

I. Introduction: This can be copied from the proposal.

The project implements an existing paper, A Neural Algorithm of Artistic Style, one of the most fundamental papers on neural style transfer. We chose this paper because it uses the basic model of convolutional neural networks, which allows us to gain a deeper understanding of the model by accessing and modifying its architecture easily. Some ablation experiments include:

1. Contrast model performance with various pooling operations (max-pooling, average pooling)
2. Adjusting the number of style layers and observe its effects by visualizing the model's output

II. Challenges:

The largest obstacle we have encountered so far is adapting the model from Pytorch to tensorflow. In terms of basic functionality, our implementation does not fully mimic the source code. If possible, we'd like to integrate our implementation in tensorflow to a more concise and efficient structure comparable to the pytorch implementation. For example, the feature extraction functionality was distributed throughout different functions in the pytorch class, but integrated as an independent function when we implemented it via tensorflow. This could be attributed to the fact that tensorflow is more static and pytorch is more dynamic in nature.

III. Insights: Are there any concrete results you can show at this point?

How is your model performing compared with expectations?

Our model does exhibit the basic functionalities of a NST model. It reasonably integrates the content image and the style image to transform the content image to the intended style while preserving the original form of the content image. However, Compared to the original implementation, our NST deforms the original content image to a greater extent. This may be caused by a larger weight of the style layer, or the lack of epochs in our training. We will look more into this.

In addition, we will visualize our model's behavior by plotting the training loss across epochs. We will also present the testing loss.

IV. Plan:

1. **Are you on track with your project?: We are mostly on track with the project.** By this time, we have finished most of the model's implementations by (1) establishing a pre-processing pipeline for style and content images (2) Implement the gram matrix to calculate style loss (3) extracted VGG19 layers to serve as style and content extraction layers (4) implemented a train function to train the model.
2. **What do you need to dedicate more time to?** By this point, we have completed the base goal of our project. For our next steps, we will start performing ablation experiments by varying the layers used for content and style extraction, or the pooling method. We will also work on the target goal and implement multi-headed neural style transfer.
3. **What are you thinking of changing, if anything?** To enhance readability of the code, we plan to divide the code into different blocks by functionality including the

preprocessing step, the utilities and loss functions, the training function, and the main function.

Content loss: Block5 conv4



Content loss: Block4 conv4



Content loss: Block3 conv4



conv2_2



The content that's being modified becomes more localized, producing small area changes

NST tf mimic torch

Original tf: [Another copy of NST_tf.ipynb](#)

NST cleaned w experiments

NST torch: [NST_torch.ipynb](#)

Poster Todos:

- ☐ Compare with the hub preload model?
- ☐ Style-content Loss curves
 - ☐ For ablation experiment, maybe a content layer depth vs total loss curve?

May 10th TODOS:

- ☐ Clean code such that it is organized like pytorch



AesFA: An Aesthetic Feature-Aware Arbitrary Neural Style Transfer

Joonwoo Kwon^{1*}, Sooyoung Kim^{1*}, Yuewei Lin^{2†}, Shinjae Yoo^{2†}, Jiook Cha^{1†}

¹Seoul National University ²Brookhaven National Laboratory

*{pioneers, rlatndud0513}@snu.ac.kr †Co-corresponding author



Introduction

- Style representations are highly correlated to spatial information.
- Existing methods heavily rely on pre-trained networks.

[Main Contributions]

- We propose a **lightweight** yet **effective** model for aesthetic feature extraction and stylization.
- This framework introduces a new module for stylization and straightforward contrastive learning, while completely excluding pre-trained networks at inference.
- Our approach achieves SOTA results **regardless of resolutions (256 to 4K)** while achieving faster inference.

AesFA MicroAST

EFDM AdaIN

Method

- Are all negative samples necessary?: The k -th nearest negatives first. In this paper, we chose $k=1$.

