Image style transfer based on Convolutional Neural Network

Shang Da

Iowa State university
Computer Science department

Abstract

Image style transfer is a task such that, given a style reference image and a content image, generate a new image that "applies" the style to original content image. In this project I implemented an image style transferrer mostly based on the work of Gatys(Gatys et al. 2015). I used a pretrained VGG-19 network to extract features from both style and content image. Then I defined style loss and content loss to measure the quality of generated image. I also defined a total-variation loss as a regularization. Then new image is generated by minimizing the weighted sum of above losses using a L-BFGS optimizer.

Introduction

Before the year 2015 image style transfer is considered more as texture transfer. Researchers developed effective algorithms to synthesize a new image by different approaches. For example, Efros and Freeman's algorithm generate a new image by stitching together small patches of existing images(Efros et al. 2001). Hertzmann's approach achieved texture synthesizing by Image Analogies (Hertzmann et al. 2001). However, these methods can only deal with shallow features.

In the year 2015, Gatys introduced a novel approach of utilizing the feature maps output by different layers of a convolutional neural network as a measurement of differences in style and content of two images and treat synthesizing as an optimization problem(Gatys et al. 2015). This new method can to some extent synthesize higher level features and produce better result. Since then different ways of style transfer has been developed based on Gatys' approach. For example, Johnson, J., Alahi, A., & Fei-Fei, L trained a generative neural network for each style and generate stylized image by a feed-forward pass to the network(Johnson, J., Alahi, A., & Fei-Fei, L, 2016). And Tian Qi Chen came up with a new objective function which is called "style swap" and feed the result of style swap to an inverse network to generate new images. This method is efficient but also allows arbitrary content and style images(Tian Qi Chen et al. 2016).

In this project I implemented an image style transferrer mostly based on the work of Gatys. In this term paper, I'll discuss the theoretical foundation of image style transfer, together with important components for its implementation.

Pretrained VGG-19 Network

VGG-19 a very deep neural network structure originally designed for image classification. It consists of sixteen convolutional layers and 3 fully connected layers. The convolutional layers are used for feature extraction and fully connected layers are used for classification. VGG-19 has better performance than earlier models such as AlexNet or Lenet because it replaces large 5×5 or 7×7 convolutional kernels by several consecutive 3×3 convolutional kernels. In this way number of parameters is reduced significantly while receptive field preserves (Simonyan, K., & Zisserman, A 2014). It achieved very high score in ILSVRC 2014.

In this project I used a pretrained VGG-19 network. The weights file is downloaded from this website. I kept convolutional layers for feature extraction purpose and fully-connected layers are discarded. In a convolutional neural network. Different layers will output different feature maps. Feature maps output by lower layers represent low-level features such as texture or stripe, higher-layer feature maps represent high-level features such as semantic and structure. These feature maps are essential for computing the loss function.

Style Loss

Style of an image consists of both texture and semantic features. To obtain these features we consider activation layers: relu1_1, relu2_1, relu3_1, relu4_1, relu5_1. Each style layer l outputs feature map ϕ_l. For each ϕ_l we define Gram matrix G_l:

$$\mathrm{G}_l = \frac{\psi\psi^T}{C_l H_l W_l}$$

Where C_l, H_l, W_l represent channel, height and width of feature map ϕ_l. And ψ is the resized 2D-matrix with size $C_l \times H_l \cdot W_l$ from original 3D feature map with size $H_l \times W_l \times C_l$. The division by $C_l H_l W_l$ is for balancing large and small feature maps. Gram matrix can measure correlation of different features. It captures which features tend to appear together.

We also define style layer weight w_l^s for each style layer l so that we can tune the importance of different layer to get the best stylized image we want.

For two images c, c' the final Style loss function is defined by weighted sum of Frobenius norm over Gram matrix:

$$\text{loss}_{\text{style}} = \sum_{l} \frac{w_l^s \left\| G_l^c - G_l^{c\prime} \right\|_F^2}{C_l H_l W_l}$$

Content Loss

Content of an image should discard textures but maintain semantic and structural information. Therefore, we look at activation layers: relu3_2, relu4_2. For two images C, C' denote feature map of each layer j as ϕ_j, content loss is then defined by weighted sum of Frobenius norm over feature maps:

$$\text{loss}_{\text{content}} = \sum_{j} \frac{\left\| \phi_j^c - \phi_j^{c\prime} \right\|_F^2}{C_l H_l W_l}$$

This measures the distance between feature maps at same layer of two images.

