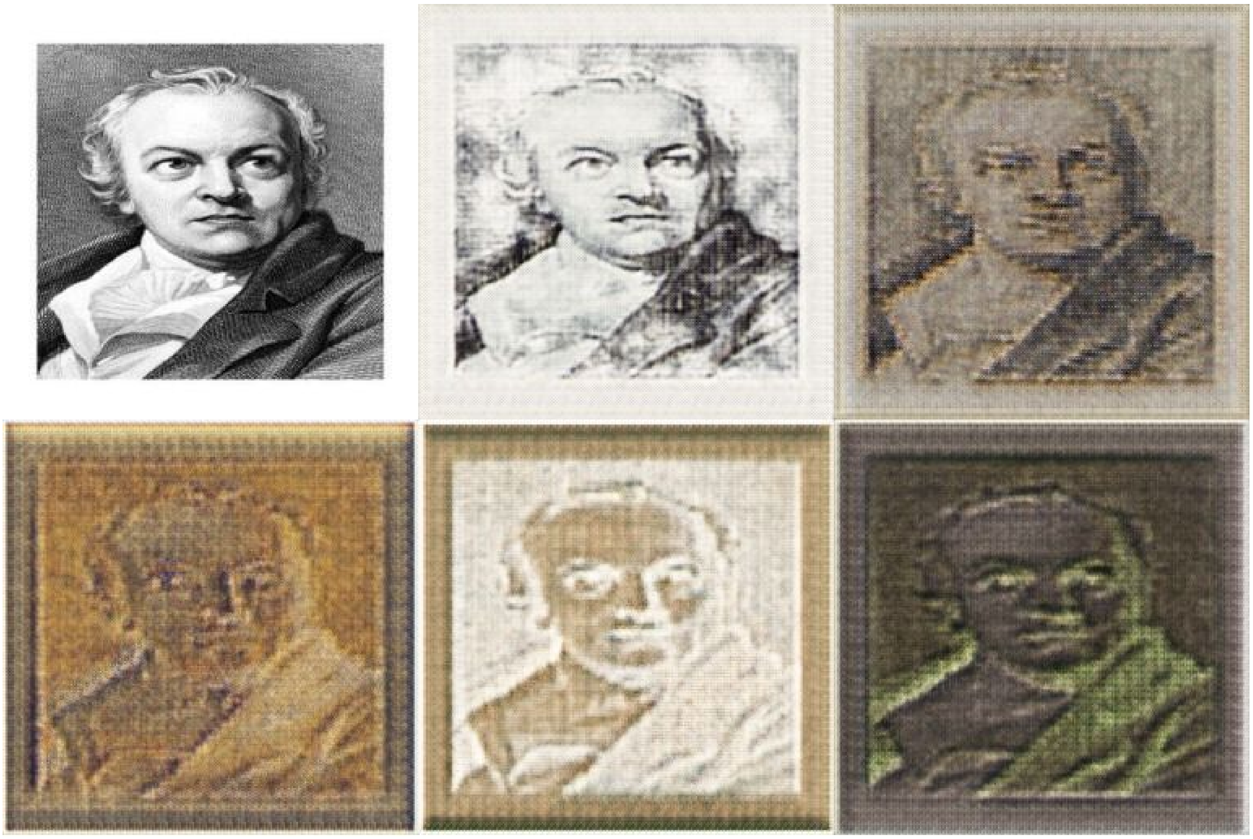
Machine Learning for the Arts
UCSD SPRING 2019
**FINAL PROJECT**

# Image Stylization by William Blake using CycleGAN



Justin Law

## DESCRIPTION

### Concept:

The idea for this project was to take any image and stylize it to how William Blake would draw the image. Eventually, this converged to the idea that I would have 5 different CycleGAN models. These goals were more or less achieved, but because of the limited information that the CycleGAN can encode I had to settle with two different sets of models: image coloring (see models color1,color2,color3) and image stylization (see models sketch and mix).

The main inspiration for this project was simply viewing William Blake paintings and sketches and noting the unique coloring schemes he uses. I believe that his style is perfect for use in image stylization, as each art piece is distinct; but also if the images are selected correctly, he reuses the same coloring schemes and has a distinguished style.

### Technique:

The cycleGAN implementation was based off the keras ipynb from the ece 188 spring 2019 class [1]. I had the modify the implementation in the following ways. As a side note, I refer to each model by the name of the folder of my github implementation [2] throughout this writeup.

#### Training set A

The training set A images for color1,color2,color3 were boat images from the paintings dataset [5]. The training set A images for mix was all the paintings from the paintings dataset [5]. The training set A images for sketch were chair images from the paintings dataset [5] (this was chosen because ~90% of these images are portraits). Training set A images are about 8629 in total and 2143, 1202 for boats and chairs respectively.

