Style Transfer

--Project name: Skrull

Team name: Team name

Yuting Jiang
Department of Electrical and
Computer Engineering
UC San Diego
La Jolla, CA 92093
A53298624
yuj010@eng.ucsd.edu

Tianyu Zhao
Department of Electrical and
Computer Engineering
UC San Diego
La Jolla, CA 92093
A53308405
tiz007@eng.ucsd.edu

Wei Wang
Department of Electrical and
Computer Engineering
UC San Diego
La Jolla, CA 92093
A53316402
wew016@eng.ucsd.edu

Link to the Github:

Cycle-GAN Style Transfer: *https://github.com/TYZHAO/ECE-285-MLIP-Project-B-Style-Transfer*

Neural Style Transfer: *https://github.com/TYZHAO/ECE-285-MLIP-Project-B-Style-Transfer/tree/Baseline*

(Demos are included)

Abstract

This report is about our final project "Style Transfer" of UCSD's course "ECE 285" (2019 Fall). Basically, we implement two different techniques. One technique is "Neural Style Transfer" from Gates Paper "A Neural Algorithm for Artistic Style", and another one is "Cycle-GAN" from Jun-Yan Zhu's paper "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". We get good and interesting results after building, developing and applying these two techniques.

1 Introduction

There are a thousand Hamlets in a thousand people's eyes. If you give one topic to different painters, they must paint paintings of their own styles. Image style transferring is exactly about transferring different image styles of just one topic. There are lots of techniques used to transfer image style since 2014 such as Neural Style Transfer, Real-Time Style Transfer, GAN Style Transfer, Cycle-GAN Style Transfer and so on. Here we choose Neural Style Transfer and Cycle-GAN Style Transfer as our two techniques to realize image style transferring.

The first part of our project is to implement the style transfer application in Leon Gatys's paper: A Neural Algorithm of Artistic Style. As described in Gatys's proposal, we started with a pre-trained VGG-19 convolutional neural network (CNN). To learn the style of an artistic work, we built a new feature space that captures the style of an input image on top of the original CNN representation. The representation computes correlations between the different features in different layers of the CNN.

The second part of our project is to implement a Cycle-GAN for image to image style transfer. Our goal is to learn a mapping from source domain to target domain without aligned image pairs. Cycle-GAN is a popular method for style transfer and that can generate high quality results. The main difference between GAN and Cycle-GAN is that Cycle-GAN contains an inverse mapping from target domain back to source domain.

2 Method Description

2.1 Description of Neural Style Transfer

In reality, how should we rebuild the pre-trained VGG model to make it understand what the style features of a specific input artistic work are? In convolutional neural networks, asides from the overall structure of the network, the key factor is the loss gradient used to update the parameters in the network. By modifying the loss function used to perform back-propagation in the network, we can achieve different goals using exactly the same pre-trained network, in this case, VGG-19. In the previous assignments, we have implemented different loss functions for different tasks including image classification and image denoising. Luckily, in this project, Leon Gatys has defined the loss function we should use in his approach.

Leon Gatys defined content loss as following: Given the original image **p** and the generated image x, he defines P^l and F^l as layer l's corresponding feature, reshaped as a 2D matrix for the original image and the generated image. The content loss is given as following:

$$L_{content}(\boldsymbol{p}, x, l) = \frac{1}{2}\sum_{i,j}\left(F_{i,j}^l - P_{i,j}^l\right)^2 \tag{1}$$

For style-loss, we need to compute feature correlation given by the Gram matrix, G^l, where $G_{i,j}^l$ is the inner product between the vectorized feature map I and j in layer l:

$$G_{i,j}^l = \sum_k F_{i,k}^l F_{j,k}^l \tag{2}$$

In the end, we minimize the mean-squared distance between the entries of the Gram matrix from the original image and the Gram matrix of the image to be generated. Let a and x be the original image and the generated image and A^l and G^l their respective style representations in layer l. The contribution of the layer to the total loss is then:

$$\frac{1}{4N_l^2M_l^2}\sum_{i,j}\left(G_{i,j}^l - A_{i,j}^l\right)^2 \tag{3}$$

the total loss and the final loss are:

$$L_{style}(\boldsymbol{a}, \boldsymbol{x}) = \sum_{l=0}^{L} w_l E_l \tag{4}$$

$$L_{total}(\boldsymbol{p}, \boldsymbol{a}, \boldsymbol{x}) = \alpha L_{content} + \beta L_{style} \tag{5}$$

Now we can use this loss function to modify the original VGG network.

2.2 Description of Cycle-Gan Style Transfer

To learn the mapping between domain X and domain Y, we need to sample x from X and y from Y. The model includes mapping $\mathrm{G}: \mathrm{X} \rightarrow \mathrm{Y}$, $\mathrm{F}: \mathrm{Y} \rightarrow \mathrm{X}$ and 2 discriminators $\mathrm{D_x}$ and $\mathrm{D_y}$ that aims to identify generated images in each of the domain. The Cycle-GAN add an additional constrain based on GAN (Generative Adversarial Network), which is inverse mapping $\mathrm{F}\big(\mathrm{G}(\mathrm{X})\big) \approx \mathrm{x}$.

