# Art Generation Through Deep Learning

By Andrew Le, Dylan Kral, Lam Nguyen, Eric Thompson, Emmanuel Alozie

## Abstract

We tend to imagine computers as rigid, objective things which take in data and produce some set output every time. Deep learning takes us a step away from this, breaking down that data through artificial neurons which attempt to mimic the structure of a human brain, drawing connections between that data and previous knowledge. Often, this technology is then used for the purposes of predicting given data, such as in image classification or natural language processing. But can we take it another step further from that initial rigidity? Can we use this technology not to identify or classify, but to *create*. We are going to see how deep learning can be used to create something which has no set output, something subjective and deeply human — artwork.

## Introduction

Artwork has always been a staple in human culture with this being traced back to early humans and cave paintings. The beauty and importance of it is it allows us as humans to express ourselves and tell various stories. With the recent development of technologies we can observe that there is also a concurrent growth in the advancement of artificial intelligence and deep learning. For example, there exists an application that uses an AI module to monitor the health of crops and agriculture. With this in mind, we can't help but wonder about the extent of deep learning and if it can be developed to comprehend and appreciate art which has been widely considered exclusively for human beings.

In this project, we will be developing a Generative Adversarial Network to be trained to create imagery and art using a variety of different input data. The expected outcome or hypothesis of the result is quite ambiguous as the goal of our project is to essentially have a non-human create what is generally derived from human emotions and thought. We will then analyze these output to better understand why it is performing the way it is and if we can optimize the model to tune it or even generate completely different pieces.

## Related Work

In the early development of our project, there were two pieces of research that inspired us the most and contributed to many decisions we made with our own GAN.

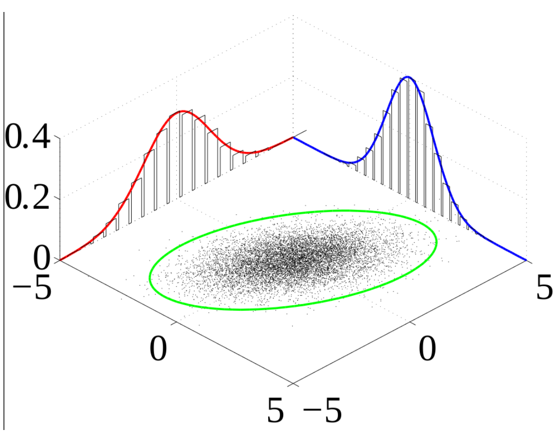
The first piece was Sakib Shahriar's *GAN computers generate arts? A survey on visual arts, music, and literary text generation using generative adversarial network*. With Shahriar's research we were able to understand if our goals were even possible since the scope is quite unique in that the output will be subjective due to art's nature. Shahriar discusses how using a GAN would be the most optimal fit in generating art and compares this with not just visual art, but the generation of literary text and also music. Although there are still many challenges and ambiguity with art generation in relation to deep learning, this piece gave us a direction for our project.

The last piece that we gathered information from was Alexander Mordvinstev's work with *Deep Dream*. What's interesting about this piece in particular was that it solidified why we had to use a GAN for our project specifically. In Mordvinstev's research he discusses how he used a Convolutional Neural Network, or CNN, instead of a GAN. The reason why a CNN is used instead is that for his project, he wants to detect and intensify patterns within the input data using algorithmic pareidolia. This was very important for us to note as it helped us consolidate the plan of creating a GAN due to the comparison of our scopes. Although Mordvinstev inspired us to undertake this project and mission, his work helped confirm the direction and methodologies we used in our project.

## Background

### Generative & Discriminative Models

With *discriminative models*, the goal is to classify given data in some way, taking some input data and assigning it to a predicted class. In contrast, *generative models* will take in input data and attempt to create new data which fits in with the given dataset.

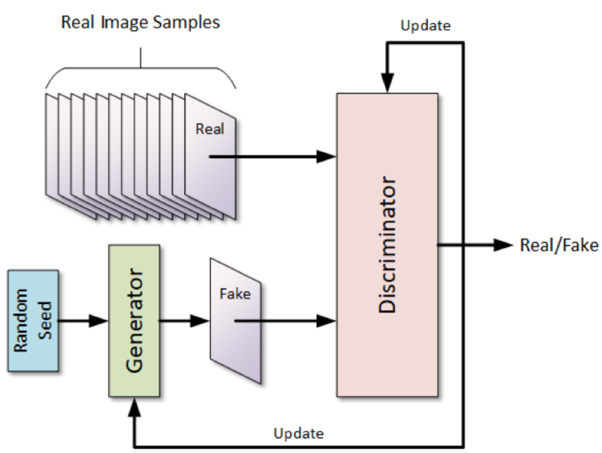



Example of joint probability between two sets of data

Discriminative modeling is based on the conditional probability — that is, the probability of one point of data given another. Generative modeling uses joint probability: the collective probabilities of every combination of data. These probabilities form the basis for the generative models' ability to create data containing the different features associated with a given output class, resulting in an output which is entirely new, yet fits in with the data it was trained on.

