Comparative Analysis of Different Methods in Neural Style Transfer

Xiaosong Wang, Mengying Yu, Zheyi Wang, Qi Xie

Introduction

Neural style transfer (NST) is a neat idea to demonstrate that artificial intelligence can play a part in producing images with artistic attributes. NST builds on the key idea which extract style information form artistic images, and recombine them with the content extracted from other natural images to form a new images that both resembles the content image as well as seemingly been "artistically produced" by a professional artist.

Problem Definition

In this project, we are looking into two different methods of NST, one is the per-style-per-model, and the other is the arbitrary-style-per-model. We aim to reproduce the image-styling process of these two models. And the result will be evaluated both quantitatively and qualitatively.

For the quantitative evaluation, we are looking at the training time as well as the "speed" of producing output images. And for the qualitative evaluation, we will manually compare the end quality of the result images to find out which method produces the optimal image reconstruction quality.

Datasets

Content Dateset: AISegment.com - Matting Human Datasets

- High resolution extraction of humans from images. This dataset, developed by AISegment aims to help by providing a solid quality dataset of images and masks.

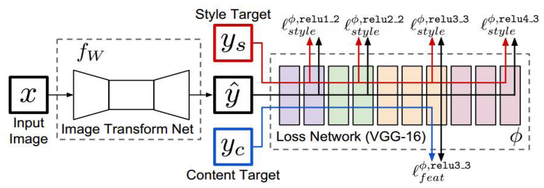
Style Dataset: A Collection of digital illustrations extracted from Pixiv.net

- Fulll image credit goes to Pixiv artists: Sakimori, Hinata, and echowater.

Approaches

1. Per-Style-Per-Model

- Fast Style Transfer Architecture based on Johnson et al (2016). In this approach, We use a pre-trained VGG19 (instead of VGG-16 used in the figure below) as Loss Network to define Perceptual Losses (Feature Reconstruction Loss & Style Reconstruction Loss).



Approaches (cont.)

- **Feature Reconstruction Loss Function:** φj is the output at the jth layer of the network φ for the input x.

$$\ell^{\phi,j}_{feat}(\hat{y}, y) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2$$

- **Style Reconstruction Loss Function:** Gj is the gram matrix of the jth layer.

$$\ell^{\phi,j}_{style}(\hat{y}, y) = \|G^{\phi}_j(\hat{y}) - G^{\phi}_j(y)\|_F^2.$$

2. Arbitrary-Style-Per-Model

- This model is based on the research conducted by T. Q. Chen and M. Schmidt (2016), which defines a new optimization objective for style transfer while only depends on one layer of the CNN.
- **Optimization Function**: C and S represent the RGB representations of the content and style images, respectively.

$$\phi^{ss}_i(C, S) := \underset{\phi_j(S),\, j \in N_s}{\arg\max} \frac{\langle \phi_i(C), \phi_j(S) \rangle}{||\phi_i(C)|| \cdot ||\phi_j(S)||}.$$

- **Stylized Image Computation**: Place a loss function on the activation space with target activations Φss(C, S).

$$I_{stylized}(C, S) = \underset{I \in \mathbb{R}^{3 \times H \times W}}{\arg\min} ||\Phi(I) - \Phi^{ss}(C, S)||^2 + \lambda \ell_{TV}(I)$$

- **The use of an 'Inverse Network'**: T. Q. Chen and M. Schmidt also proposed another way of optimization, which is to train an inverse network to approximate the optimum of the loss function.

$$\min_{\widehat{\Phi^{-1}}} \frac{1}{n} \sum_{j=1}^{n} ||\Phi(\widehat{\Phi^{-1}}(\Phi_j)) - \Phi_j||^2 + \lambda \ell_{TV}\left(\widehat{\Phi^{-1}}(\Phi_j)\right).$$

Result & Analysis

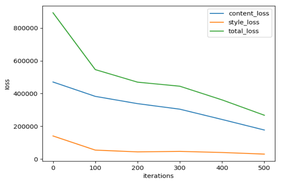
1. Model 1

- In this method, the training set of content is 463 pictures of human faces and resize these pictures to 256*256. We train this model with batch_size=4, giving 5 epochs over the training data. Learning rate is 0.002. Traning time is about 4 mins and 40secs.
- Weights:

	Content_weight	Style_weight	TV_weight
Style 1	1.0	3.0	0.01
Style 2	1.0	15.0	0.01

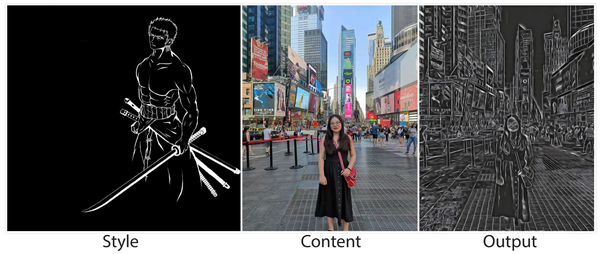
- Because the training set is focusing on faces, so we found the model works well on changing the style of faces. However, it's not so good on other parts of the pictures.

Style Content Output

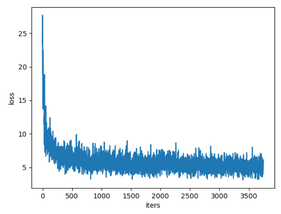


1. Model 2

- The training pictures are resized to 256x256. We train this model with batch = 4, giving 5 epochs over the training data. Learning rate: 0.001 TV_weight = 0.0001.
- It takes about 1 hour to train the model with training set size equals to 10,000, but once the training finished, it only takes a few seconds to do the style transfer with a GTX1070.



Style Content Output



- When the dataset is small, it is better to use the **Per-style-per-model** approach for NST.
- The **Arbitrary-style-per-model** works very well if we want to transfer the picture style to an extremely distinctive style. But the performance is not so good if we need a complex

Reference

- J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in European Conference on Computer Vision, 2016, pp. 694–711
- T. Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," in Proceedings of the NIPS Workshop on Constructive Machine Learning, 2016