American University of Sharjah
Sharjah, United Arab Emirates

COE476 - Neural Networks

Artwork Generation Using Conditional GANs

Submitted to Dr. Imran Zualkernan

TABLE OF CONTENTS

I. Introduction

Humans have been creating visual art since the dawn of our species, beginning with powerful depictions of game animals found in caves ranging from France to Indonesia. Paintings constitute an important form in visual arts, bringing in elements of such as drawing, composition, gesture, narration, and abstraction [1]. The oldest known paintings are approximately 40,000 years old. Over time, painting has captured the interest of millions of people and has been transformed from a skill to a hobby and an occupation. A series of art movements in the late 19th and early 20th centuries led to the introduction of several painting styles such as Impressionism, Post-Impressionism, Fauvism, Expressionism, Cubism, etc., all of which have been adopted by renowned artists [2]. However, innovation in technology and the modernization of society have revolutionized the idea of paintings. These revolutions resulted in the development of digital and algorithmic arts. Algorithmic art, also known as computer-generated art, is visual art in which the design is generated by an algorithm. Algorithmic artists are sometimes called algorists. Roman Verostko argues that Islamic geometric patterns are constructed using algorithms, as are Italian Renaissance paintings which make use of mathematical techniques, in particular linear perspective and proportion. Some of the earliest known examples of computer-generated algorithmic art were created by Georg Nees, Frieder Nake, A. Michael Noll, Manfred Mohr, and Vera Molnár in the early 1960s. These artworks were executed by a plotter controlled by a computer, and were, therefore, computer-generated art but not digital art. From one point of view, for a work of art to be considered algorithmic art, its creation must include a process based on an algorithm devised by the artist. Here, an algorithm is simply a detailed recipe for the design and possibly execution of an artwork, which may include computer code, functions, expressions, or other input which ultimately determines the form the art will take [3]. This input may be mathematical, computational, or generative in nature. The main challenge for digital artists and algorists is to utilize digital tools to create memorable images that genuinely touch the viewer's feelings and intrigue their intellect. As a result, visual art is always evolving. For example, there was a time when oil easel painting was a new, transformational medium. In the coming years, we'll be unleashing the artistic potential of Virtual Reality and Artificial Intelligence (AI). For a few decades, AI enthusiasts have been trying to integrate human creativity into computer intelligence and the potential of generating creative and new art is considerably exciting.

There are several algorithms and learning methods that can be employed to generate paintings. The challenge is to optimize creativity at a level that replicates the work of a human being. Furthermore, for an image of reasonable size, even the simplest algorithms require too much calculation for manual execution to be practical, and they are thus executed on either a single computer or on a cluster of computers.

In this project, our team has attempted to venture into the possibilities of generating creative art, i.e., content generation, using neural networks. A neural network is an interconnected group of artificial neurons that are organized in the form of hidden layers that use a mathematical or computational model to determine the relationships between the inputs that it is served. We will be feeding the neural network a dataset of different genres of paintings by different artists and run it through a generator and discriminator which will generate paintings of a specific style as per our discretion and then accurately classify the painting based on the whether it was a real image or a fake made by the generator. The neural network has limitations in terms of the type and content of the dataset that is taken as input, i.e, it needs to be served with a well-refined dataset of paintings that are significantly different in genres and styles for the network to accurately generate and classify them. Furthermore, since art is subjective and different art styles may overlap, the classification and generation of computational art is a challenging task. However, the brilliant synergy between art, math, science, and technology has opened a spectrum of possibilities and perhaps we will soon be able to generate art that is original and creatively inspiring.

II. Previous approaches

Art generation through AI has been on the rise in popularity in recent years and has come quite a long way since the idea was first proposed by Ada Lovelace [4]. Lovelace came up with the concept of “poetical science” in 1842, where she speculated that computers would eventually be able to compose music; however, the term poetical science has a grander meaning, and it referred to an algorithm's capability to generate art. Ada Lovelace’s theory inspired Harold Cohen to create AARON in 1973, which is regarded as the first AI capable of generating art (as seen in figure 1)[4][5]. AARRON was programmed to paint specific objects with the irregularity of freehand drawing, but the programming later displayed results that Cohen himself had not imagined before; this unexpected output seemed almost like an artistic decision.

