Neural Style Transfer From Scratch

Abstract

In these days, improvment of AI has changed our life. A new image can be produced with CNNs. This process is named Neural Style Transfer. In this writing, Gatsy el. The method will be explained and will be given some information about implemantation.

1.INTRODUCTION

The Convolutional Neural Networks(CNN) architectures have completely changed image processing. In fact, this change was reflected in the painting. It is possible to create a new image by combining a content image and a stylized image thanks to Convolutional Neural Networks(CNN). A content image can be anything, maybe your picture, maybe a landscape icture etc. A stylized image generally is the painting of famous artist, this picture can be van Gogh's "The Starry Night". After all, there is a final image which is a content image merged with style image.

2.MATERIAL and METHODS

2.1 Images

As said above, content image can be anything and in implementation, my image is used for content image. The stylized image chosen as Impression, Sunrais(1872, Cladue Monet). Therefore, it can be said that final result also named as combine image will be Monet style.

Fig.1. Content Image

Fig.2. Style Image

2.2 Architecture

Convolutional Neural Networks(CNN) are used for Neural Style Transfer (NST), because every object must be detected to protect objects' shapes. When Convolutional Neural Networks are trained on object recognition, they develop a representation of the image that makes object information increasingly explicit along the processing hierarchy. Therefore, along the processing hierarchy of the network, the input image is transformed into representations that increasingly care about the actual content of the image compared to its detailed pixel value[1]. Many pre-trained model is available for object recognition, one of these is VGG19. For Neural Style Transfer, VGG19 pre-trained model's weights and biases are used. Each layer of units can be understood as a collection of image filters, each of which extracts a certain feature from the input image. Thus, the output of a given layer consists of so-called feature maps: differently filtered versions of the input image[2]. In the original paper, all convolution layers aren't used. 0. , 5. , 10. ,19. ,29. layers are used for reconstruction. These layers information is really good enough, and reconstruction is almost perfect.Therefore, just these layers are used in implementation. Also, fully- connected layers aren't used because extracting feature maps is the main goal. In the below, VGG19 newtork architecture can bee seen.

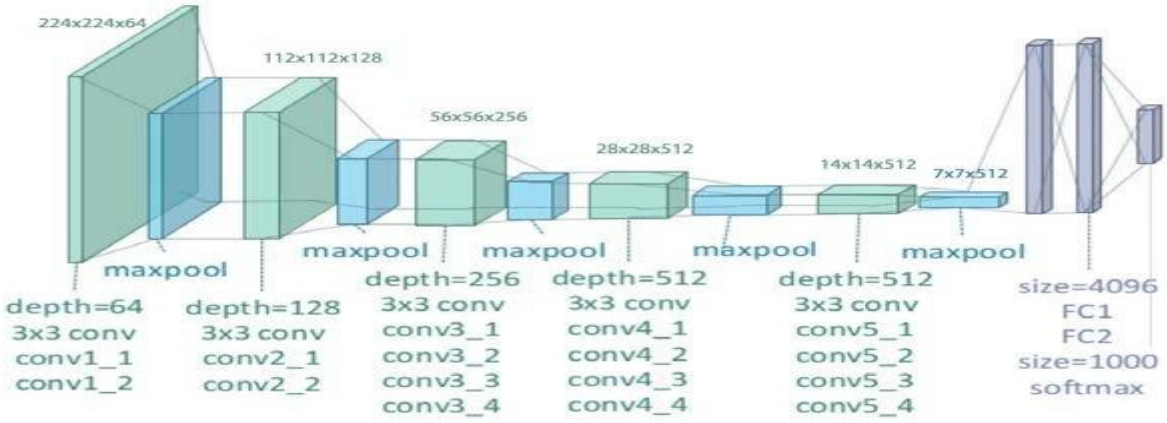

Fig.3. VGG19 Network Architecture

2.3 Hyperparameters

The evaluations of NST algorithms remain an open and important problem in this field. In general, there are two major types of evaluation methodologies that can be employed in the field of NST, i.e., qualitative evaluation and quantitative evaluation. Qualitative evaluation relies on the aesthetic judgments of observers. The evaluation results are related to lots of factors (e.g., age and occupation of participants). While quantitative evaluation focuses on the precise evaluation metrics, which include time complexity, loss variation, etc[3]. There is a question which evaluation methods should be used. There is no exact answer fort this question. However, hyperparameters should be tuned according to loss and evalution metrics.In implementation, qualitative methods are used as evaluation metrics. That means, result image is good in my opinion.

