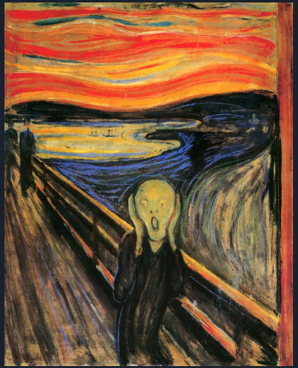
Style Transfer Using VGG

Group 16

Gaotong Wu Xin Du Zhuomin Zhang

Neural Style Transfer --Background

- Artistic Style:
  - Post-Impressionism: Van Gogh's The Starry Night
  - Traditional Japanese: The Great Wave,
  - Expressionism: The Scream
- Apply these artistic styles for our own images, so these artistic styles can not only be represented in the famous painting but also many different images;
- Anime production company use this technique to generate a specific anime style;



Neural Style Transfer -- Literature Survey

- Efros and Freeman: use an image quilting algorithm to do texture transfer in 2001.
  - Given an input image, produce an output image that are both visually similar to and pixel-wise different from the input, and having possibly a larger size;
  - Analysis: estimating a set of relevant statistics from the input texture image;
  - Synthesis: computing an image that satisfies the statistical constraints estimated during the analysis step.

- Sebastian Penhou¨et and Paul Sanzenbacher: Deep Photo Style Transfer
  - runs several pipeline steps for computing the final transfer image;
  - creating a segmentation mask, grouping segmentation classes, defining and precomputing loss functions and gradually optimizing the transfer image;

- <u>The most famous work is done by Leon A. Gatys, Alexander S.Ecker and Matthias Bethge in 2016:</u> use deep learning network to extract features and content from the input image, train a capable network to generate a specific style.

Neural Style Transfer -- Dataset

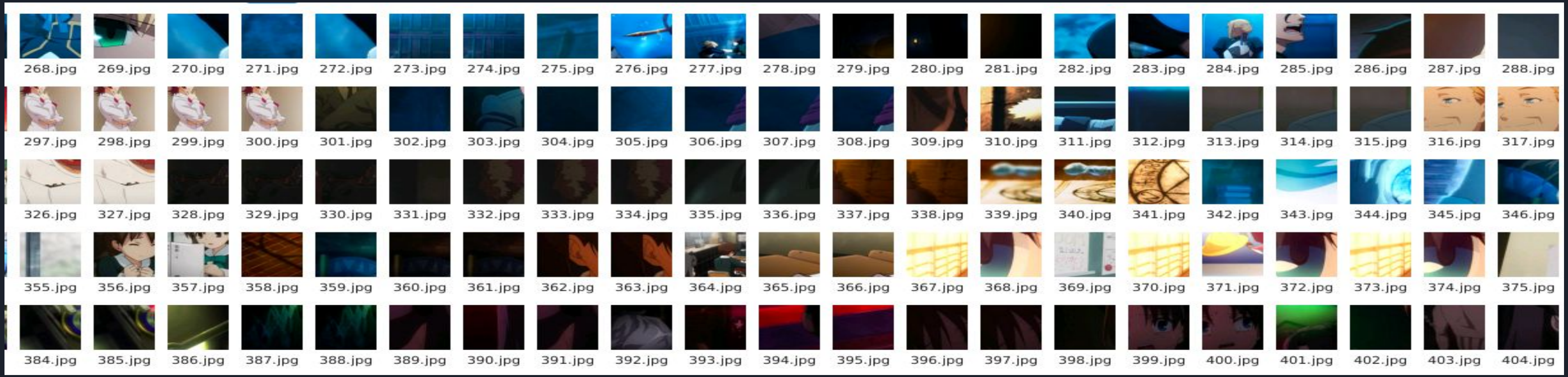
- The dataset we use: ImageNet1000(mini)

  https://www.kaggle.com/ifigotin/imagenetmini-1000

  It is a subset of ImageNet. It has 1000 categories of image, but in each category, it only contains dozens of pictures which is less than ImageNet.

Neural Style Transfer -- Dataset

- Self-collected dataset:
  - Collect from the anime called "*fate zero*";
  - Backup dataset prepared for use in further work;
  - Can be used to train anime style transfer network;
  - Can be used to compare the performance differences between self-collected dataset and professional online dataset;



Neural Style Transfer -- Feature extraction

- The critical problem: How to represent the content and style of an image separately?

Content representation:

Consider a CNN trained on object recognition. If the network is robust enough, then the output should be sensitive to the content of the image, but invariant to the precise appearance.

Therefore, we can regard the features generated by the convolution layers near the output layer as our content representation.

Style representation:

We use the Gram matrix on a feature space to be our style representation.

$$G_{ij} = \sum_k F_{ik} F_{jk}$$

The intuition of this matrix is that, it calculates the difference between two features. If two features are same, then the entry in G will be large, and if two features are different, then the entry in G will be small.

