Deliverable 2: Neural Artistic Style Transfer

1. Problem Statement

Neural Style involves forming by retaining the style of a style image and combining it to the content of a content image.

2. Data Preprocessing

For my final implementation, I'm going to use the COCO Dataset which is already split in training, test and validation sets. Since this is a first implementation though, for the moment I'm only using the pertained VGG-19 Neural network, which been trained on a large dataset of pictures. I figured since this isn't the final submission, I wanted to get to experiment with a model performing regression on the content and style losses.

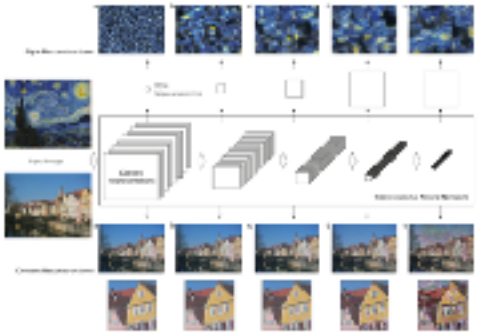
3. Machine Learning Model

For the current deliverable, I'm using a model based off of Leon A. Gaty's and al. paper "A Neural Algorithm of Artistic Style". This model consists of separating the style and content representations in the VGG-19 Convolutional Neural Network. We then select layers from the neural network on which we shall compute loss functions. We first compute the content loss by making a list of layers where we want to compute the content loss. We then calculate the Euclidean distance between our content image output and pastiche image output at the given layer. We then repeat this process for the style image, but this time, we compare the Gram matrices of the outputs at the given the layer. We then calculate the total loss by summing the style and content losses and want to minimize this total loss. Here is the formula:

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

Now one tweak I made to this formula is that instead of multiplying alpha and beta to the content and style losses after summing each of their Euclidean distances, I multiplied each representation by alpha and better before finding the difference and summing since this produces a stylization much quicker. Here is the formula:

$$\mathcal{L}_{content} = \sum_{i} \sum_{j} (\alpha C^{l}_{i,j} - \alpha P^{l}_{i,j})^2.$$



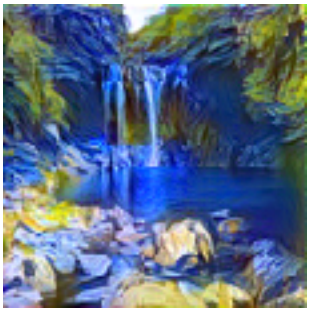
Convolutional Neural Network from Gaty's paper

4. Preliminary results

Content Image

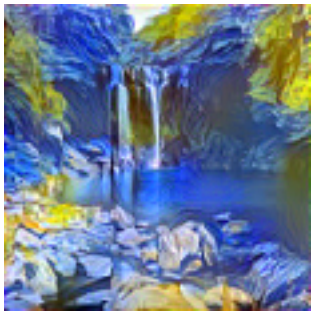
Style Image

After 1st iteration



After 10 iterations

After 20 iterations



After 30 iterations

5. Next steps

The next step would be to implement the architecture specified in Johnson and al. paper “Perceptual Losses for Real-Time Style Transfer and Super-Resolution”. The main issue with trying to minimize the content and style losses is that it’s very slow. Therefore, in order to speed things up, I would add a feedforward image transformation network. With this architecture, we would train a network a network to do stylization beforehand in order to produce stylized images instantly. As mentioned earlier, for my first implementation, and in order to understand the basic concepts, I simply used Gatys’ model.