**GAN**

The GAN, or *Generative Adversarial Network*, is a type of generative model based on the use of neural networks. It is composed of two different models: The *Generator* and the *Discriminator*. The Discriminator is simply a classifier, it is trained on a dataset and then determines whether data we pass in fits with that given dataset or not. In our case, it uses binary classification — true or false, art or not-art. The Generator has the goal of creating new examples of the dataset, ones which are good enough to fool the Discriminator. As the models run, they work to train each other. The Generator creates fake samples to be fed in with real ones through the Discriminator, which attempts to classify them. The Discriminator improves its classification ability as it runs through these different samples, becoming more difficult for the Generator to trick. Conversely, the Generator is trained based on how well the Discriminator is performing, such that it gets better and better at generating new data.

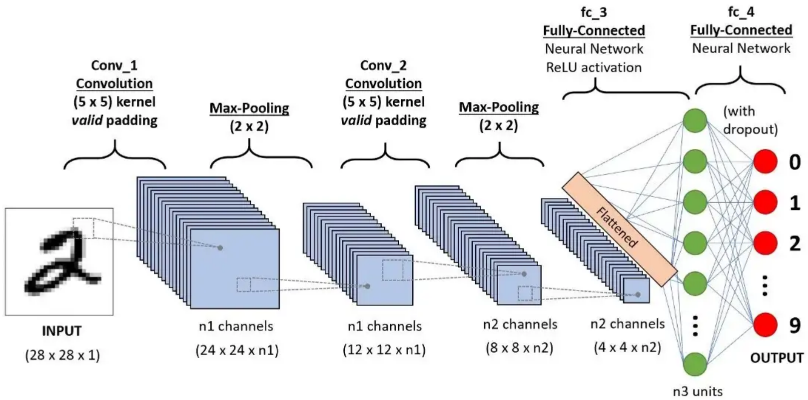


Architecture of a GAN, showing the relationship between the Generator and Discriminator

The Generator is fed in a vector of random starting values, essentially random noise, as its input. This random noise corresponds to a location in the *latent space*, a distribution of all the data points which the Generator maps to unique outputs, meaning every combination of random input values corresponds to a unique image (though this mapping changes every time the Generator is trained). Ideally, the GAN reaches a point where the Discriminator can never tell the difference between the real or fake results.

**CNN**

A CNN, or *Convolutional Neural Network*, is an ideal choice for image identification, making it perfect for constructing our Discriminator. It breaks an image into chunks of data rather than flattening it and going pixel by pixel, so that it does not lose the context of each point of data in relation to its surroundings. The CNN's *kernel* is the shape of each piece the data gets broken down into (I.E. 5x5x1 = 5 pixels x 5 pixels x 1 depth). This also allows us to avoid processing the entire image at once, meaning less computational resources used and an easier time processing our data.



