Experience on Painterly Harmonization

Tianyu Jiang, jianti@iu.edu

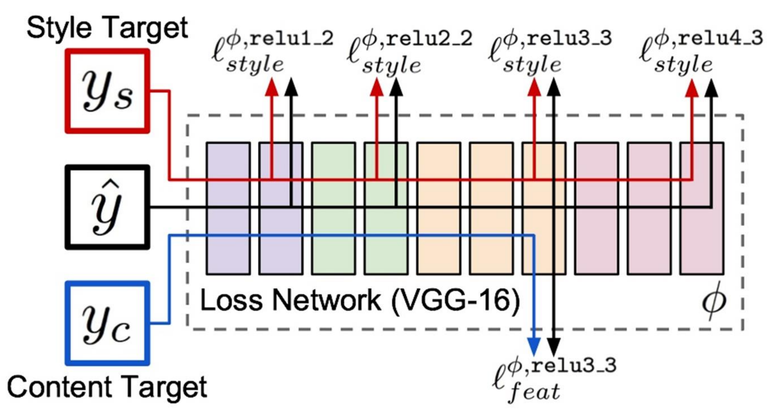
Introduction

When we add a new pattern to an image, we often want the pattern to fit perfectly with the original image. It could be done easily using Photoshop, however, I'll try to get a better result by building a model with VGG neural network [1], which can transfer image style and splice these two images. In this project, I mainly re-implement the paper "Deep Painterly Harmonization“ [2].

Methods

Neural Style Transfer

Transfer one image into another style using neural networks.



Based on the famous neural style transfer model using VGGNet [3] [4], this project will more focus on local style transfer instead of the entire image to make the new added pattern and the oil painting blend well.

We'll use an pre-trained VGG-16 Network as the feature extractor, which is used widely in style transfer field.

First Pass: Robust Coarse Harmonization

VGG-16 neural network optimized with content loss and style loss.

- Start from the simple cut-and-paste image.
- $\mathcal{L} = w_s\mathcal{L}_s + w_c\mathcal{L}_c$
- Content layer: conv4_2

  Style layer: conv1_1, conv2_1, conv3_1, conv4_1, conv5_1

Second Pass: High-Quality Refinement

VGG-16 neural network optimized with content loss, style loss, total variation loss and color (RGB) standard variance loss.

- Start from the intermediate image (the output of the first pass).
- $\mathcal{L} = w_s\mathcal{L}_s + w_c\mathcal{L}_c + w_{tv}\mathcal{L}_{tv} + w_{rgb}\mathcal{L}_{rgb}$
- Content layer: conv4_2

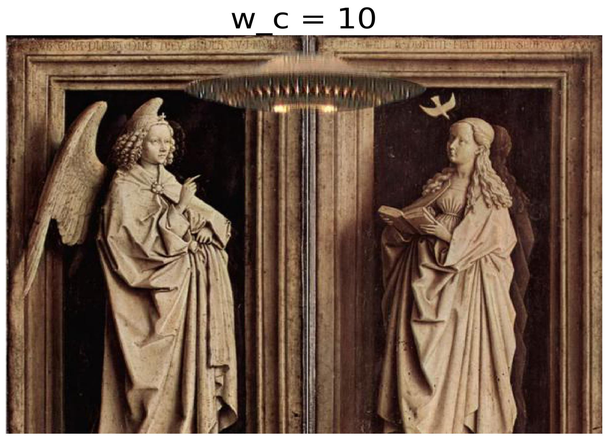
  Style layer: conv1_1, conv2_1, conv3_1, conv4_1, conv5_1

Weight Selection for Pass 1

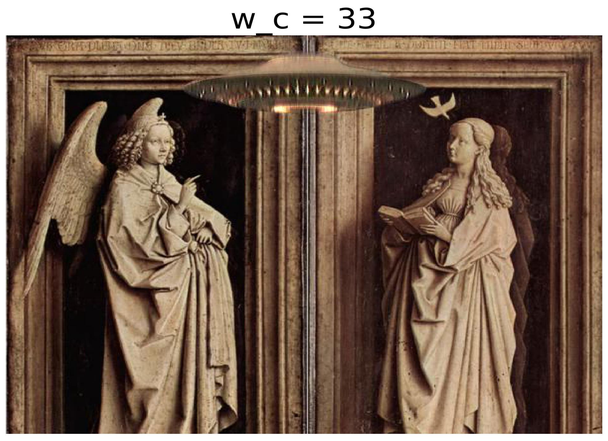
Firstly, I try some different content and style weights for the following example oil painting. The content weight varies from 1 to 1000, and the style weights remain at fixed values. We only observe the output of pass 1 for choosing a better intermediate image for pass 2.