#### Hyperparameters and implementation

For the color1,color2,color3 models I preprocessed the training set A to be black and white. I reduced the hidden layers from 3 to 2, as with 3 hidden layers the generation will try to replicate too much of the William Blake image.

With 2 hidden layers, the model is able to extract just the color and not converge to an image that tricks the discriminator. The learning rate for the color1,color2,color3 models were kept at 2e-4 for both the discriminator and generator. The biggest and most important change was to use SGD (stochastic gradient descent) for both the generator and discriminator. This increased the runtime but the results were much higher quality and generalized to the validation set better. The model was also modified from 128x128 input images to 256x256 images, which increased the training time.

The color1,color2,color3 models take roughly 30 minutes to train on 1 2080ti nvidia GPU. I added the option to train on multiple gpus, but possible due to tensorflow keras this actually made the models train slower. To reproduce the results, it is best to use a multiple GPU machine and try to train multiple models at once. The code was modified to make this possible by making each model train on an individual GPU.

The mix models and sketch models also have a learning rate of 2e-4 for both the discriminator and generator, use SGD (stochastic gradient descent), and take 256x256 images. The mix model is trained for 200 epochs. The sketch model is trained for 1000 epochs. Both the sketch and mix models have about 10-15 images for training set B. Other than these differences the implementations are almost exactly the same.

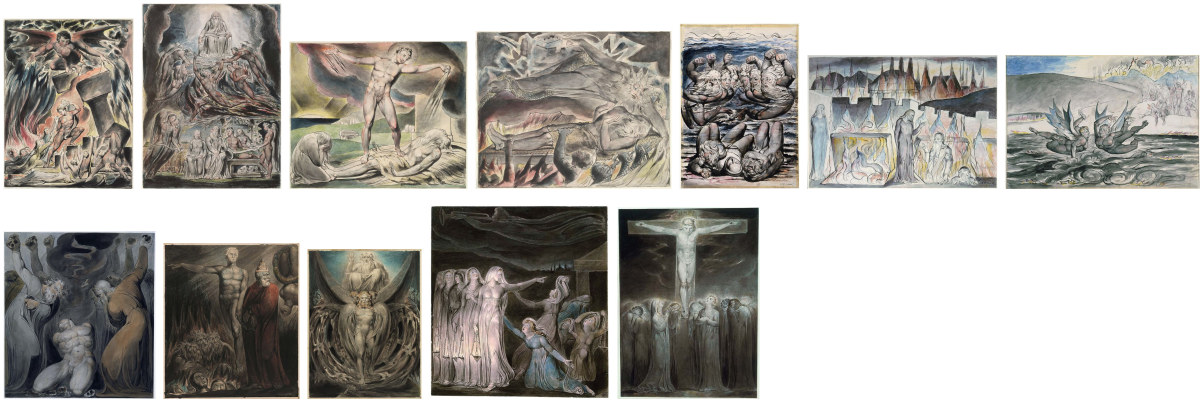
## Process:

The cycleGAN paper and website [3],[4] were helpful references for this project. In particular when it describes common pitfalls of the CycleGAN, namely that the discriminator will often be tricked, as well as the importance of the training set B (the training set that I use to stylize). From the paper I was inspired to go back to project 4 and curate the images, in which I ended up with 5 different datasets corresponding to the mix,sketch,color1,color2 and color3 models. Mix is a dataset of William Blake paintings/sketches with varying colors. Sketch is a dataset of black and white sketches from William Blake. Color1, color2, and color3 are all select images that had interesting coloring schemes that I hand picked. The training images (referred to as training set B or simply the William Blake images throughout this paper) are shown on the next page [figure 1].

The process was mainly intuition along with trial and error. I had two main insights. One key insight I had was that for coloring the CycleGAN should use two hidden layers, as the generator was learning to trick the discriminator by the time the features of training set B (the William Blake images) were learned. The other insight was to use stochastic gradient descent rather than Adam, along with a dropout rate of .4-.7. I had to use a lot of trial and error to find the right dropout rate for each model. I also used a lot of trial and error to find the correct number of epochs. Too many epochs caused the generated images to all converge to one image, and too little epochs had blurry outputs from the generator or simply results with high loss.