$L_{Aes} = L_{Aes1} + L_{Aes2}$

$L_{total} = L_C + L_S + L_{Aes}$

Qualitative Results

- AesFA achieves promising outputs with higher image quality.
- Our method faithfully transfers tones, texture, brushstrokes, and grains.
- Suitable for **real-time ultra-high resolution rendering (4K)** applications.

Content Style AesFA MicroAST EFDM Style2 AdaIN AdaIN+ AdaIN+ AesFA+ AesFA+ EFDM+

Experiments & Results

- Dataset: COCO (Content), WikiArt (Style), Nvidia A100
- User Study: 1,580 valid votes from 79 subjects.
- AesFA outperforms all baseline models in various metrics.

Resolution (px)	Model	Style Loss (L1)	LPFS (L2)	MSRM (L2)	Time (sec.)	PSNR (dB)
256	AdaIN (2017)	0.016	0.079	0.246	4.017	23.20
	AdaIN (2017)	0.021	0.071	0.220	0.001	21.40
	MicroAST (2021)	0.006	0.070	0.240	0.001	22.20
	MicroAST (2021)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
512	AdaIN (2017)	0.016	0.079	0.246	4.017	23.20
	AdaIN (2017)	0.021	0.071	0.220	0.001	21.40
	MicroAST (2021)	0.006	0.070	0.240	0.001	22.20
	MicroAST (2021)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
1024	AdaIN (2017)	0.016	0.079	0.246	4.017	23.20
	AdaIN (2017)	0.021	0.071	0.220	0.001	21.40
	MicroAST (2021)	0.006	0.070	0.240	0.001	22.20
	MicroAST (2021)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
2048	AdaIN (2017)	0.016	0.079	0.246	4.017	23.20
	AdaIN (2017)	0.021	0.071	0.220	0.001	21.40
	MicroAST (2021)	0.006	0.070	0.240	0.001	22.20
	MicroAST (2021)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20
	AdaFA (2024a)	0.006	0.070	0.240	0.001	22.20

Method

- Frequency Decomposition: To better disentangle aesthetic styles.

VGG19

CS2470 DL Day: Smart Ink

12/12/2022

Tengfei Jiang, Yuanfeng Li, Zhangyi Shen



Scan this code to see our output video

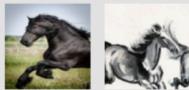
Introduction

For the purpose of converting real images into other styles, such as cartoons, neural style transfer has been extensively employed. We observed that most of them concentrate on western styles but we opted to explore the Chinese ink style, which is difficult because it resembles freehand brushwork. In our project, we applied GAN and additional constraints to make neural style transfer compatible with Chinese painting techniques like blank-leaving, brushstroke, or ink washing.

Preprocessing

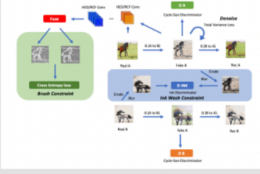
Our horse dataset consists of 718 horses and 912 images of paintings by the famous horse painting expert Xu Beihong.

For data preprocessing, all the images will be reshaped to a squared size of (256, 256). During the training, we realized some images in the datasets, such as a fake horse from a video game, yield comparatively poorer results. We customized the datasets by removing "fake" real images. We applied data augmentation techniques, such as flipping the images or altering the contrast ratio of the images, to the dataset because there aren't many authentic paintings and qualified images.

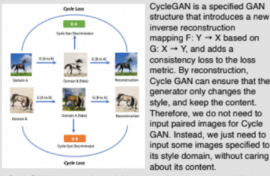


Model Architecture

We adopted CycleGAN as the basic model structure and added several constraints and refined the loss metric to make the model specified to the Chinese ink style.



CycleGAN



CycleGAN is extremely useful for our project since we can't find sufficient paired data consisting of real images and corresponding masterpieces of Chinese Ink Paintings.