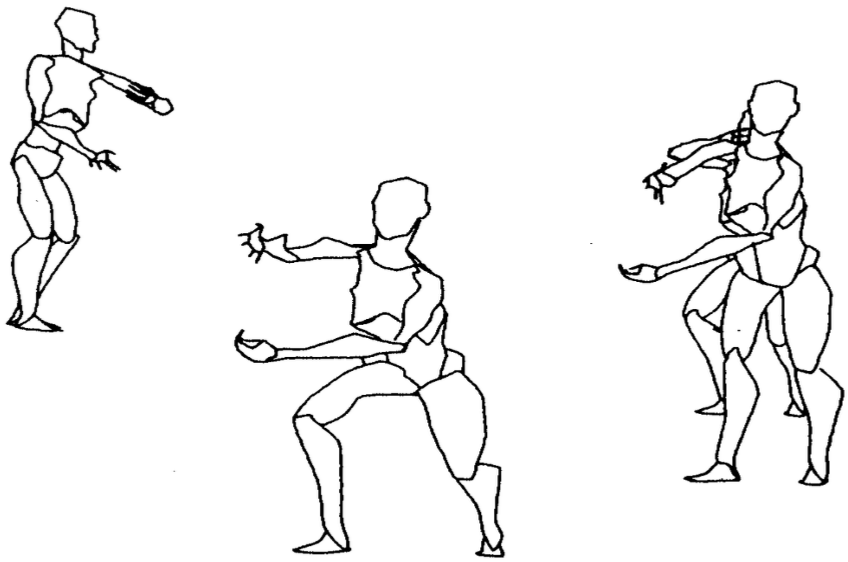


Figure 1: Image generated by AARON [4]

Since then, multiple more attempts have been made at art generation using artificial intelligence. And these attempts lead us to state-of-the-art Generative Adversarial Networks (GANs). GANs operate on the principle of an adversarial game played between a discriminator neural network and a generator neural network. The discriminator is trained on a set of data, such as painting for artist Z, and learns how to teel the difference between real paintings of painter Z and fakes; the generator, however, will attempt to generate those fakes and fool the discriminator. GANs display a tremendous ability to generate new artworks based on a specific style. Upon supplying a GAN with a dataset of Van Gogh paintings, as in the prior example, it should be able to create an image that is very similar to Van Gogh's previous works. Not only that but if one feeds a set of multiple artists' works to a GAN, the generated images will contain a form of Eclecticism.

Eclecticism in the fine arts refers to combining multiple styles or borrowing techniques from prominent artists to apply to a single artwork. The term eclectism was first used by a group of greek and roman philosophers that based their philosophical beliefs on a selection of doctrines from existing beliefs [6]. However, the term later found its way into the visual arts, where it was used to describe the works of a famous family of painters from the 1500s called the Carracci; whose artworks combine Michelangelo's lines, Titian's colors, Correggio's use of strong contrasts, and Raphael's symmetry and grace [6].

An example of this particular method is the French Collective Obvious' *"Portrait de Edmond de Belamy"* [5], which was a GAN-generated artwork where the GAN was trained on 15,000 portraits from the 14th to 20th century. The portrait sold for $432,000 and is regarded as a large milestone for AI-generated art.

Figure 2: Portrait of Edmond de Belamy [5]

Following the topic of GANs, we introduce the concept of Conditional GANS (cGANs). cGANs work very similarly to normal GANs with the exception that the generator and the discriminator are conditioned to use auxiliary information; cGANs generate an output based on "class-conditional distributions" [7]. This means that we can add an extra input in the form of a one-hot encoded vector of image/painting labels (in our case the labels are the painters in question), and this vector can guide the generator in terms of what image to produce. cGANs can help us direct the prior mentioned concept of eclecticism, as we can use this auxiliary information vector to favor the influence of one painter or a style or an era of paintings over the other. cGANs are the approach taken for this project.