Four major hyperparameter must be tuned in NST. These are learning rate, alpha, beta, iteration number. Alpha is a coefficient that determines how similar the result image will be in the content image. Beta is a coefficient that determines how similar the result image will be in the style image. In below, it is the loss formula(Fig.4.).

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

Fig.4. Total loss formula

In implementation, alpha is 0.9 , beta is 0.1 , iteration number is 10000 and learning rate is 0.001. Also, Adam is used as optimizer.

2.4 Traning

In training, there are content loss and style loss and their sum is equal to the total sum. Content loss is calculated by subtracting the content feature from combine features then it is squared. There is gram matrix. It aimed the maximum similarity. 5 layers that said in the above, style loss calculates for each one individually. Minumum style loss is aimed and this is important for similarity to style image. Style loss is calculated by subtracting the gram combine features from gram style features then it is squared. The total loss is calculated according to a formula that said above. Finally, the total loss updated every iteration. In implementation, model trained 10000 iterations . Every 500 iterations , the total loss was printed and the result is below(Fig.5.)

```
tensor(2899147.7500, device='cuda:0', grad_fn=<AddBackward0>)
tensor(30435.2148, device='cuda:0', grad_fn=<AddBackward0>)
tensor(13538.7041, device='cuda:0', grad_fn=<AddBackward0>)
tensor(8538.5586, device='cuda:0', grad_fn=<AddBackward0>)
tensor(5968.9717, device='cuda:0', grad_fn=<AddBackward0>)
tensor(4294.2949, device='cuda:0', grad_fn=<AddBackward0>)
tensor(3081.3838, device='cuda:0', grad_fn=<AddBackward0>)
tensor(2190.8784, device='cuda:0', grad_fn=<AddBackward0>)
tensor(1557.1614, device='cuda:0', grad_fn=<AddBackward0>)
tensor(1127.3026, device='cuda:0', grad_fn=<AddBackward0>)
tensor(840.2488, device='cuda:0', grad_fn=<AddBackward0>)
tensor(645.8766, device='cuda:0', grad_fn=<AddBackward0>)
tensor(508.4633, device='cuda:0', grad_fn=<AddBackward0>)
tensor(408.5963, device='cuda:0', grad_fn=<AddBackward0>)
tensor(336.1024, device='cuda:0', grad_fn=<AddBackward0>)
tensor(282.0875, device='cuda:0', grad_fn=<AddBackward0>)
tensor(245.6360, device='cuda:0', grad_fn=<AddBackward0>)
tensor(219.8161, device='cuda:0', grad_fn=<AddBackward0>)
tensor(198.7534, device='cuda:0', grad_fn=<AddBackward0>)
tensor(184.4397, device='cuda:0', grad_fn=<AddBackward0>)
```

Fig.5. Total loss every 500 iteration.

3. CONCLUSION

Convolutional Neural Networks are really good at extracting features. Therefore, any manipulation can be easily done on the image. It was also seen that it is possible to make aesthetic changes in the pictures. These aesthetic changes have subjective evaluation criteria that vary from person to person. On the other hand, there is a question whether these are works of art. However, in terms of image processing, it can be said that the result is successful. Also, there are many NST techniques can be used for this task. In implementation, Gatys technique is used. It can be said that this technique is the most popular and successful one. Hyperparameters should be tuned according to the output you want to obtain. Final combine image is the below.

Fig.6. Output Image

4. REFERANCES

1. Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576* (2015).
2. Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576* (2015).
3. Jing, Yongcheng, et al. "Neural style transfer: A review." *IEEE transactions on visualization and computer graphics* 26.11 (2019): 3365-3385.
4. Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576* (2015).