We apply adversarial loss to both mappings.

$$\mathrm{L_{GAN}}(\mathrm{G}, \mathrm{D_Y}, \mathrm{X}, \mathrm{Y}) = E_{y\sim Y}[logD_Y(y)] + E_{x\sim X}[\log\Big(1 - D_Y\big(G(x)\big)\Big) \tag{6}$$

G is trying to map a sample x from domain X to the domain Y and $\mathrm{D_y}$ is trying to distinguish generated image $\mathrm{G(x)}$ from sample y from domain Y. We have a similar adversarial loss for mapping $\mathrm{F}: \mathrm{Y} \rightarrow \mathrm{X}$.

We modeled image style domain X and Y as two different distribution, then if we map a sample x from X to Y and map it back to X the result should be identical to x, i.e. $F(G(X)) \approx x$. The cycle-consistency loss is formed as:

$$L_{cyc}(G,F) = E_{y\sim Y}[\| G(F(y)) - y \|_1] + E_{x\sim X}[\| F(G(x)) - x \|_1] \tag{7}$$

The final loss is:

$$L(G,F,D_X,D_Y) = L_{GAN}(G,D_Y,X,Y) + L_{GAN}(F,D_X,Y,X) + \lambda L_{cyc}(G,F) \tag{8}$$

Now we can use this loss function to modify the Cycle-GAN network.

3 Experimental Settings

3.1 Experimental Settings of Neural Style Transfer

We use dataset from Wikiart and Flicker like in Cycle-GAN.

The α to β ratio is 1/10000 and epoch number is 300.

When training and testing, the original images are mostly about nature scenes. The training set, testing set and validation set are basically separated by artificial selection.

3.2 Experimental Settings of Cycle-GAN Style Transfer

For datasets, we have conducted 4 pairs of style transfer from photograph to Van Gogh, Monet, Cubism and Pointillism. Two of them are from famous painters and another two of them are from two typical genres. Painting data are selected from Wikiart dataset and photograph are selected from Flickr dataset.

For network architecture, we use networks containing two stride-2 convolutions, 9 residual blocks, and two fractionally strided convolutions with stride $\frac{1}{2}$ for two generators. For the discriminator networks we use 70×70 PatchGANs, which are FCN fashion network. We also applied two common techniques to stabilize GAN training:

1. Replace the log likelihood loss function with a least-square loss which is more stable and steady while training.
2. The discriminator is updated with a history buffer of 50 previous generated images.

Same as the origin paper, in all experiments, we set $\lambda = 10$. We use the Adam optimizer with learning rate of 0.0002 and batch size of 2. All the results are generated after 1000 epoch. We also applied data augmentation methods like random crop, random flip to prevent overfitting.

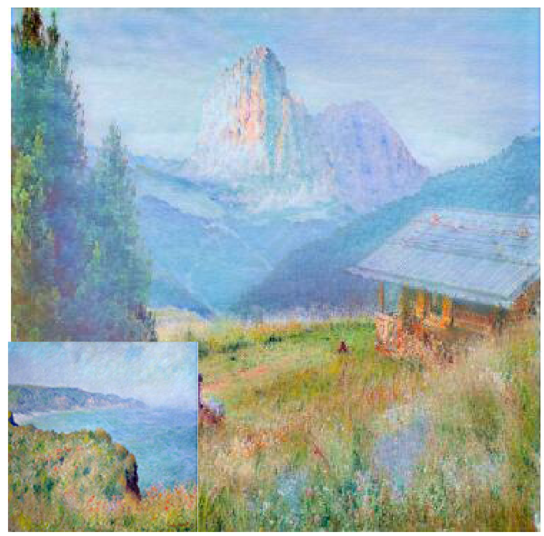
4 Results

4.1 Results of Neural Style Transfer

With the same input original image and different input style image, we can get the final results as following pictures:

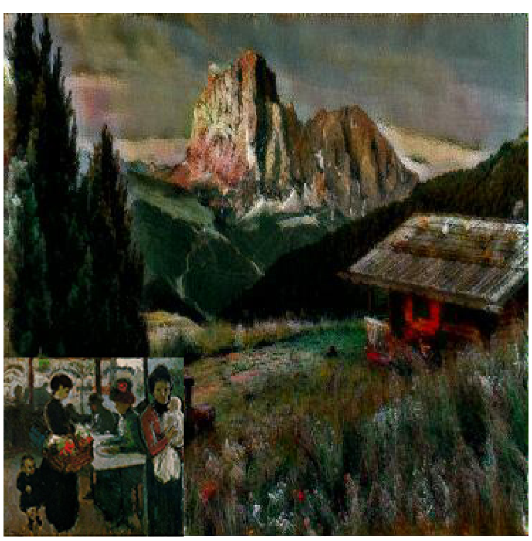
(a) oiginal Image

(b) Expressionism – Martiros Saryan



(c)Impressionism – Van Gogh

(d) Impressionism – Claude Monet



(e) Impressionism – Pablo Picasso

(f) Romanticism – Cezanne

Figure 1: Results using Neural Style Transfer

4.2 Results of Cycle-GAN Style Transfer

Figure 2: Results using Cycle-GAN Style Transfer (1)

As we can see from the figure 2 above, the very left column are original photographs of landscape. The right 4 columns presented generated different painting styles of Van Gogh, Monet, cubism and pointillism respectively.

