Artistic Style Transfer Network
Project Milestone

Mengying Bi (mybi), Cathy Jia (cathyjia), Geoffrey Li (geoffli)

Problem Description

Our project goal is to apply convolution neural network (CNN) to design and develop an artistic style transfer tool. With deep CNN, it is possible to extract both the *content* and *style* of images. The problem we want to solve is to find an appropriate network design for the style transfer, to define a loss function that optimizes for combination of the original image's content and the selected artistic filter's style. We can fit as many different types of networks, hyperparameters and art works as possible. Style loss and content loss will be used to judge how well a network transfers artistic style to a given image. By the end of this project, we expect to get a network that successfully transfers a set of random images and generates into styled images efficiently with minimal noise. We also plan to design a web interface in which users can upload an image and select an artistic filter to generate a new image with style transferred.

Dataset

We will use a standard Image Classification benchmarking dataset, ImageNet, to train a deep CNN to learn low-level and high-level feature representations of common types of objects. The artistic style data will be a collection of images with various artistic styles. The intermediate layers of this CNN (without the fully-connected layers) will allow us to perform the content and style transfer.

Image Data Info	Artist(s)	URL
ImageNet dataset	N/A	http://www.image-net.org
European fine arts (3rd-19th centuries), high resolution painting images	van Gogh, Monet, Gauguin, Da Vinci, Rousseau, Gainsborough	https://www.wga.hu
Chinese ink wash painting	Baishi Qi, Beihong Xu	https://www.comuseum.com/painting/
Pablo Picasso's famous painting	Picasso	https://www.pablopicasso.org

Methods, Progress and Plan

We have read current state-of-the-art research papers related to style transfer using CNN. We tested several core PyTorch built-in functions which will be useful in network implementation. We decided to use VGG19 (Fig.1) as the baseline CNN architecture for implementing our own neural network.

- For our preliminary baseline, we will use the intermediate layers (feature-level) of the trained network to use to extract "content features" from the input content image. From the input artistic images, we will extract the "style features" particular to that art piece (derive the spatial correlations in each of the layer's activations) by computing the Gramian matrices across different feature maps

$$G^l_{cd} = \frac{\sum_{ij} F^l_{ijc}(x) F^l_{ijd}(x)}{IJ}$$

- Then, we'll set up a dual loss function, where the loss function is a weighted average of one loss that optimizes towards the content features of the input content image, and one loss that optimizes towards the style features of the input art image. In this way, the combined image will optimize towards minimizing both content loss and style loss. One hyperparameter of our network is the relative weight of optimization towards content and style loss, a parameter we intend to experiment with and evaluate how the results turn out.
- As an enhancement to our network, instead of transferring artistic style from a single image, we aim to extract features and style from a set of painting works by an artist. Again, the style loss and content loss will be computed to evaluate how well an artistic style is transferred.

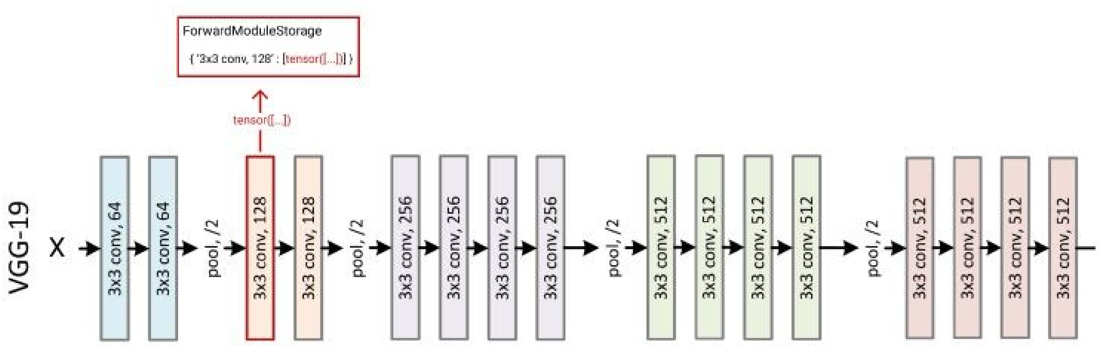

Fig. 1: VGG19 network architecture.