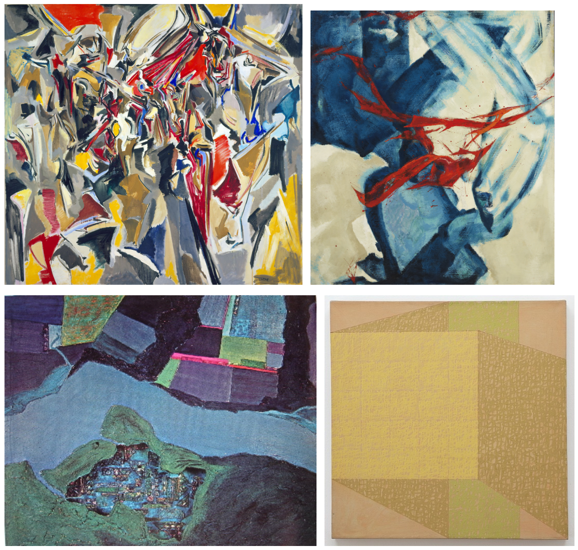

Architecture of a CNN

# Workflow

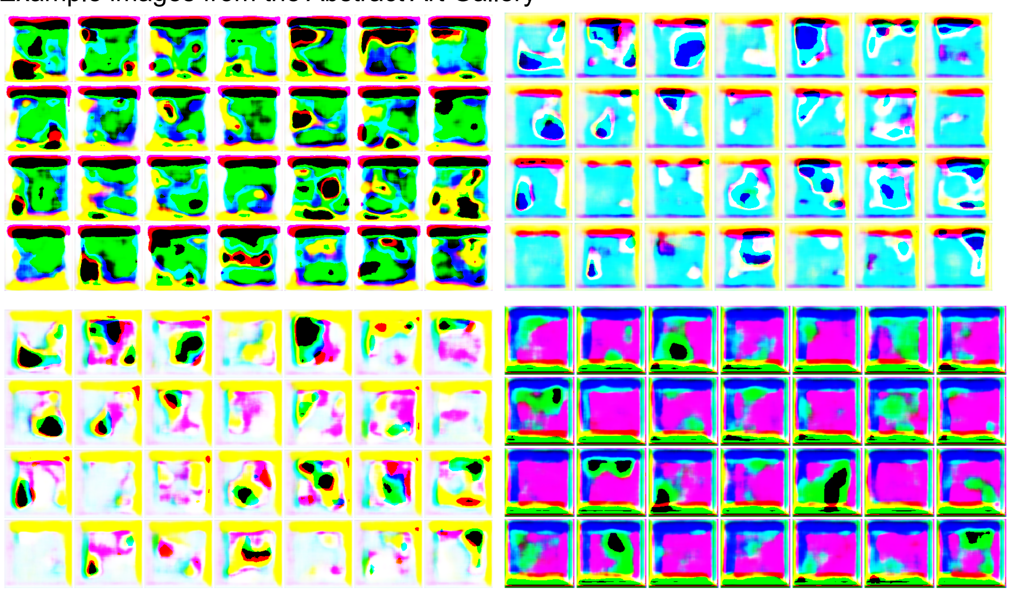
1. Gan was chosen as the approach for this project.
    a. Unsupervised Learning while creating high-quality synthetic images/data
    b. Want to produce sharper discrete data as opposed to blurry averages from other Networks
2. The “Abstract Art” dataset will then be chosen and tested.
    a. Abstract art tends to contain a multitude of primary shapes easy to train in few steps.
3. The data will then be distributed throughout the model
4. The model will then be trained against the data.
5. Parameters and Sizes may have to be adjusted if needed.
    a. Taking into consideration overfitting, underfitting, layer features, number of layers, and initial parameters.
6. Testing stage.
7. Repeat
    a. It is common to refit the model with new parameters, layer refinement, and other variable remastering (Step 5).

# Findings

Throughout this project, the dataset that was used initially is called “Abstract Art Gallery”. Although in the previous report, we have stated that this dataset may not be the best option when it comes to this project because the contents covered in this project may be too broad and thus making it difficult for the model to learn. However we still kept on with this dataset, and continue to implement it in our project. A few sample images of this dataset can be found in the following images below. In addition the dataset itself can be found in the following link: “https://www.kaggle.com/datasets/bryanb/abstract-art-gallery”.

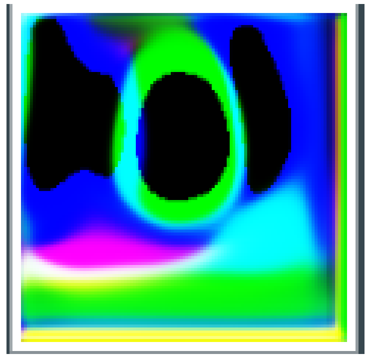


Example images from the Abstract Art Gallery

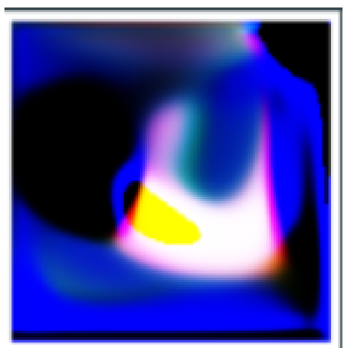


Initial results of our GAN model

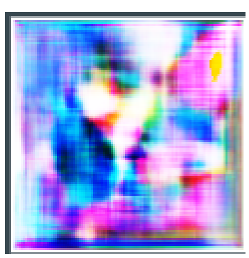
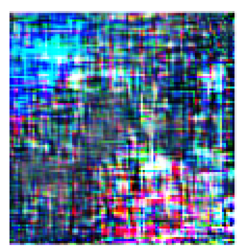
During the last report, these images above were the result of the generation from the model. The images had a very chaotic appearance, with color textures that were all over the place. With that being said, the outputs generated by the model were very unappealing to the regular eyes, so this was the reason why we really thought that this dataset was not suitable for the implementation of this project. However, there was a significant improvement in terms of visual aesthetic when implementing this dataset later on. The following images below were images that were generated using the “Abstract Gallery” dataset. The aesthetic aspect of the images is significantly better than the outputs from the previous report.

