image where the filter did not fit. This called valid padding which keeps only valid part of the image.

Fully connected Layer is to be flattened for the original feature matrix in to vector and feed in it.

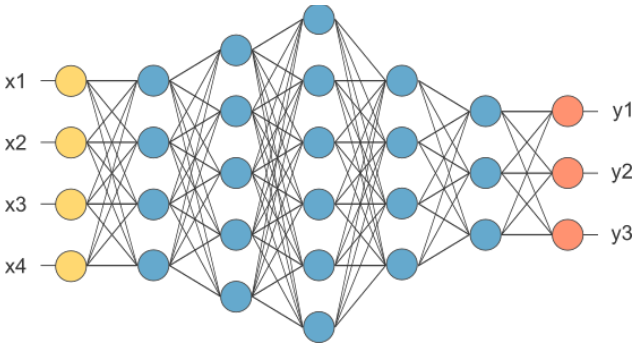

Figure 2, After pooling layer, flattened as FC layer

Then complete the CNN architecture with this fully connected layer. Combine the features to create the CNN model. Last, set activation function as sigmoid or softmax to classify the output as decided result. (objects)

B. VGG-19 pre-trained model

VGG-19 is a CNN network model released by Visual Geometry Group in Oxford University in 2014. This model contains up to 19 layers, which consist of 13 convolution and relu layers, 4 pooling layers, and 3 full connected layers. This model structure proves that the depth of net structure can improve the performance of whole model.

However, because of the too many layers in this model, it will take much more time to generate the result and consume more computing resource. Thus, in this experiment, only the parameters of weight and bias in the first 5 layers are used to do the style transfer. And the output of the 5th layer is used to calculate the loss of transfer for optimization.

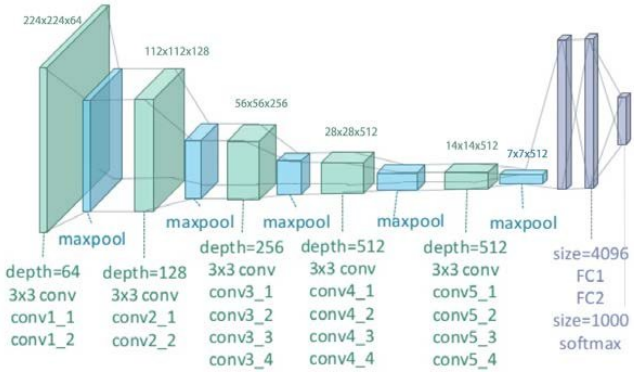

Figure 3. The structure of VGG-19 model

C. Style transfer of an image

In Gatys *et al.* [10] perform an artistic style transfer, which is the combination of a *content* and scenario style images. (The image has similar style.) This is the jointly minimizing features created of the loss which based on

features extracting which recognized from the pretrained network. Although the output is in high quality, but the spending is huge because it is depending on solving the optimization problem.

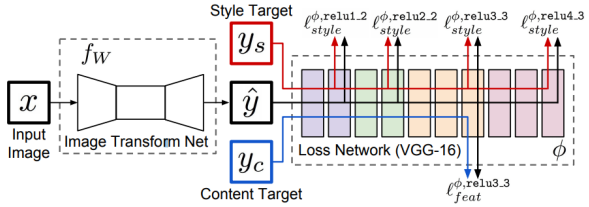

Figure 4. Overview of the system for the transferring the style of image. Loss network is pretrained for image classification to define *perceptual loss function.* Thus, the measurement could be reach between content image and style images.

D. Feed-forward image transformation

The method which training a deep convolutional neural network with each-pixel has been wide use in recent year for the image transformation.

Doing the segmentation of the image to classify the image in different area for classifying. When the training process, a dense scene label by running in the network mention in the previous call fully-convolutional manner of the input image, the each-pixel classification loss will be trained.

In the fees-forward model is trained for the using on each-pixel loss to transform the grayscale image to RGB image.

E. Combination of different style of images

To implement the combination of different style of image. Different method is provided in recent year. A different convolution neural network provides the different training beget to different desired output images.