i. Literature Review

Reference	Features	Advantages	Disadvantages
[8] S.-A. Chen, C.-L. Li, and H.-T. Lin, "A unified view of cGANs with and without classifiers," *arXiv [cs.CV]*, 2021.	Proposes ECGAN, which justifies the use of a classifier alongside the generator and the discriminator in a conditional GAN	Adding a classifier improves the performance of both the generator and discriminator and helps in learning the join distribution	Requires a large computational resource
[9] T. Miyato and M. Koyama, *Arxiv.org*, 2018. [Online]. Available: https://arxiv.org/pdf/1802.05637.pdf. [Accessed: 09-May-2022].	State-of-the-art cGAN that does not use a classifier. Presents a projection-based way to incorporate the auxiliary/conditional information to the cGAN	It is a much better method than arbitrarily attaching the auxiliary information to the input matrix, and presents a logical basis for the addition	Further improving the quality of generated images requires considerable time. And this type of cGAN does not preform very well with multimodal data
[10] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," *arXiv [stat.ML]*, pp. 2642–2651, 06--11 Aug 2017.	Proposes the idea of AC-GANs, a generative adversarial network that uses class conditioning to generate images	High quality images can be generated much better than the average GAN, giving us some control over the style through the label data inserted.	The auxiliary classifier might encourage the generator to produce images that are easy for the auxiliary classifier to classify and deviate from the desired data distribution

[11] H. Chen *et al.*, "Creative and diverse artwork generation using adversarial networks," *IET Comput. Vis.*, vol. 14, no. 8, pp. 650–657, 2020.	Allows us not only to generate new images based on a dataset but also to extract features from an image and apply its style to the generated images. However, instead of using a GAN to generate the images and then apply the style transfer separately (two separate operations and two separate models to train), this paper combines the two into one model by using a Gram matrix to constrain the style of the generated images.	Saves time that would be used to train two models if one wanted to do a style transfer GAN. In addition, the implementation of 2 discriminators improves the quality of generated images	Requires a very large dataset in order to produce favorable results
[12] W. R. Tan, C. S. Chan, H. E. Aguirre, and K. Tanaka, "ArtGAN: Artwork synthesis with conditional categorical GANs," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3760–3764.	This paper proposes a cGAN called ArtGAN that not only adds the one-hot encoded conditional information to the generator input but also relays the same conditional information to the discriminator. In other words, when the conditional information is given to condition a generated image to one class, it is passed alongside the image to the discriminator and the error is propagated back to the generator.	Improves the quality of generated artwork, it is also much better at learning more abstract concepts/features	Paper only provides one comparison metric, and the improvement is not substantial
	This paper introduces a third component to the adversarial game played	Greatly improves classification results by training it	This paper is one of two or three papers discussing this

[13] S. Vandenhende, B. De Brabandere, D. Neven, and L. Van Gool, "A Three-player GAN: Generating hard samples to improve classification networks," *arXiv [cs.CV]*, 2019.	between the generator and the discriminator. A classifier is added to the GAN where the goal is for the generator to not only learn the distribution of actual data belonging to the classes but to also learn the distribution of the hard to classify data.	on synthesized hard-to-classify samples	approach, so there is not enough data or experiments to go on
[14] D. Yang, S. Hong, Y. Jang, T. Zhao, and H. Lee, "Diversity-sensitive conditional generative adversarial networks," *arXiv [cs.LG]*, 2019.	The paper proposes a simple regularization function on the generator that directly penalizes model-collapsing behavior, thus helping avoid model collapse.	Adresses the model-collapse problem with a simple function	Causes an imbalance between realism and diversity in generated images
[15] M. Hu, D. Zhou, and Y. He, "Variational conditional GAN for fine-grained controllable image generation,"	This paper discusses a variational generator framework for conditional GANs to "catch semantic details for improving the generation quality and diversity."[15] This is done by introducing a variational inference	The framework proposed is flexible and easily applicable to GANs. Overcomes the problem of the blurry image generation problem of its predecessor CVAE-GAN	-

arXiv [cs.CV], 2019.	into the generator.		
[16] M. Mirza and S. Osindero, "Conditional generative Adversarial Nets," *Arxiv.org*. [Online]. Available: https://arxiv.org/pdf/1411.1784.pdf. [Accessed: 09-May-2022].	Introduces the Conditional Adversarial Networks that is an extension of GANs to a conditional model by conditioning the generator and discriminator to some extra auxiliary information such as labels or data from other modalities.	The controlled content-generation helps to logically visualize and understand the performance of the GAN. Furthermore, a multi-model was employed for tagging images and pretrained on the full ImageNet dataset thereby indicating robust training.	The hyper-parameters and architectural choices were based random grid search and manual selection which limited the search space. Additionally, model performance analysis was not done thoroughly.
[17] A. Elgammal, B. Liu, M. Elhoseiny, and M. Mazzone, "CAN: Creative adversarial networks, generating 'art' by learning about styles and deviating from style norms," *arXiv [cs.AI]*, 2017.	The system generates art by looking at art and learning about style; and incorporates creativity by generating art that is deviant from all art styles but is still within the spectrum of being classified as art. Simply put, the model attempts to generate art that is novel but not too novel (that it exceeds the domain of art).	Incorporates style classification loss and a style ambiguity loss to achieve optimal arousal. Generated aesthetically pleasing GAN paintings.	The evaluation metric used was a survey done by humans in distinguishing original paintings from GANs.