Clear style are shown in these generated paintings, but there is one problem -- undesired artifact texture showed up in the sky, which is rare in real painting.

Some of the generated Monet paintings are almost identical to the photograph with a little blurry. Probably because Monet's paintings are close to photograph and the style are hard to be captured.

When transferring to paintings, generator performed well when dealing with compact subject like wheat field, flowers and bush. But when the source photographs contain large plain field or sky, generated painting can't express these subjects in the way Van Gogh and Monet does, such as the following paintings in figure 3.

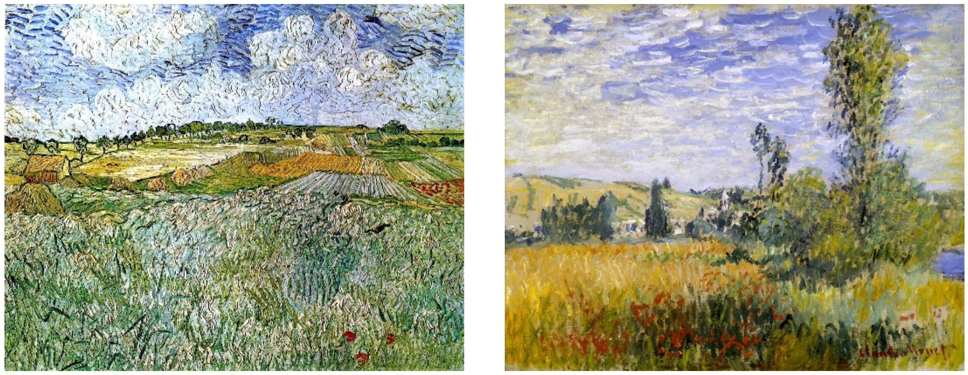


Figure 3: Results of large field or sky using Cycle-GAN Style Transfer

Figure 4: Results using Cycle-GAN Style Transfer (2)

Cubism is a distinct style of using large color blocks with sharp contrast like in figure 4. The results have successfully mimicked the cubism way to express landscape. For pointillism, a painting style that uses color dots a lot, we have produced great samples that is very similar to the real paintings.

The main challenge Cycle-GAN has met is translating photographs with large single colored blocks, such as sky, plain field and ocean to corresponding painting styles. The generated results are always accompanied by unwanted artifact texture and blurry. Also for cubism which has lots of colors to fulfill, the result can’t be very well like in figure 4.

5 Conclusion and Discussion

To finish image style transferring, we used two techniques -- basic original neural style transfer and Cycle-GAN style transfer. And we do get some good results in style transferring like show in all the figures above which is nearly as good as in their original papers.

From this project, firstly, we review and cement some methods we learnt and used before like VGG. Secondly, we learn and develop some new knowledges. For example, when using Cycle-GAN method, we not only cement some basic knowledges of GAN, but also learn how to use this new method on both side to avoid getting bad fake generating images like in normal GAN. What's more, we also learn how to run a big project more efficiently. For instance, we try use GPU and try to build good small testing set and so on.

The difficulties during the project is how to debug and how to develop the results although we already have and build the right model. For example, we get some pictures of wrong size and with black area at the beginning using Cycle-Gan, so we went back and found that we didn't preprocess the images well. And we also got bad results and tried to develop them. For example, we got pictures in different styles when only choosing one style so we went back to adjust the dataset because we found that the style dataset has some colorful landscape paintings but also some portraits in black and white on the contrary.

Also, there are others needed to be discussed. In Neural Style Transfer, when we choose higher layer as content layer, the image structures are matched on an increasingly large scale, leading to a smoother and more continuous visual experience. The change of α to β ratio effects how much the content of the output image is influenced by the style image. If the ratio is too small, then more likely the content of output image will be affected by the style image instead of maintaining the original content. In Cycle-GAN Transfer, since the both target and the source datasets are noisy, in other words, they contain a number of confusing or even false bad samples. Moreover, the style of the same artist or same genre can change when painting different subjects. So, we tried to train Cycle-GAN on a better dataset refined by ourselves and results are better. But with less data, the model also loses the ability to generalize to different subjects. So we need to balance it and avoid overfitting.

We need to try more in our two models like in parameters and datasets. Also, furthermore, we should try some other techniques in style transfer like "Real-Time Style Transfer".

References

[1] Leon A. Gatys, Alexander S. Ecker and Matthias Bethge. (2015) A Neural Algorithm of Artistic Style, *Nature Communications*, URL: https://arxiv.org/pdf/1508.06576.pdf.

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola and Alexei A. Efros. (2017) Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, *International Conference on Computer Vision (ICCV)*, URL: https://arxiv.org/pdf/1703.10593.pdf.