One of the combinations is to create the image feature mask. Applying different style image on different segmentation. Another way is to distribute the styling weight in the parameter where in the pretrained network

III. METHOD

By implement the mixture of different style of an image, we decide to train a VGG16 (weight of input image matrix) to create the motivation output image. This is not only the image style transfer but also presenting the differential of style images.

A. Loss function

By style transfer, the system consists of two components: *image transformation network* and a *loss network.* Image transformation network is a deep trained leaving convolutional neural network's parameter named weights W. This weight decides the transformation from the input to the output. On the other hand, the loss function measured the difference between input and output. To minimize the loss function where weight is trained by using stochastic gradient descent with following equation.

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

B. Image transformation networks

The image transformation networks follow the architectural by Radford [11] without using any pooling layers. All of non-residual convolutional layers all of used in ReLU nonlinear activation function

C. Feature Reconstruction Loss

Without the parameter weight to exactly match to the exactly target output image, use the similar feature representations compute by the loss network.

$$\ell^{\phi,j}_{feat}(\hat{y}, y) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2$$

As feature map shape C*H*W. The feature representation is as the loss equation as former shows.

Reconstruct from the lighter layer that the image content and overall spatial structure are preserved but color, texture, and exact shape are not. By the reconstruction of feature loss, we could train our network to implement the image transformation.

D. Style Reconstruction Loss

After the feature reconstruction which penalizes the output. We desire the penalize difference in style with colors textures and common pattern. By Gatys *et al,* it propose by the *style reconstruction loss.*

$$G_j^{\phi}(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}.$$

E. Segmentation

The purpose of the Segmentation is to decide the mask of the different styling part for mixture the style with mask. This called the semantic segmentation. And by using semantic segmentation, we obtain the mask of an image. The dataset we use is from the MATLAB toolbox.

F. Vanilla Neural Style

Assume that image I and a style image S. To build the image of output X. Use the pretrained VGG model that is deep enough for the requirement of transformation. We choose vgg19 because the depth and the pretraining data is enough for the detecting objects in the image. The pooling is set to max for every layer.

With the neural style, we combine with capped gradient. Not only pass through CNN in every iteration but do twice. First complete the TV loss, and apply the gradient found on the whole image. In the second pass, complete the style loss, this only propagated to the desired segments of the image. This applies on the mask filer for creating the combination of mixture style transformation.

IV. EXPERIMENT

For trying different learning rates and different weight of style and content, we find that the learning rate should not be that frequently or the content image will get large loss function. Which means the result would be corrupt by the huge loss function.

Figure 5. image of style we try to use

Figure 6. show the learning rate influence the result of style transfer. Left with learning rate 200, right with 500.

Now trying the mixture style transfer. We take same weight for each input of the style to see what the best case will be.

Figure 7. Left is the original image. Middle is the transformation in 1 style image. Right is the combination of 3 different style image in different amount input images.

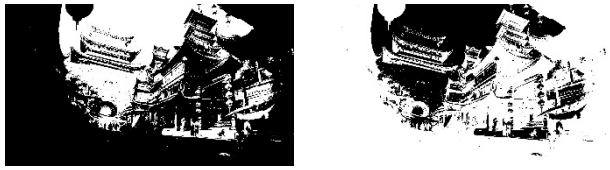

Figure 8, the two masks of the content image.

White regions will be transferred, and black area will remain. After transferred the original image for two different styles, multiply them with the mask respectively, and then combine them to get the mixture-styled picture.

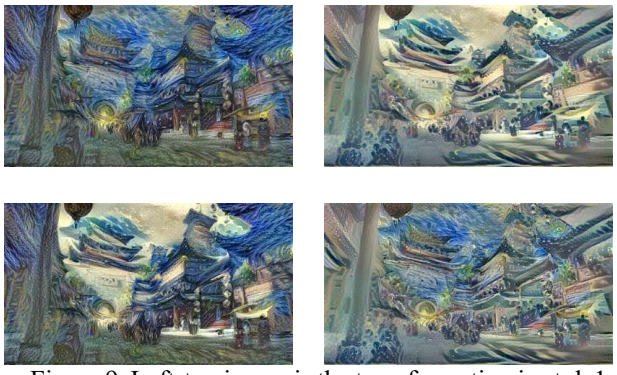

Figure 9. Left-top image is the transformation in style1. Right-top image is the transformation in style2. The two images in the bottom are the mixture of two style.