[18] A. Xue, "End-to-end Chinese landscape painting creation using generative adversarial networks," *arXiv [cs.CV]*, 2020.	Proposed Sketch-And-Paint GAN (SAPGAN), the first model which generates Chinese landscape paintings from end to end, without conditional input.	Disentangles content generation from style generation into two distinct networks. Data processing and models are likely generalizable to any dataset encodable via HED edge detection. 512x512 HED edge maps are generated and concatenated with dataset images in preparation for training to counteract Canny edge detection's loss of important high-level edges as well as production of disconnected low-level edges.	PaintGAN candidates do poorly at the granular level needed to fill in Chinese calligraphy, producing blurry characters. Model is limited to generating landscapes. Hence, its performance is based on a very domain.
[19] H. Kazemi, S. M. Iranmanesh, and N. M. Nasrabadi, "Style and content disentanglement in generative adversarial networks," *arXiv [cs.CV]*, 2018.	This paper describes the Style and Content Disentangled GAN (SC-GAN), a new unsupervised algorithm for training GANs that learns disentangled style and content representations of the data.	UsesInstance Normalization (IN) carry the style information of an image. Residual blocks are equipped with AdaIN layers whose parameters are dynamically generated by the MLP block from a style code. Proposed disentanglement of style and content representations comes at least with no cost in terms of the quality of synthesized images.	Without the diversity loss, the model suffers from partial mode collapse, in which many style codes render the same texture on the output images. Limitation of same quality of images to be compared between SC-GAN and LSGAN.

[20] A. Radford, L. Metz, and S. Chintala, “Under review as a conference paper at ICLR 2016 UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS,” *Arxiv.org*. [Online]. Available: https://arxiv.org/pdf/1511.06434.pdf. [Accessed: 09-May-2022].	Introduce deep convolutional generative adversarial networks (DCGANs), that have certain architectural constraints, and demonstrate that they are a strong candidate for unsupervised learning.	Directly connecting the highest convolutional features to the input and output respectively of the generator and discriminator worked well. To further decrease the likelihood of the generator memorizing input examples, simple image de-duplication process was performed.	No data augmentation was applied to the images. As the models were trained longer they sometimes collapse a subset of filters to a single oscillating mode. Requires a very large dataset and large training time to generate one type of image.
[21] *Researchgate.net*. [Online]. Available: https://www.researchgate.net/publication/343952095_RPD-GAN_Learning_to_Draw_Realisti	Propose RPD-GAN (Re-alistic Painting Drawing Generative Adversarial Network), anunsupervised cross-domain image translation framework forrealistic painting style transfer.	Embed a convolutional blockcomprising cascaded Convolution –AdaIN –Relu layersat the middle of the residual blocks. The framework enables flexible	Partial missing of content features in the generated results

c_Paintings_with_Generative_Adversarial_Network. [Accessed: 09-May-2022].		control of content-style tradeoff bytuning the weights of the content-consistency loss Lcontent andstyle-alignment adversarial loss Lstyle−adv , which strengthenmodel's content preservation and style capturing ability respectively.	
[22] Implementation of Art Pictures Style Conversion with GAN, *Techscience.com*. [Online]. Available: https://www.techscience.com/jqc/v3n4/46223. [Accessed: 09-May-2022].	Analyzes the performance of the CycleGan. Proposes the concept of learning the mapping relationship and inverse mapping relationship between the source domain and the target domain which can reduce the mapping and improve the quality of the generated image.	Through the idea of loop, the loss of information in image style conversion is reduced. Only changes the texture and line of the image to reach the level similar to the artist's style.	The loss curve of the generator first decreases, then stabilizes, and then rises. Requires a massive 50K–100K epochs to produce optimal results.
[23] A. Mufti, B. Antonelli, and J. Monello, "Conditional GANs for painting	Implemented Spectral Normalization GAN (SN-GAN) and Spectral Normalization GAN with Gradient Penalty.	Normalizes the spectral norm of the weight matrix W at each layer in the discriminator. The gradient penalty enforces 1-Lipschitz continuity by adding a	Only 100 images are generated and compared to 100 randomly selected images from the training data. The model does not perform well on a dataset of faces.