Neural Style Transfer -- Optimization Goal Intuition

- We want content features of the result image to be as same as the natural image as possible.
- We want style features of the result image to be as same as the style image as possible.

Based on the intuition, we can design our loss function:

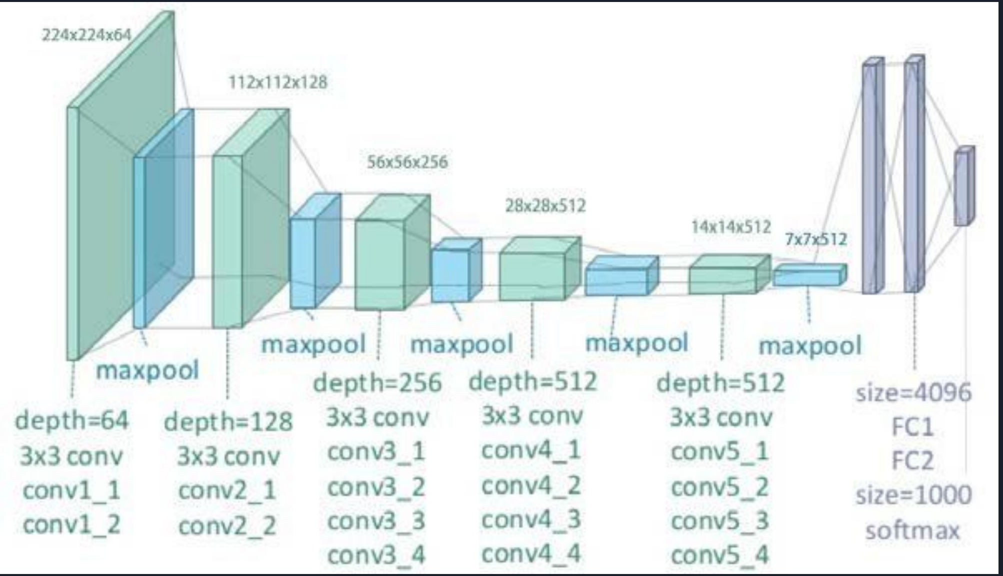
$$\mathcal{L}_{content} = \frac{1}{2}\sum_{i,j}(F_{ij}^{l} - P_{ij}^{l})^2$$

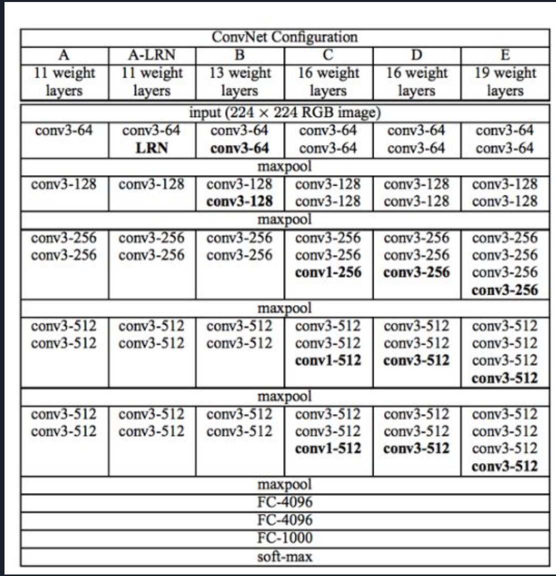
$$\mathcal{L}_{style} = \frac{1}{2}\sum_{l=0}^{L}(G_{ij}^{l} - A_{ij}^{l})^2$$

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{content} + \beta\mathcal{L}_{style}$$

Neural Style Transfer -- Model Introduction

VGG19 architecture





ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Notice: ‘19’ doesn’t mean that it has 19 layers, it means that it has 19 weight layers.

Neural Style Transfer -- Details in model

Adjustments on VGG19:

Use pre-trained VGG19 without the dense layers and input shape.

```
vgg = tf.keras.applications.VGG19(include_top=False, weights='imagenet')
```

Input image batch: Natural image, style image, natural image with white noise(combination image).

Content: We use the output of block5_conv2 as our content representation

Style: We use the gram matrix of outputs of 'block1_conv1','block2_conv1','block3_conv1', 'block4_conv1','block5_conv1' as our style representation.

Goal: On each epoch, adjust the combination image to make the loss function smaller.

Output image: the combination image which has the smallest loss value.