Figure 10. The left is the original image, middle is the transferred image without color preservation. And picture on the right is the result of color preservation.

To implement color preservation, we transferred the image to YIQ space which color information is restored in I and Q space. And then on I and Q space, matches the histogram of transferred image to the original image.

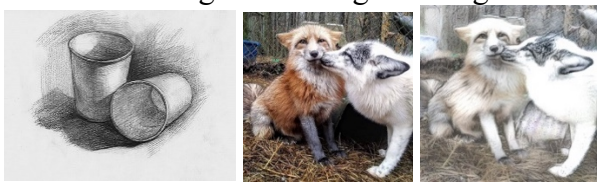

Figure 11. Input sketch picture for transfer to sketch picture.

The result of the sketch transfer did not show a good result because the color still maintains in the output picture. We designed two ways to solve this problem. The first is just convert the result into a grayscale image. The second way is to implement color preservation discussed previously. The result shows below.

Figure 12, left image shows the grayscale of original image. Right one shows the color preservation discussed previously.

Figure 13. Transformation of single style

In this case. We use the original wave style image transfer the fox image. We observe that part of the segmentation is not transferred cause the feature is confuse when learning the part of furs. Thus, the layer did not pass the style feature in the output image.

Besides, to explore how to improve the performance of our model, we use both Relu and Sigmoid activation function after convolution layer. The results are shown as figure 14:

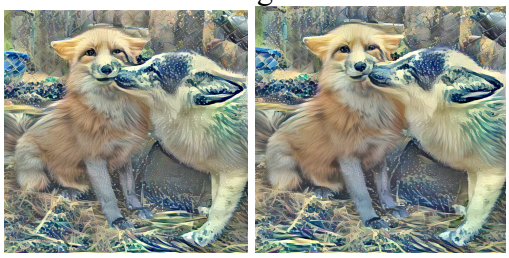

Figure 14. Result of Sigmoid function (left) and Relu function (right)

Through the comparison of these two results, there is little difference between these two functions. The performance of Relu is a bit better than Sigmoid. The reason may be in back propagation, sigmoid is easy to lead to “gradient disappear” because the gradient of sigmoids becomes increasingly small as the absolute value of x increases. And therefore, some information may be lost [9].

In addition, different weight of style loss and content loss can also result in different output. As shown in figure 15, it is obvious that with larger style weights and less content weights, the output will become contain more feature of style image and less its origin feature.

Figure 15. Style weight=100(left) Style weight=2000 (right)

V. CONCLUSION

We combined the benefit of feed-forward image transformation and optimization-based method for image generation with each type of loss function. Also, we have applied this method to style transfer via comparing the improve of speed with existing method.

In the future we would explore the use of perceptual loss function on another image transformation task. We also plan to do a random

learning program that could draw the artificial by itself not only just transfer from the training style samples.

We also find the features did not transform in some part of the image. Like the furs of fox. In future works, we think to separate the complex feature part and design a CNN network for complex feature learning method. Thus, we could apply to the method we have now.

On the other hand, we also planning to adjust the CNN network for transferring. Like adding some random drawing or object detection for guessing new object and adding to the output for another creative artificial.

REFERENCES

[1] Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. (2015)

[2] Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 415–423

[3] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. CVPR (2015)

[4] Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in Neural Information Processing Systems. (2014) 2366–2374

[5] Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2650–2658

[6] Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2015)

[7] Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H.: Understanding neural networks through deep visualization. arXiv preprint arXiv:1506.06579 (2015)

[8] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In Proceedings of the IEEE International Conference on Computer Vision, pages 1529–1537, 2015.