# Figure 1 - Training set B for each Model

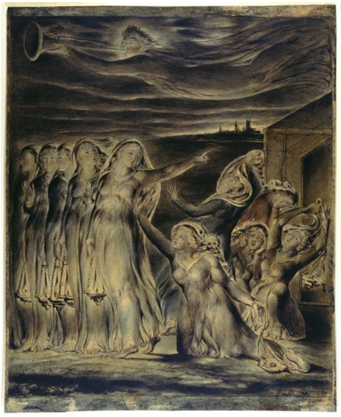
paintings/dataset/ was trained on:



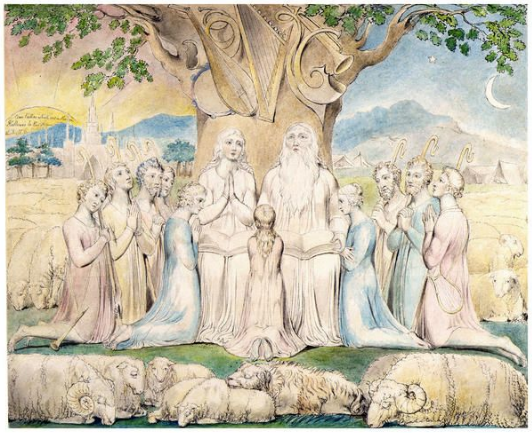
sketch/ was trained on:

Death's Door
I found him beneath a Tree
Earth
He struggles into Life

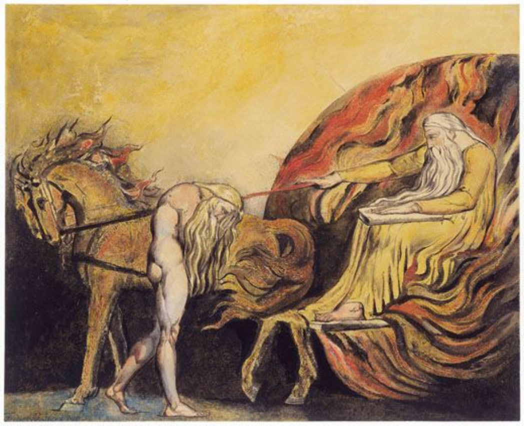
color1/ was trained on:



color2/ was trained on:



color3/ was trained on:



### Result:

The images displayed in the **RESULT** section on pages 7-12 are images generated by the 5 CycleGAN models with inputs of images that were selected by me. It should be obvious from coloring of the image which model they are generated from. I placed original images (before the forward pass and before conversion to black & white for the color1,color2,color3 models) next to the generated image to make this clear.

If you want to try the models on your own images, use the generate/ folder on github. You will need to download the weights in the form of h5 files. The process for this is explained in the github README. The only requirement for the image is that can be reshaped to 256x256, meaning that if it has less pixels the models will not work.

### Reflection:

Recall that the color1,color2,color3 models have 2 hidden layers and correspond to the purpose of coloring. Mix and sketch are used for stylization. Sketch seems to give the best stylization result, and the color1,color2,color3 models are all good at coloring in the image.

As mentioned before both models use 3 hidden layers, and the generator is able to extract the style of William Blake images. The sketch model is a perfect example of this. For the sketch model the generator can take any image and it will learn on its own to make it black and white, and add a stenciling the is similar to the William Blake sketches.

A note on implementation. My github implementation [2] has a folder for each model and the hyperparameters vary from model to model. But they share code with each other and are practically all the same other than the number of hidden layers and the preprocessing step for the coloring models. A more reusable implementation could parameterize one python notebook for number of layers and the how to preprocess training set B.

## Future Directions:

The sketch GAN and color3/ are my favorite effects. From here, I could envision using the pix2pix HD described in https://github.com/NVIDIA/pix2pixHD in order to do what I was trying to do with color3/ in HD (and better results), or creating a CycleGANHD and implementing the sketch GAN, or trying to recreate the effect of the sketchGAN by finding William Blake's sketches and then labeling them with the corresponding painting. The latter idea could only work if there is enough examples (which is doubtful).

The results *could* be improved by using the wikiart dataset as training set A. The dataset for training set A is clearly not optimal, as it meant for classification. But the variation in the dataset from using the painting dataset [5] could also be a good thing, causing the models I trained to generalize better to validation images.

## REFERENCE:

[1] cycleGAN in keras:  https://github.com/tjwei/GANotebooks
[2] github implentation https://github.com/ucsd-ml-arts/ml-art-final-justin-law
[3] cycleGAN paper: https://arxiv.org/abs/1703.10593v6
[4] cycleGAN website: https://junyanz.github.io/CycleGAN/
[5] paintings dataset: https://www.robots.ox.ac.uk/~vgg/data/paintings/
[6] wikiart dataset: https://github.com/cs-chan/ArtGAN/tree/master/WikiArt%20Dataset

**CODE:** https://github.com/ucsd-ml-arts/ml-art-final-justin-law

## RESULT:

From the style of the image it is clear which GAN they are generated from.

generated image original image