Generative Adversarial Networks Take on Hand Drawn Sketches: An Application to Louisiana Culture and Mardi Gras Fashion

Stephanie Hines

Follow this and additional works at: https://repository.lsu.edu/honors_etd

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Recommended Citation
https://repository.lsu.edu/honors_etd/690

This Thesis is brought to you for free and open access by the Ogden Honors College at LSU Scholarly Repository. It has been accepted for inclusion in Honors Theses by an authorized administrator of LSU Scholarly Repository. For more information, please contact ir@lsu.edu.
Generative Adversarial Networks Take on Hand Drawn Sketches: An Application to Louisiana Culture and Mardi Gras Fashion

by

Stephanie Hines

Undergraduate honors thesis under the direction of

Dr. Anas Mahmoud

Division of Computer Science and Engineering

Submitted to the LSU Roger Hadfield Ogden Honors College in partial fulfillment of
the Upper Division Honors Program.

April 2022

Louisiana State University

& Agricultural and Mechanical College

Baton Rouge, Louisiana
Abstract

A Deep Convolutional Generative Adversarial Network model and its hyperparameters are adjusted and compared in order to produce hand-drawn sketch-like images that are based on Mardi Gras parade costumes.

1 Introduction

Starting as early as the 1960s, computers have been used to create art. Machines conceived to crunch numbers were yet used for the humanities. Early art was geometric and algorithmic in nature, since early technology usage was mostly limited to mathematicians and scientists, and user interfaces were limited relative to modern day advancements. In the modern day, computers and other devices are used daily by hundreds of thousands of users to edit photos, draw pictures, animate models and more. From Instagram to Photoshop, user friendly yet advanced tools aid in the creation of creative works.

In 2014, Goodfellow et al., published a paper detailing a new advancement in generative networks. Called Generative Adversarial Nets (GAN), this generative model used two opposing models to train and produce more and more realistic data. Radford et al. (2015)[8] released a follow up paper detailing a modified approach that uses layers in the generator to produce more and more detailed images.

Louisiana has a rich culture and history of art, spectacle and celebration. Mardi Gras is the most well known concentration of this art and culture. The earliest celebration of Mardi Gras in Louisiana dates back to the 1700s, and the first New Orleans Mardi Gras parade rolled in 1857. New Orleans parade Krewes have captivated tourists and locals alike with their extravagant floats, costumes and throws. Currently, over 70 Krewes host parades in the New Orleans area.

This paper explores the application of DCGANs for creating hand-drawn sketches, specifically modeled after historical Mardi Gras parade costumes. This computer generated take on an art form that has impacted Louisiana serves to visualize just what exactly makes a great costume.
2 Existing Literature

2.1 GAN origins

Generative Adversarial Networks are a form of artificial intelligence based on an unsupervised learning approach. GANs work by using game theory between two players: a generator and a discriminator. The generator uses a generator function and previous values to turn unstructured noise into recognizable images, in this application, humanoid sketches of costumes. The discriminator judges images and guesses if they are real or fake. This is used to compete with the generator to lead to more realistic images being created. To further incentivize play, each player is given a cost, and each has a goal to minimize their cost. (Goodfellow et al., 2014)[2].

2.2 Other Methods for Unsupervised Learning

Unsupervised learning is a common problem in computer vision research. GANs are just one type of unsupervised learning with unlabeled data. There are several methods that do not utilize GANs and can still provide successful results in terms of clustering like images. However GANs, and specifically DCGANs are successful at finding important layers that comprise an image and at discerning realistic and unrealistic generated images.

2.3 Modifications to GANs

After the first GAN paper was published, several other researchers have proposed new or different ways to optimize the program for better results. Each generator/discriminator pair uses a mathematical equation to maximize or minimize, respectively, and by slightly modifying this equation, the GAN can possibly improve its results, shorten the training time, or better guarantee a stable training process. There are several loss functions that have been proposed, and each has different strengths and drawbacks. The GAN model that will be tested in this application include the DCGAN, or a Deep Convolutional GAN first detailed by Radford et al., (2015)[8] where convolutional and convolutional transforms are used as layers by the generator and discriminator. The DCGAN model uses incremental convolutions, which allow the model to learn from its downsampling. Downsampling is lowering an image’s pixel density by calculating what pixel color occurs most frequently.
Other modified loss functions and approaches to GANs exist, but the DCGAN is well documented and frequently used, and is successful in producing high quality images. DCGAN’s ability to learn from downsampling is especially applicable to the sketches used in this model due to the outline of the figures in dark ink, which will be used when constructing images, as later building layers will add in this detail. DCGANs have been specifically constructed to ensure relatively stable training, which will be necessary for such a small training set.