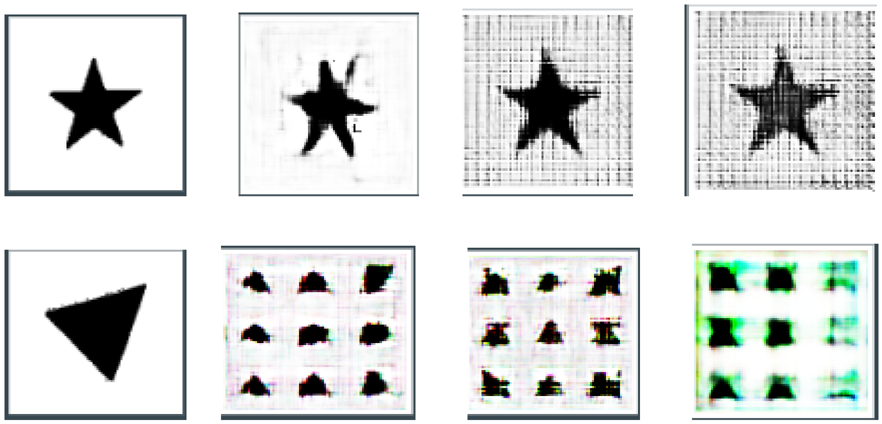

In addition, one of the group members ran the same program using the same dataset but with an adjustment of using lower batch size, and these images below were the result.




In addition, another type of input with the name of  “Cubism” was also used as input in order to train the Gan Model as well. The following pictures below were generated from using the new input of “Cubism”.

Initially, the photos generated when using the Cubism files were very blurry, with the image being mostly gray, with slight signs of colorful patterns in the background, but as time goes on, the images get clearer and clearer. In order to expand the diversity in the appearance of our result, we have also implemented another dataset. The name of this new dataset is called “Stars” and when using this dataset, the output was very normal at first, because the images that were generated were initially just a single regular star, but later on, as the model continues to run, the output then starts to get bizarre. For example, instead of generating a regular star, the model started generating fragments of the star, such as only generating the corner section of the star. Eventually the model started generating multiple fragments of pieces from the star, and eventually the images started to take on a green color.



*Note- These images were created using 1500 epochs

Following the tests and trials above, there are a few considerations and changes we would like to implement moving forward. Firstly, a huge obstacle that affected the entire scope of the project was the fact that a significant amount of runtime and resources were being directed towards working with 128x128 images. A solution that could be experimented with is the concept of dividing-and-conquering. The idea is that higher quality imaging can be obtained while keeping high accuracy and sample diversity by breaking down the imaging into smaller components. However given the scope and scale of the model, we realized parameters such as batch size, kernel size, and epoch heavily influenced our test results. Given more refinement and observations we may have been able to produce better images.

Secondly, another aspect that we would like to address is the input data itself. As we all know, art is subjective and any imagery could be argued as a piece of art. Therefore, as we continue to develop this program towards creating imagery outside the scope of abstract art, there needs to be more focus on what kind of input data is being fed into the model. With many art styles, there are specific conventions and styles that may affect the output. Moreover, although the output is based on the idea of the machine creating its own art, these methodologies need to be further parameterized moving forward to create distinction between different expected outputs.

Finally, the last consideration we would like to explore is the usage of different activation functions and optimizers. Although many tests and epochs were iterated, there are still millions of different combinations and patterns that we could implement to better train the model. An important thing to note too is that certain models may work better with certain art styles and be worse in others. Not only do the activation functions and optimizers play a huge role in the output of the program, but the epoch size and other parameters do too. All of these important aspects of our deep learning model should be experimented to observe changes in the model that are both good and bad. This way we can better optimize the program in respect to different use cases and art styles.

## Conclusion

In conclusion, the model that we created successfully performed the task we set out to solve, and that was to create images using a generative adversarial network. When using black and white images as the input, the model was able to display very intricate images of different shapes such as stars, circles and triangles. The model did great when testing the Abstract Art galley as the input but since the data was very broad, it seemed to have a hard time distinguishing what was considered abstract art and what wasn't. Though subjectively the images could be considered stunning and abstract in their own ways. The overall image size that we were able to generate was only 128x128, mainly being limited by memory space on the allocated graphics cards of google collab. This restriction in size made it difficult to branch out to higher resolution images with more detail to possibly attempt different styles of art.
For any attempts made in the future to build upon this idea, it would be important to work on a way to output higher resolution images in order to create usable content for different situations such as phone, laptop, or desktop background generation, or even apply it to different fields of graphic design as it could be utilized to generate new logos or profile pictures without the need of hiring a designer, leading to a very lucrative business.

## References

https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

https://www.analyticssteps.com/blogs/generative-vs-discriminative-models-machine-learning

https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

https://www.sciencedirect.com/science/article/abs/pii/S0141938222000658

https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html