Traditional Chinese Ink Painting Neural Style Transfer

Pu Ke
Stanford University – CS231N Course Project

Introduction

Traditional Chinese ink painting is unique in its drawing techniques:

- Leaves a lot of blank space
- Focuses more on the stroke and lines
- Color comes with a gray shade
- …

Dwelling in the Fuchun Mountains, by Huang Gongwang, painted in 14th century.

Problem Statement

Train and evaluate different classical and popular neural style transfer models and our modified versions and compare the results both quantitatively and qualitatively.

- Original Models: Original Neural Style Transfer (Gatys et al.)[1], Fast Neural Style Transfer (Johnson et al.)[2], CycleGAN (Zhu et al.)[3]
- Modified version: Use edge maps (HED) of the content images as input
- Evaluation: Visual Quality Comparison, Human Study, Nearest Neighbor Test, Stylization Speed, Training Time, Training Loss comparison

[1] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. CoRR, abs/1508.06576, 2015.
[2] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. CoRR, abs/1603.08155, 2016
[3] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. CoRR, abs/1703.10593, 2017.

Dataset

1976 landscape themed Chinese ink paintings (style images) [1] 256x256

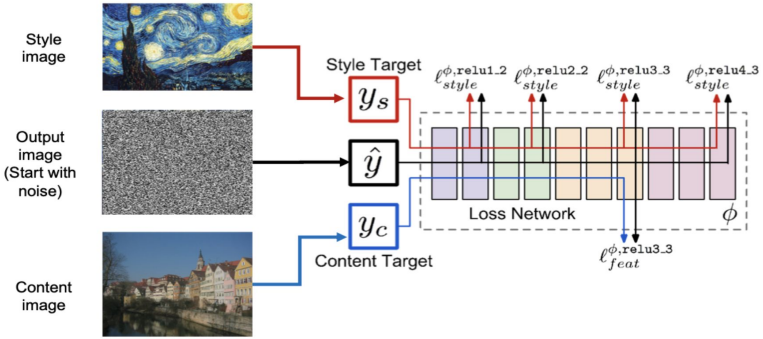
1976 landscape photos (content images) [2] 256x256

The one style image used for Original/Fast NST methods

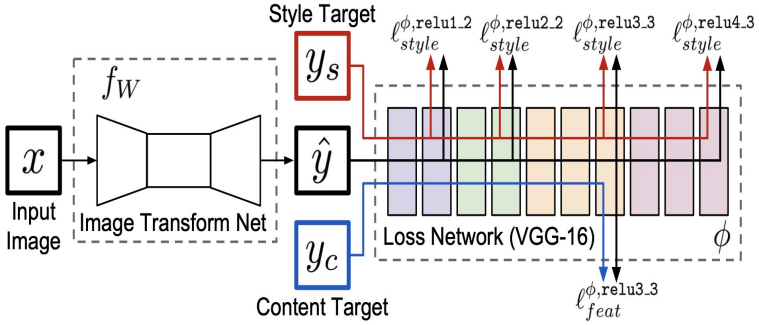
[1] Bin He, Feng Gao, Daiqian Ma, Boxin Shi, and Ling-Yu Duan. Chipgan: A generative adversarial network for chinese ink wash painting style transfer. In Proceedings of the 26th ACM International Conference on Multimedia, MM'18, page 1172–1180, New York, NY, USA, 2018. Association for Computing Machinery.

[2] Alice Xue. End-to-end chinese landscape painting creation using generative adversarial networks. CoRR, abs/2011.05552, 2020.