generation," in *Twelfth International Conference on Machine Vision (ICMV 2019)*, 2020.		regularizing term to the cost function of the GAN, rather than altering the weights. Uses the Sliced Wasserstein Distance (SWD) as a quantitative evaluation measure that effectively compares training and generated images in both appearance and variation at different resolutions.	
[24] X. Xie and B. Lv, "Design of painting art style rendering system based on convolutional Neural Network," *Sci. Program.*, vol. 2021, pp. 1–11, 2021.	This paper uses the improved cycle generative adversarial network (CycleGAN) to render the current image style.	Replaces the deep residual network (ResNet) of the original network generator with a dense connected convolutional network (DenseNet) and uses the perceptual loss function for adversarial training. The painting art style rendering system built in this paper is based on perceptual adversarial network (PAN) for the improved CycleGAN that suppresses the limitation of the network model on paired samples. Improves the quality of the image generated by the artistic style of painting and further improves the stability	Although the proposed model has diminishing value of loss, it is still relatively high. The perceptual adversarial network can more effectively minimize the loss function.

		and speeds up the network convergence speed.	
[25] L. Zhang, Y. Ji, X. Lin, and C. Liu, "Style transfer for Anime sketches with enhanced residual U-net and auxiliary classifier GAN," in *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, 2017, pp. 506–511.	Integrated residual U-net to apply the style to the gray-scale sketch with auxiliary classifier generative adversarial network (AC-GAN).	Networks are fine-tuned by shifting two loss functions. Applies global style hint to U-net	The batch size during training is limited to 4.
[26] R. Nakano, "Neural Painters: A learned differentiable constraint for generating brushstroke paintings," *arXiv [cs.CV]*, 2019.	Propose a method for encouraging an agent to "paint" images by following human-like strokes when reconstructing digits.	Presents the concept of intrinsic style transfer, i.e., by minimizing only the content loss from neural style transfer, the artistic medium, in this case, brushstrokes, are allowed to naturally dictate the resulting style	The images produced are of very low quality.
	Collected a large set of both sketch data and Chinese Shanshui painting data to train	Designed an interactive system to generate the Chinese Shanshui painting	The output is highly reliant on the information provided by the user, such as

[27] L. Zhou, Q.-F. Wang, K. Huang, and C.-H. Lo, “An interactive and generative approach for Chinese shanshui painting document,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 819–824.	the model of cycle-GAN Developed an interactive system called Shanshui-DaDA (i.e., Design and Draw with AI) to generate Chinese Shanshui painting documents in real-time.	documents, where the users only need to input the sketch simply. Cycle consistency loss Lcyc(G,F) is added as a regularization term to prevent the learned mappings G and F from contradicting each other. The proposed system generates new types of Chinese Shanshui paintings instead of simulating the existed paintings only.	number of lines for landscapes. Since the domain of Shanshui paintings is still in its infancy of research, the results are unreliable as they are limited to the optimized dataset used.

Figure 3: Literature Review

III. Data selection

In order to select the right dataset, we discussed what type of eclecticism we wanted to aim for. The options were either merging different art eras, different styles within a single artist's works, or even different styles belonging to different painters. We attempted to train our cGAN on a set of Van Gogh paintings categorized by style, but the dataset was too small and the styles too similar within a single artist. So we look for a dataset of different artists but found none that were large enough for valid results. So, we resulted to combining three different datasets, each dedicated to a painter, to form our final dataset. Our dataset consists of three different artists: Monet, Picasso, and Van Gogh; and the dataset comes up to around 6 thousand images.

IV. Data cleaning and Feature Engineering

After creating our dataset we had a little less than 9000 images from 3 different artists. However, each image or set of images would come in different sizes. To keep things consistent and working we used a resizing function to transform all images to the shape 128x128x3. We also made sure that the images were rescaled and augmented to provide for stable training. Each image had its pixel values placed within the range [0,1]. We also had random height and width shifting to help prevent overfitting and generator collapse.