Total Variation Regularization

Total variation regularization is defined as:

$$\text{loss}_{\text{tv}} = \sum_{c} \sum_{h} \sum_{w} \left(\left(x_{h,w+1,c} - x_{h,w,c} \right)^2 + \left(x_{h+1,w,c} - x_{h,w,c} \right)^2 \right)$$

Which measures difference of every pixel with pixels around it. It can significantly reduce components that look "unnatural".

Total Loss

We have defined Style Loss, Content Loss and Total Variation Regularization. The total loss is represented by a weighted sum of above loss function:

$$\text{loss}_{\text{total}} = \lambda_{\text{style}} \text{loss}_{\text{style}} + \lambda_{\text{content}} \text{loss}_{\text{content}} + \lambda_{\text{tv}} \text{loss}_{\text{tv}}$$

Style Transfer as Optimization Problem

With well defined total loss function. We can then treat the image style transfer problem as an optimization problem that minimize $\text{loss}_{\text{total}}$ with respect to the generated image. The generated image can be initialized as a blank image with some gaussian noise. I used a L-BFGS optimizer. BFGS is a second order optimization algorithm. It guarantees to converge faster in the sense of number of iterations. It also can jump out of local minima and saddle points compared with first-order optimization algorithms such as stochastic gradient descent.

Style Transfer by Feed-forward style network

Using completely the same loss function. We can also build a generative neural network and train a "general" image style transferrer for a fixed style image and all content images(Johnson, J., Alahi, A., & Fei-Fei, L 2016). The training takes a few days but generating arbitrary stylized image only takes a few second since it requires only a forward pass to generative neural network. This technique can be further extended to real-time video style transfer.

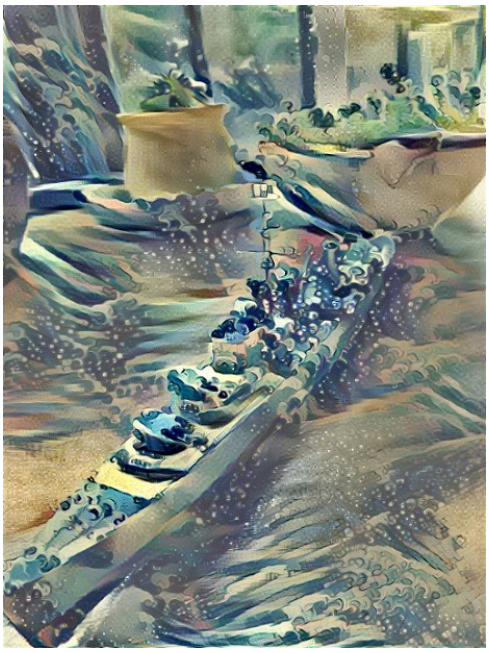
Experiments

I used different style images and content images to test the result.

1. Style: The Great Wave off Kanagawa, Hokusai, 1829-1832.

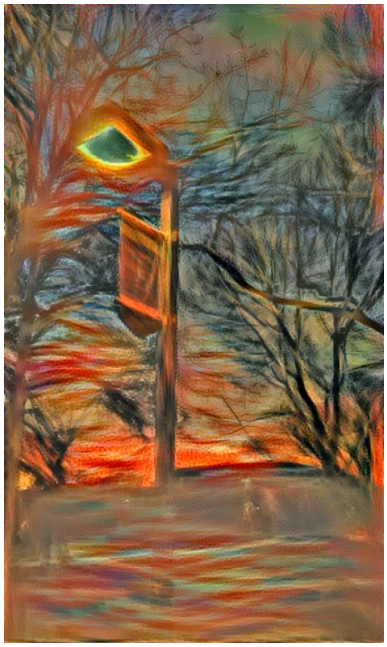
Content:

Stylized:



2. Style: The Scream Content:

Stylized:



3. Style:

Content:

Stylized:

Conclusion and Future work

In this project I implemented an image style transferrer mostly based on the work of Gatys. The result is satisfactory but to generate each stylized image takes approximately 15min, which is a lot of time. If I want to further extend this work, I'll focus on generative network approach.

References

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.

Efros, A. A., & Freeman, W. T. (2001, August). Image quilting for texture synthesis and transfer. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (pp. 341-346). ACM.

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., & Salesin, D. H. (2001, August). Image analogies. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (pp. 327-340). ACM.

Johnson, J., Alahi, A., & Fei-Fei, L. (2016, October). Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision (pp. 694-711). Springer, Cham.

Chen, T. Q., & Schmidt, M. (2016). Fast patch-based style transfer of arbitrary style. arXiv preprint arXiv:1612.04337.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.