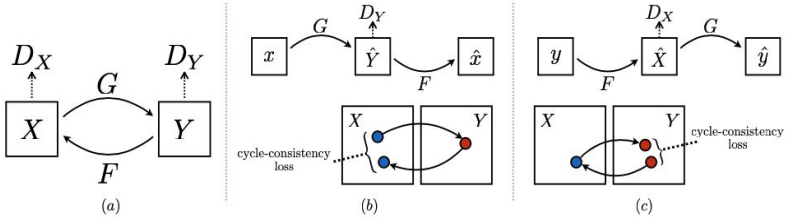
Methods



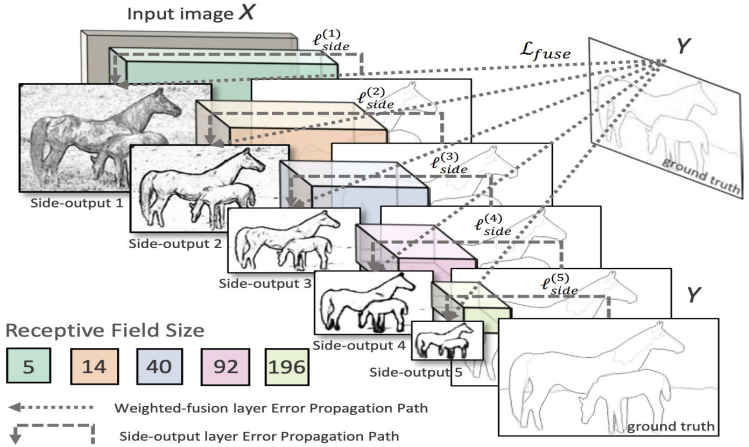
Original Neural Style Transfer



Fast Neural Style Transfer



CycleGAN



HED

Training Details

Original NST (+) HED:

- 1 style image + 1 content images for online inference
- Style weight/Content weight: 5e4
- Each generation takes 500 iterations

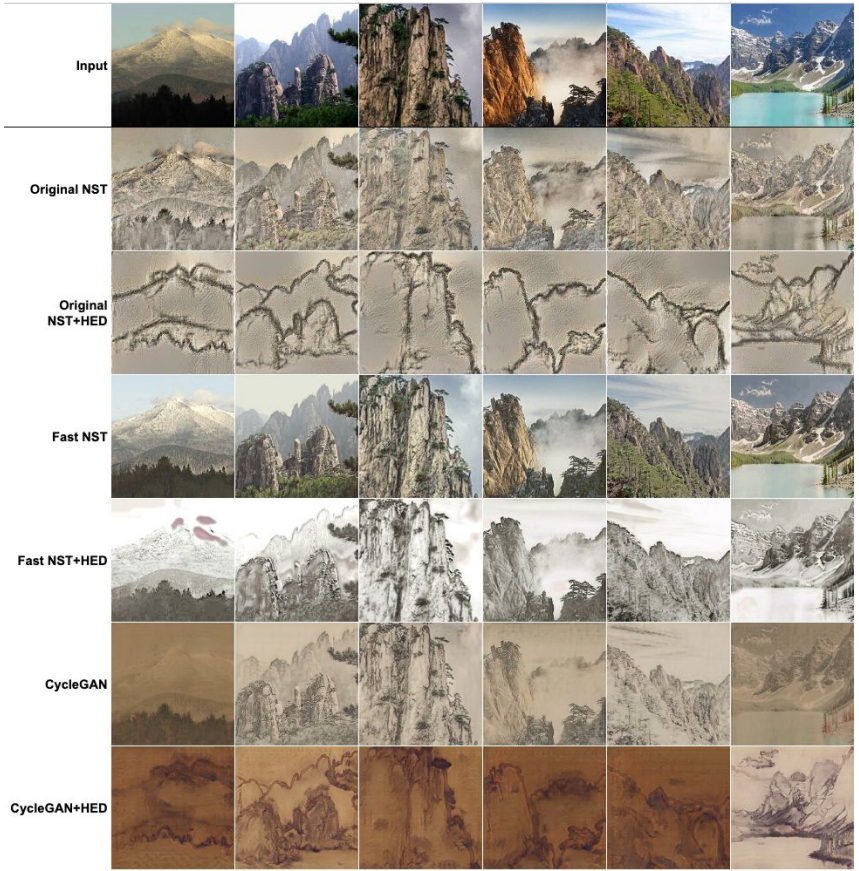
Fast NST (+) HED:

- 1 style image + 1778 content images for training
- Style weight/Content weight: 5e4
- Batch Size: 4
- Epoch: 2

CycleGAN (+) HED:

- Unpaired 1778 style images + 1778 content images
- Batch Size: 1
- Epoch: 85
- Learning rate: 0.0002 without learning rate decay.

Results – Visual Quality Comparison & Human Study



Red trees!