3 Data Set Used and Its Significance

The sketches used as the training data set are a collection of Mardi Gras costumes found in the Louisiana Digital Library, found in the Louisiana State Museum Carnival Collection[7]. From the Krewe of Mokana, Venus, Proteus, Bacchus, and more, these sketches date between the 1930s and 1990s, and have been scanned in from their original paper copies. The Louisiana Digital Library (LDL) is an online database containing scans uploaded by member institutions. These scans include images and data concerning a wide variety of artifacts and paper goods from across Louisiana. Recently, the LDL and librarians at LSU have begun an initiative to use the LDL as Data. [9] This initiative encourages scholars to use the collection as a whole, rather than individual images, in order to create useful tools. This generative model aims to be one of those tools.

In order to perform the necessary training of the model, images were manually selected and cropped to square dimensions containing the most information relevant to the costume. Each image was then further resized programmatically to 64x64 pixels. Figures 1, 2, and 3 are sample images used to train the models, in their cropped, square format, but in a higher resolution than the model would see.

4 Technical Implementation

I will be using a Goodfellow inspired DCGAN based on a pytorch tutorial by Nathan Inkawhich (2021)[5], with 64x64 sized image inputs. Aligning with Goodfellow’s research and Inkawhich’s tutorial, “the initial model weights are randomly created from a Normal distribution with mean=0, stdev=0.02.
Figure 1: Costume design for Krewe of Mokana, costume for maid 1, entitled "Amphitrite," 1959. Creators Liuzza, Lucia, and Liuzza, Philip

Figure 2: Costume design for Krewe of Mokana, costume for maid 2, entitled "Aphrodite," 1959. Creators Liuzza, Lucia, and Liuzza, Philip
The weights.init function then takes the initialized model as input and reinitializes all convolutional, convolutional-transpose, and batch normalization layers to meet this weight distribution. This function is then applied to the models immediately after initialization." (Inkawhich, 2021)[5]. The model is trained, with the generator trying to maximize its loss function, defined as $\log(D(G(z)))$, and the discriminator trying to maximize its loss function defined as $\log(D(x)) + \log(1 - D(G(z)))$. These are calculated by a PyTorch function called BCELoss, which is defined as

$$
\ell(x, y) = L = \{l_1, ..., l_N\}^\top, l_n = -[y_n \cdot \log(x_n) + (1 - y_n) \cdot \log(1 - x_n)]
$$

This function then used to calculate the results of the loss functions during and after each round of training.

The pytorch Dataset and Dataloader classes were used to iterate through the cropped images. Each image was transformed into a 3x64x64 tensor. The first model, in figure 4 shown below was created using 86 costume images and 500 epochs of model training. Figure 5 shows a collection of the training images side by side with the model’s generated images. Notice the large amount of noise present in the generated images. Figure 6 shows the loss function graphed over the course of the model’s training. is calculated using The large spikes, especially later in the iterations demonstrate that this model can be further improved, as it is unstable in this current state. The final ratio
of loss between the generator and the discriminator leave the generator much room for improvement.

Figure 4: Generated images in test run 1 based on 86 costume sketches

The win/loss ratio for the generator, which ended at about .8, was still nowhere near .5, which is where a well trained GAN will be, since the Generator will be creating half and the rest will be real. The Discriminator should also approach .5, as that would mean it is identifying half correctly, which means that it is identifying them all as real, and getting the real images
correct, while guessing incorrectly for the generated images. To improve the model, the next step would be to increase the number of epochs for this set of training images. I then trained the model using 2148 epochs, as an increase in training might yield clearer images. The resulting win/loss ratio was still not making significant movements towards .5, as the average win/loss was still around .9 Large spikes of loss as shown in Figure 8 indicate that the generator lost definition, but the accompanying large spikes of Discriminator loss seem to mean that the Discriminator was possibly over-trained on too small of a dataset.

\[ \text{Loss}_{D} = 0.0394 \quad \text{Loss}_{G} = 5.0924 \quad D(x) = 0.9767 \quad D(G(z)) = 0.0153 / 0.0116 \]

The next phase of tuning the GAN is to speed up the processing time, as the 2148 epoch runtime was over 3 hours. Along with speeding up the processing time, in order to get better results, more images were added to the training set to expand the costume library. This increased the range of colors and introduced more parade themes and Krewe styles into the mix. The different eras and figure shapes added even more variation to the set.