Constraint—Brush Stroke

One important characteristic of Chinese Ink Painting is that the lines of the horse's body and texture should be very strong and bold. Therefore, for the generator, the generated image must be able to ensure that all body contours and textures are effectively retained. To force our model to focus on contours and textures, we added a new constraint to the generator loss.

First, we will use a holistically nested edge detector(HED) to get the edge maps EA and EB of the real image (real A) and the generated painting (fake B). Second, we will add the cross entropy loss between EA and EB to the generator loss. With this constraint, we can ensure that the generator will retain the texture information well and transform these textures to mimic the "brush stroke" effect.

We also implemented the Richer Convolutional Features Edge Detector(RCF). This is a more refined edge detector compared to HED.



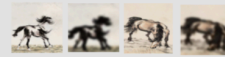
Constraints and Improvement



Chinese ink drawings are often black and white, thus the generated image's color distribution should be comparable to that of the original. As a result, we added a discriminator called D_J as a constraint to the CycleGAN model.

After the fake image was created, rather than immediately sending it to the original D Y to see if it was similar enough to a real ink drawing to trick the discriminator, we applied max-pooling and a gaussian filter to it. The blurred image will then be compared to the real image to see if they have comparable color distributions with the new discriminator D_J. Additionally, the generated fake image will be placed through the cycle and discriminator once again to see if it acquires the character of an ink painting. Therefore, we can enhance the quality of the output image by ensuring it still preserve the color distribution of the real image after transforming to ink-style paintings.

Example of the blurred image



Evaluation and Losses

To evaluate the quality of output images, we applied an evaluation model based on the paper NIMA: Neural Image Assessment which offers a method based on image classifier architectures. The CNN-based evaluation model aims to produce prediction with a correlation with not only noise, blur, or compression, but also aesthetic assessment of images. Along with the human aesthetic rating, the combined score has been used to improve our model in aesthetics.

To improve, we also applied SSIM loss in place of CycleGAN's original MSE loss function. The structural similarity (SSIM) index compares two images based on structural similarity, contrast similarity, and luminance similarity information to determine how similar they are. CycleGAN can achieve convergence more quickly and use fewer training iterations by using SSIM, which modifies pixel values from three dimensions and has a faster convergence speed than MSE loss in image reconstruction.

$$SSIM(x, y) = \frac{1}{N} \sum_{i,j} \frac{(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y} = \frac{1}{N} \sum_{i,j} \frac{(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y}$$

We also observed that there were some images with noise. Inspired by neural image style transfer, we also added a total variance loss in generator noise to denoise the image. It has the advantage of removing noise while preserving the boundary information of the image.

$$TVL(x) = \sum_{i,j} |x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}|$$

Results and Discussion

The ink style paintings which transferred from real images :

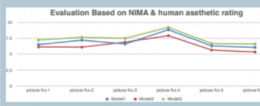


The "fake" horses generated by the model are improving in their similarity to real Chinese Ink paintings during epochs.

The Losses Decline during training:



The Aesthetic Evaluation:



Acknowledgements

- Similarity functions. CycleGAN. [arXiv:1703.10590](https://arxiv.org/abs/1703.10590), 2017. Retrieved December 10, 2022, from <https://arxiv.org/abs/1703.10590>
- Unpaired image-to-image translation using cycle-consistent adversarial networks. CycleGAN Project Page. (n.d.). Retrieved December 10, 2022, from <https://github.com/junyanz/CycleGAN>
- He, B., Gao, F., Ma, D., Shi, B., & Duan, L.-Y. (2018). Chogan: Proceedings of the 26th ACM International Conference on Multimedia. <https://doi.org/10.1145/3209088.3209093>
- Agarwal, (n.d.). ADAPTOR: Aesthetic NIMA: A Pytorch NIMA implementation using Denoiser. Nima is a research endeavor that rates the aesthetic quality of images. GitHub. Retrieved December 10, 2022, from <https://github.com/agargal/MU-Aesthetics-NIMA>