Result

content

style

Result

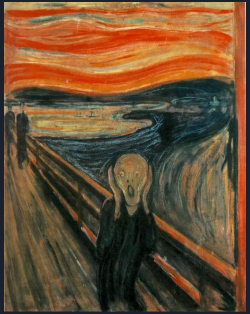
content

style

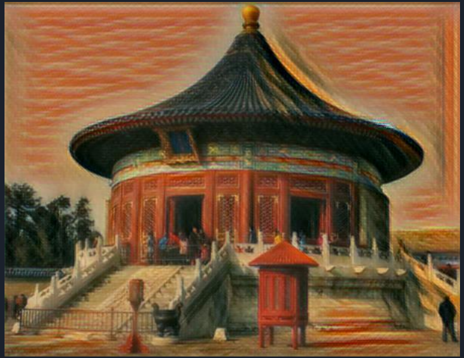
output

Result

The Muse,
Pablo Picasso, 1935

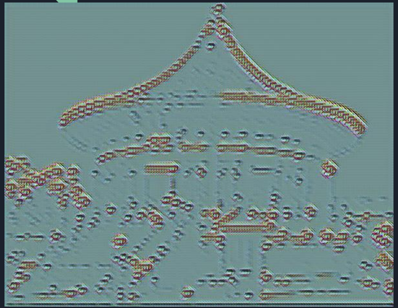


The Scream,
Edvard Munch, 1893

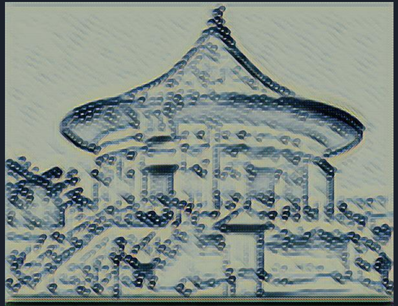


Training process

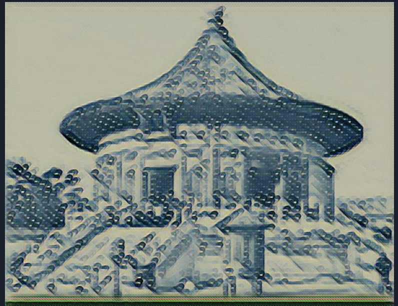
batch-size=4



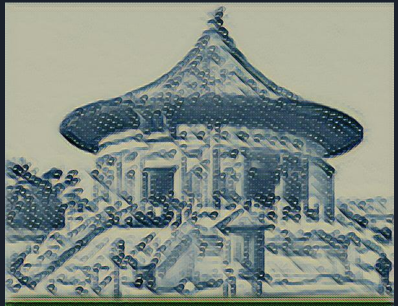
200 iterations



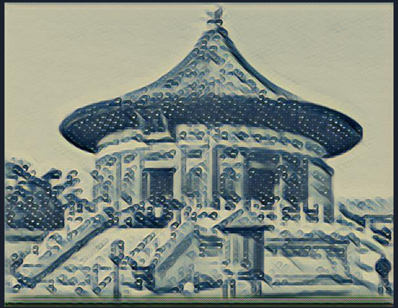
400 iterations



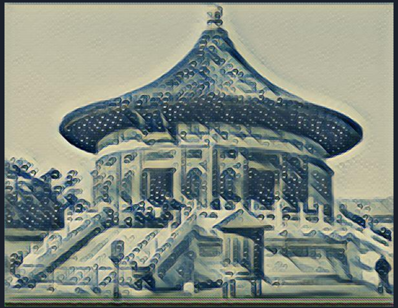
600 iterations



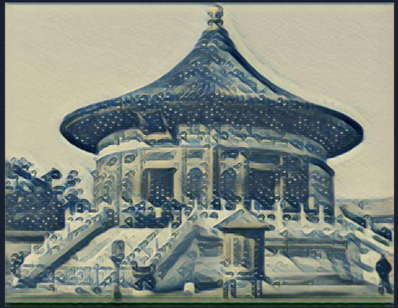
800 iterations



1400 iterations



2000 iterations



3200 iterations

Fully optimized

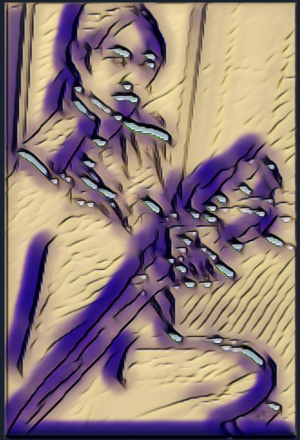
Further Work

1. Optimization on style representation

   Try out different layers as style layers, test their outputs.

   For different style, adjust the weight of style loss and content loss used in total loss function.

2. Try out and improve other styles (eg. Anime) rather than world famous paintings with distinct artistic style.



Reference

Gatys, L.A., Ecker, A.S. and Bethge, M., 2016. Image style transfer using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2414-2423).

Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky. "Instance normalization: The missing ingredient for fast stylization." arXiv preprint arXiv:1607.08022 (2016).

Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." European conference on computer vision. Springer, Cham, 2016.