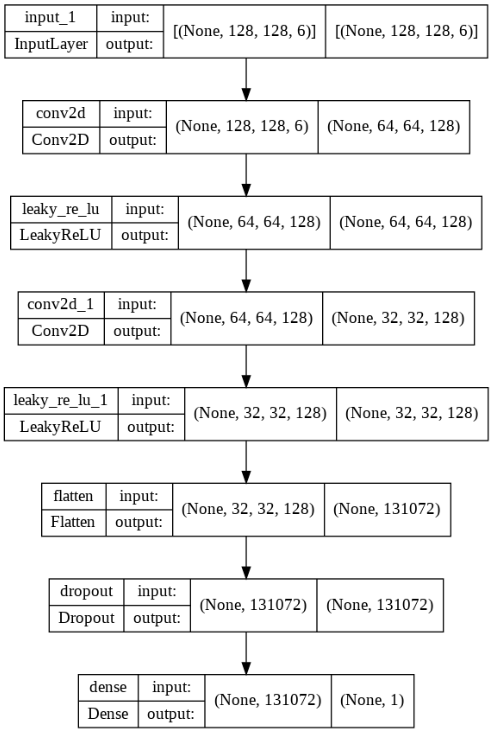
For cGANs, we used the classes (the artists) as supplemental input into the generator. This was in the hopes that the generator would be able to generate paintings by a certain artist when given their respective label.

V. Architecture Selection

We have drafted 3 models, all conditional GANs, to be trained on the Van Gogh-Monet-Piccasso dataset:

1. Model A

   We decided that for our first model we would attempt very simple generator and discriminator architectures, consisting of 2D convolutions dense layers. This model was inspired by [28], with minor modifications to the generator layers. This model was intended as a baseline.



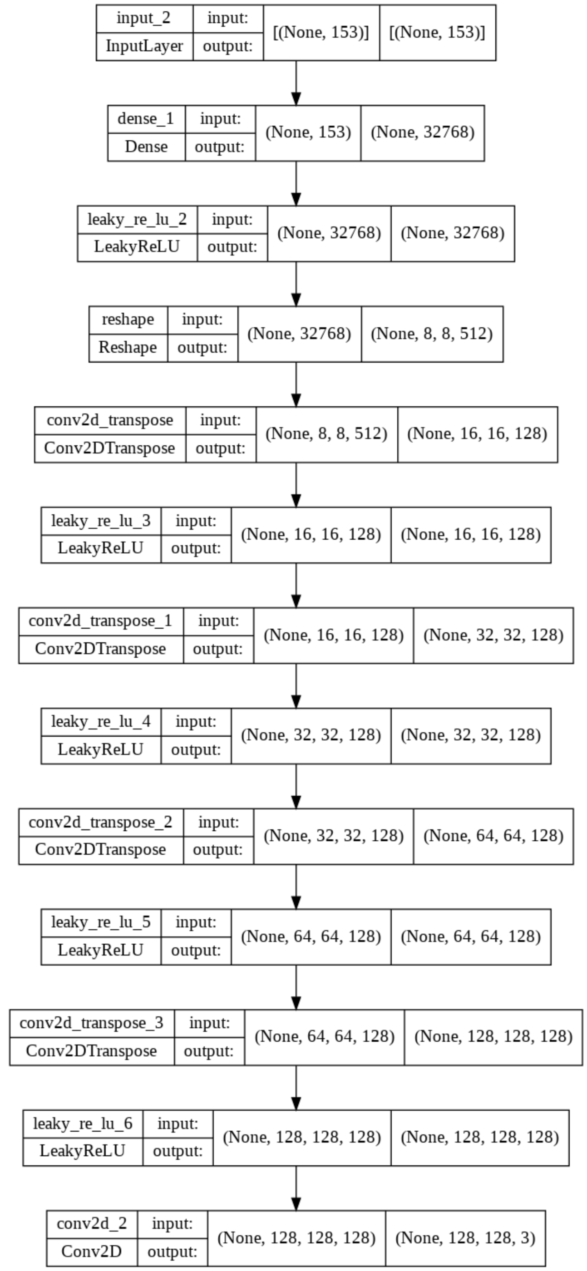


Figure 5: Model A Discriminator

Figure 6: Model A Generator

2. Model B

This model takes its architecture after ArtGan [12] which is described earlier in the peer review. This architecture contains leakyRelU layers, dense layers, and 2D convolutions similar to model A, with the added batch normalization layers and dropout layers. Not to mention Model C is significantly larger than previous model.