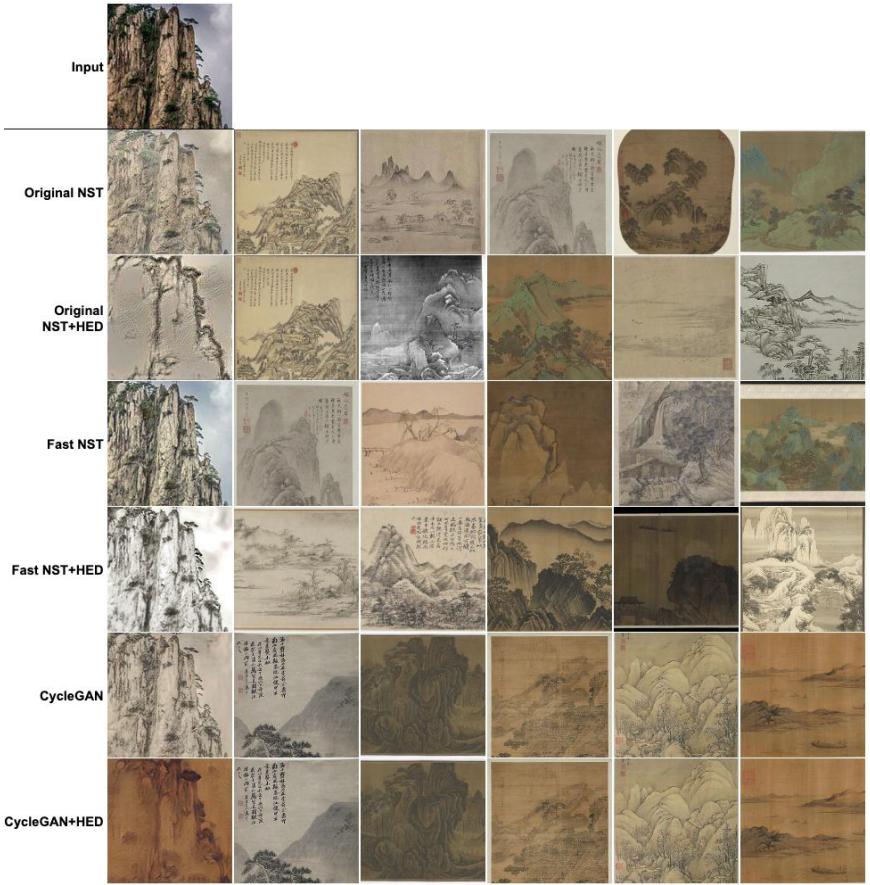
Pinky patches!

Methods	Average Score
Original NST	3.9
Original NST+HED	5.4
Fast NST	5.22
Fast NST+HED	2.7
CycleGAN	1.4
CycleGAN+HED	2.3

Average score among 10 participants. 1 is the best score. 6 is the worst score.

Disclaimer: the model is trained under many constraints and the final result might not be the truth.

Results – Nearest Neighbor Test



The first column shows the generated images through 6 methods with the same input content image.

The other five columns show their corresponding 5 nearest neighbors in the training dataset.

The nearest neighbor is based on the L2 distance of pixels.

Results - Stylization Speed & Training Time

Methods	Time (s)
Original NST	6.164
Original NST+HED	6.336
Fast NST	0.312
Fast NST+HED	0.484
CycleGAN	0.008
CycleGAN+HED	0.180

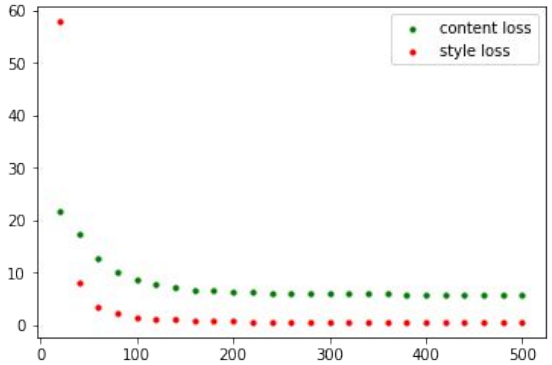
Time used to generate one 256x256 stylized image

Training time:

