

### Introduction:

We are generating art. We will begin by working with the abstract art dataset, but we hope to generate art based upon the landscape and portrait datasets as well. This is an unsupervised learning problem.

### Related Work:

This blog post uses pytorch to generate art using GANs.

<https://blog.jovian.ai/generating-art-with-gans-352ceef3d51f>

The author used a dataset “Best Artworks of All Time” of 64x64 images found here:

<https://www.kaggle.com/ikarus777/best-artworks-of-all-time>

They used a CNN architecture for their discriminator. Generator input was vectors of random numbers to be used as a seed for images, and the generator architecture was a transposed convolution (ConvTranspose2d layer from PyTorch). They trained the discriminator and generator in tandem, alternating between them for a few epochs at a time, using an Adam optimizer. Their results were decent, although from a preliminary viewing, the model tended to have trouble with representing tangible objects and people in its generated images.

### Data:

We will be using datasets of images taken from WikiArt.org. The images have already been compiled on Kaggle (<https://www.kaggle.com/ipythonx/wikiart-gangogh-creating-art-gan>) and split by art genre. We are interested in using abstract, landscape, and portrait art, each of which has 15,000 images. Before using this data, we'll have to resize images to make sure they are all the same shape and resolution.

### Methodology:

We are using GANs to generate the artwork, so our architecture will consist of three main networks, the generator and the discriminator. The discriminator will be a CNN, as it is being used to classify images. Similar to the aforementioned blog post, our generator architecture will likely be a transposed convolution, or deconvolution. The adversarial network

### Metrics:

Base Goal: Use GANs to create artworks based upon abstract art dataset

Target Goal: Use GANs to create artworks based upon abstract, portrait, and landscape datasets, comparing differences

Stretch Goal: Have GANs generate abstract art realistic enough to be mistaken for real art by a human, not just our discriminator

### Ethics:

Our dataset is a subset of the Wiki-Art visual art encyclopedia. There are some ethical considerations as to how it was collected, as the Wiki-Art database contains some artwork without the artist's permission. On Wiki-Art's site, they say “mostly we try to get artists' permission to demonstrate their art”, but this implies that there are also pieces of art on the site that are being used without permission. If we are training our model on art without the

permission of the artists, it would seem ethically questionable, especially if our model brings us any type of financial gain.

Another ethical question that comes up with art generation is whether or not art should actually be automated in this way. If we are successfully able to generate art using deep learning, does that hurt human artists? Just as jobs in industries such as manufacturing have been automated away, it's possible that artists' jobs could be taken over by automation as well.

Division of Labor :

Nick - image preprocessing,

Olivia and Kevin - begin creating the GAN

### **Update:**

Introduction: We are generating art. We will begin by working with the abstract art dataset, but we hope to generate art based upon the landscape and portrait datasets as well. This is an unsupervised learning problem.

Challenges: We've run into a few difficult bugs in implementing a simple GAN network. Our biggest challenge that we are working on currently is changing our generated images from black and white to color by creating 3 channels.

Insights: We don't have any concrete results from our model just yet. Although we expected training GANs to be difficult, it has surpassed our expectations.

Plan: Currently we are using dense layers in our generator and discriminator networks, but we will change these to convolution layers. We are also currently only using 1 channel, so our images are coming out black and white, so we will be changing this to have three channels for RGB.