Figure 9 and 10 demonstrate that the settings for this training session were not optimal in any way. This increased dataset size led to a significantly longer runtime than previous runs, and the loss functions failed to converge with any meaning. Adjustments were made, and the model was trained again.
The batch size was lowered to speed up the processing. During the following smaller batch size, 500 epoch training session, the images got pretty detailed, and then rapidly degraded around the 1400 iteration mark, indicated by the large spike in generator loss seen in figure 12. The earlier generated images are actually quite good dupes for the costume images, as can be seen in figure 11.

This sudden rise of noisy images is what I think to be a case of convergence failure. Convergence failure is when the model fails to reach an equilibrium where the loss functions for each sit near .5, and in an attempt to restart and get closer, the generator resorts back to noisy images in order to regenerate it’s layers in a better way. This likely means the model was overfitted for the size of the dataset.

Adjusting the epochs to run up to before the previous run began to have significant loss yielded these results found in figure 13, 14 and 15. 15 demonstrates that at the end of the run, there is no convergence failure, but still not an ideal loss ratio. Figure 13 demonstrates part way through the iterations, where the discriminator loss function is still much more variable. Figure 14 is the final output of this run, and the generator has produced humanoid silhouettes, paper colored backgrounds, and a good variety of colors across

Figure 6: Loss function graphed over number of iterations through training set
Figure 7: After increased epochs (over 2000), figures gained more variation, but even more noise, with some barely appearing humanoid.
After yielding this relatively successful outcome, the next step is to tune the hyperparameters to see if a better result can be obtained from this model. The learning rate was manipulated, but yielded unsuccessful models, as seen in figure 16. This slight adjustment, although a learning rate of .0001 was suggested for the original GAN model, completely broke this version.

5 Analysis

The DCGAN led to images containing humanoid figures full of color with a stable background. With the varied themes of costumes found in the database, it is not surprising to see such a wide variety of shapes and colors. The relatively low amount of training info is likely the cause for a lack of better distinction within the images, as many training sets used in other DCGAN applications have several thousand images to train from. Manipulating the training rate negatively affected the model, causing irreversible convergence failure of both generator and discriminator loss functions.
Figure 9: Results after increasing dataset size
Figure 10: Loss with less iterations but larger batch size

Figure 11: Images generated over time during 1400 iteration training
Future Plans

Further experimentation within the hyperparameters of the two GAN models used in this paper may yield better results. The focus of this experiment included the learning rate and the epochs needed to train a model on a smaller training set of real images.

Further experimentation with more variations of GANs could be implemented to find the true optimum solution. Examples include a minimax based GAN, an EGAN, a WGAN or a non-saturating GAN. Different loss functions or modified model building might yield better results for this test case.

Distributed computing with higher resolution images could both increase accuracy and reduce training time for the models.

Finding more costumes to reference would allow the models to grow their reference data. More images from other various databases or Krewes could lead to a better dataset to train from and yield more realistic images. Most applications of GANs use databases with thousands of images, while this dataset was only several hundred images large.
Figure 13: Early generated images in final default run
Figure 14: Final generated output of default run
List of Figures

1  Costume design for Krewe of Mokana, costume for maid 1, entitled "Amphitrite", 1959. Creators Liuzza, Lucia, and Liuzza, Philip ................................................................. 4
3  Costume design for Krewe of Mokana, costume for krewe maskers, entitled "Athletes," 1959, creators Liuzza, Lucia, and Liuzza, Philip ................................................................. 5
4  Generated images in test run 1 based on 86 costume sketches ........................................ 6
5  Side by side comparison of real images and fake images. Notice the noise in the generated images ................................................................. 7
6  Loss function graphed over number of iterations through training set ........................................ 8
7  After increased epochs (over 2000), figures gained more variation, but even more noise, with some barely appearing humanoid ........................................ 9
8  Loss over 2000 iterations on a small data set ................................................................. 10
9  Results after increasing dataset size ........................................................................ 11

Figure 15: Graph depicting loss functions for final default parameter training
Figure 16: Final generated images for slower learning rate
References