We have tested the performance of a pre-trained VGG19 network transferring a chosen style onto a given image. The style loss and content loss were reduced as we increased the number of transfer run. For our initial results, we got a style loss of 0.626190, and a content loss of 4.148500 after 500 runs. The results are shown in Fig. 2 below.

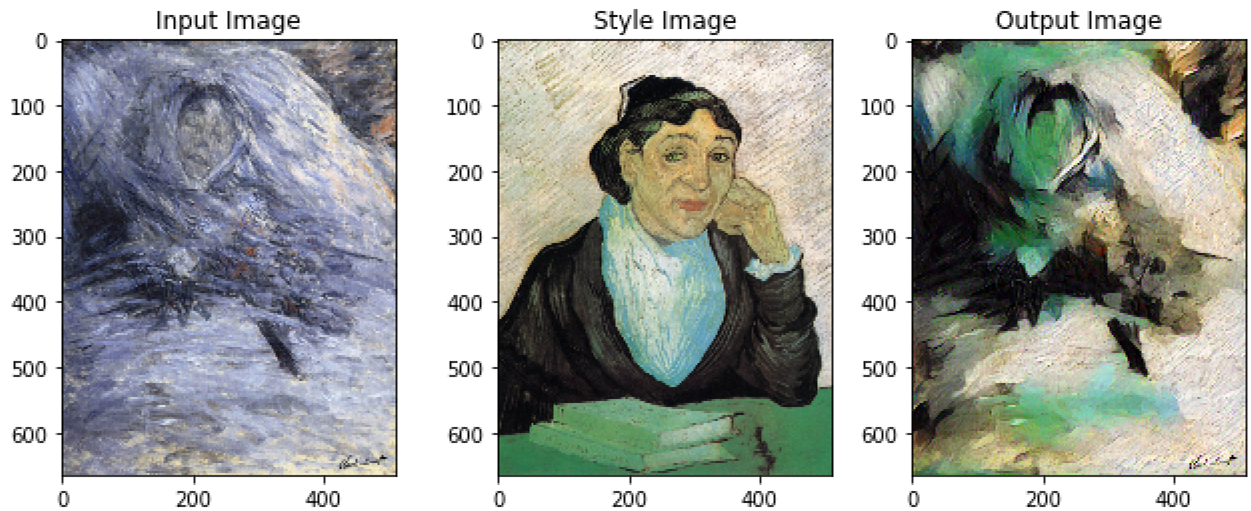

Fig. 2: preliminary results of applying the Style to the Input image.

Through our plan, we have several potential challenges:

- The image sizes of input images are uniform. We will write an image loader with size conversion so the model will accept images with different sizes. The size conversion can be done with rotating, resizing and cropping. The problem is how to optimize this image loader with minimal resolution reduction.
- For network enhancement, we plan to extract features from a certain artists rather than a piece of artwork. We need to consider the dimension of style tensor when we do the style transfer, or an alternative way of combining multiple style tensors.
- We will also need to experiment with different regularization parameters. Without regularization, we've found that high frequency artifacts (noise) are often produced in the output images, essentially resulting from an over-abundance of edge detectors. By using a sufficient regularization parameter on the higher frequency components, these artifacts can be avoided.

There are several things we would like to try for potential improvements on our model:

- For portrait photos, we want to preserve the human face while applying the artistic style only to the background, so our generated image will not be as "abstract" as some of the original pieces. We will plan to implement this feature using inspiration from object detection algorithms to apply only localized style features.
- Another enhancement we are considering is trying to develop a "reverse style transfer" i.e., taking an input art image and using the learned features from our pretrained CNN

layers to produce a “realistic-looking” image from the art image. Our hypothesis is that this would work best only for types of objects already contained in our pretrained model (i.e. the types of objects in ImageNet). As far as the mechanics are concerned, this may involve aspects of image retrieval algorithms, where we may need to first find a similar “realistic image” that the model has seen before and then use this as the basis for the content optimization.

- We would like to also include options to select different aspects of style transfer; for example: preserving original content colors but transferring artistic style.

Time Schedule

Week	Task	Due Date
Week 9	Collect art style image source	11/19
	Project milestone	11/20 11:59pm
	Finish the first basic network for testing	11/23
Week 10	Refine and improve network Data training	11/27
	Develop UI in Plotly Dash for live demo	11/27
	Add improvement and extensive work	11/30
Week 11	Poster session	12/4 3:30-6:30pm
	Project writeup	12/4 11:59pm