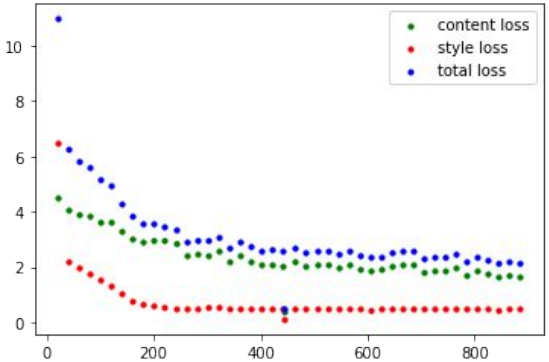
- Fast NST: ~4 hours for 1778 content images + 1 style image for 2 epochs
- CycleGAN: ~23.5 hours for 1778 content images + 1778 style images for 85 epochs

Results - Training Loss Comparison

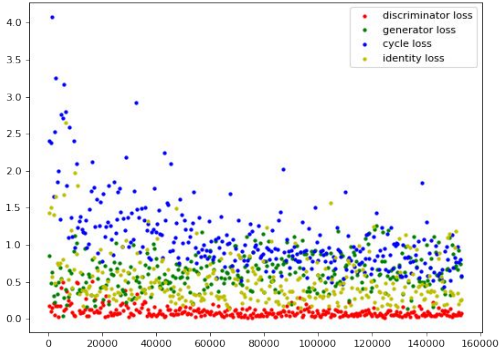
Note: For Original/Fast NST (+) HED, the loss is plotted every 20 iterations. Each batch has 4 samples and the model is trained for 2 epochs. For CycleGAN (+) HED, the loss is plotted every 500 iterations. Each batch has only 1 sample and the model is trained for 85 epochs. So the flavor of the training loss images might look very different.



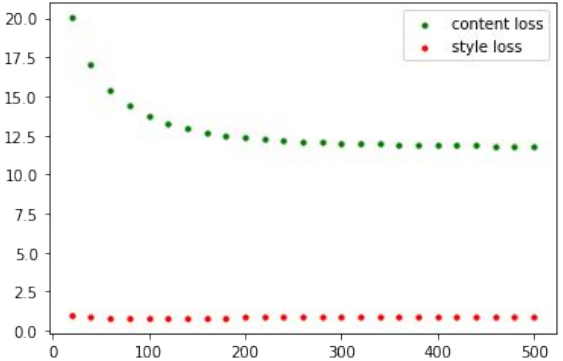
Training loss for Original NST



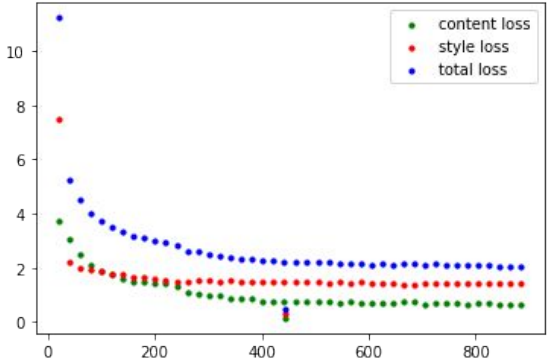
Training loss for Fast NST



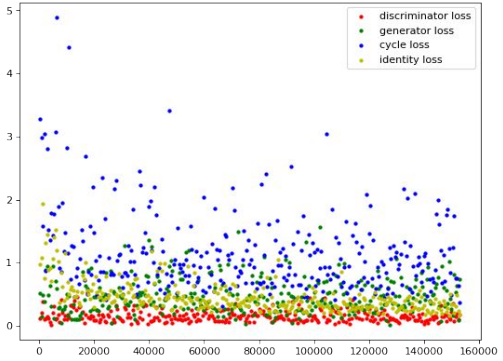
Training loss for CycleGAN



Training loss for Original NST + HED



Training loss for Fast NST + HED



Training loss for CycleGAN + HED

Conclusions & Future Work

Conclusions:

- Best visual effect: CycleGAN
- Worst visual effect: Original NST + HED.
- Fast inference: CycleGAN (+) HED
- Slowerst inference: Original NST (+) HED
- Longest training time: CycleGAN (+) HED
- Shortest training time: Original NST (+) HED

Future Work:

- Fine tune the hyperparameters
- Probe into the reason why adding HED is not working for Original NST and CycleGAN
- Train for longer with learning rate decay for CycleGAN (+) HED
- Analyze the opposite direction generator and discriminator in CycleGAN (+) HED