0 Cut and Paste

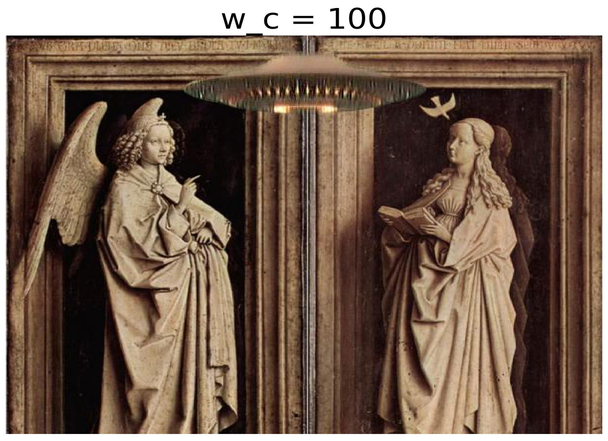
1 w_c = 10



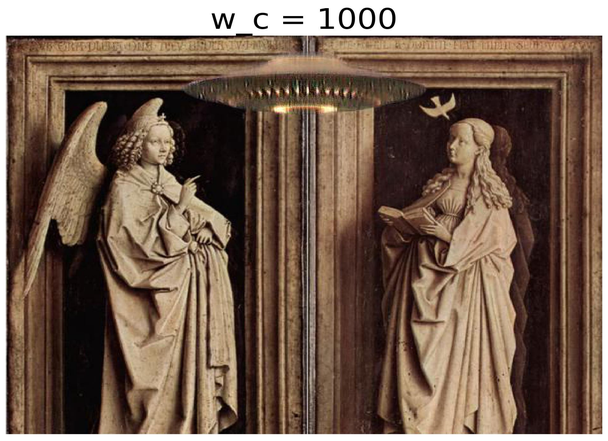
2 w_c = 33



3 w_c = 100

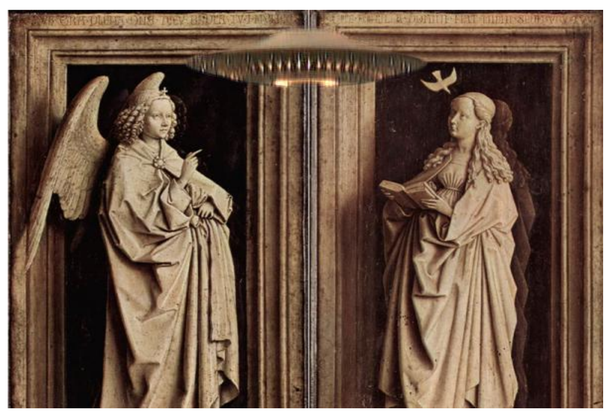


4 w_c = 1000



Weight Selection for Pass 2

- Lower content weight, 33 -> 3
- Higher style weight, [30,10,1,0.3,0.1] -> [30,10,5,4,2]
- Add total variation loss and color loss



Pass2 Output: w_tv=0.01, w_rgb=100

Conclusion

In general, we transfer the added pattern's style successfully, although there're still many problems with the details.

- Too much noise
- Losing content details

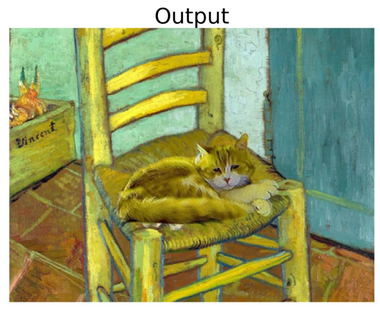
In [2], they applied two other methods to improve the image quality. Due to time limitations, I haven't implemented these two optimizations yet.

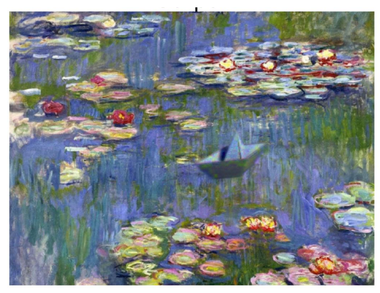
- Chrominance Denoising
- Patch Synthesis

Other Results

Cut and Paste

Output





References

1. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

2. Luan F, Paris S, Shechtman E, et al. Deep painterly harmonization[C]//Computer graphics forum. 2018, 37(4): 95-106.

3. Gatys L A, Ecker A S, Bethge M. Texture synthesis using convolutional neural networks[J]. arXiv preprint arXiv:1505.07376, 2015.

4.Gatys L A, Ecker A S, Bethge M. Image style transfer using convolutional neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2414-2423.

Data Science Department
School of Informatics, Computing & Engineering
Indiana.edu