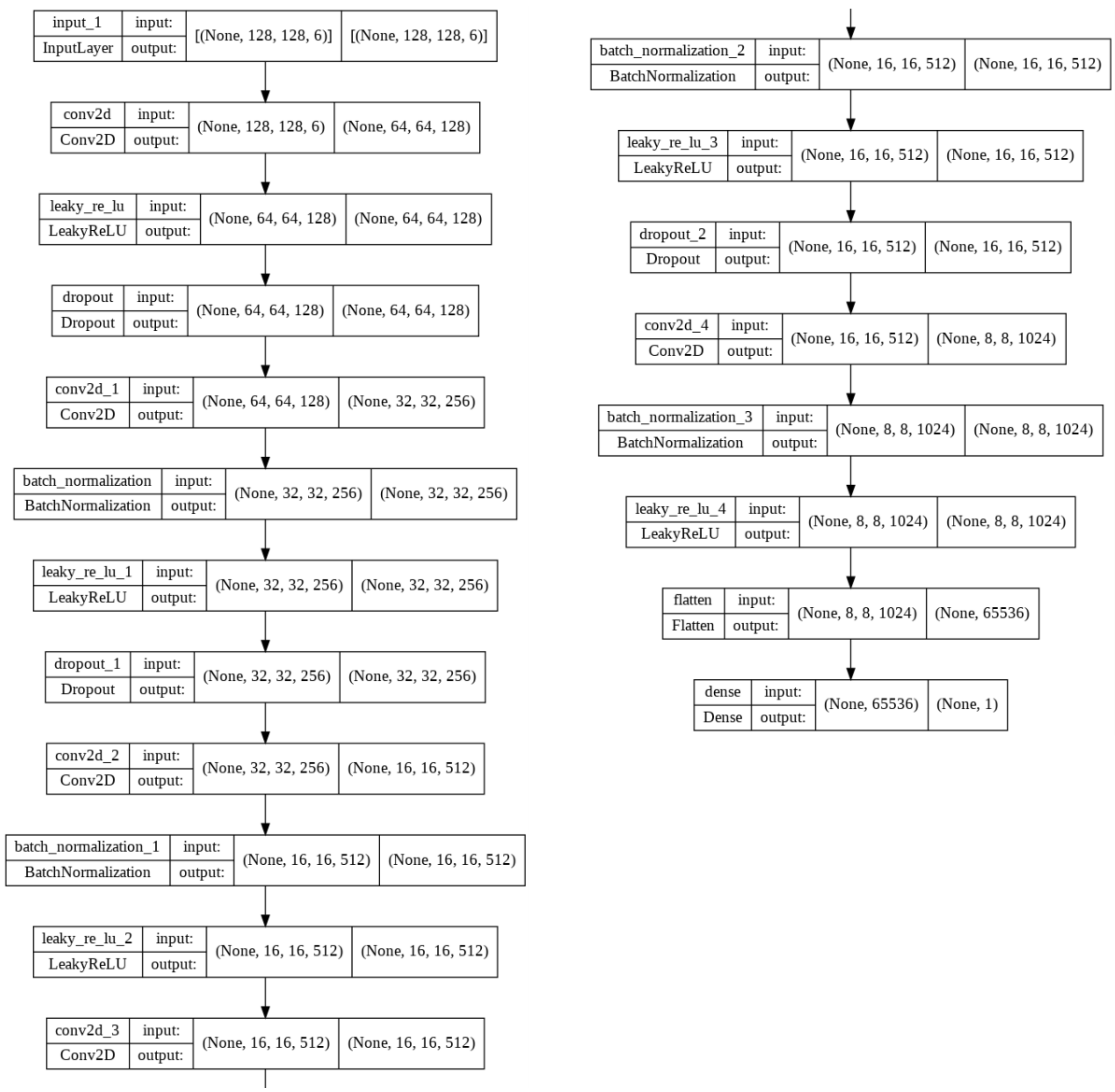


Figure 7: Model B Discriminator

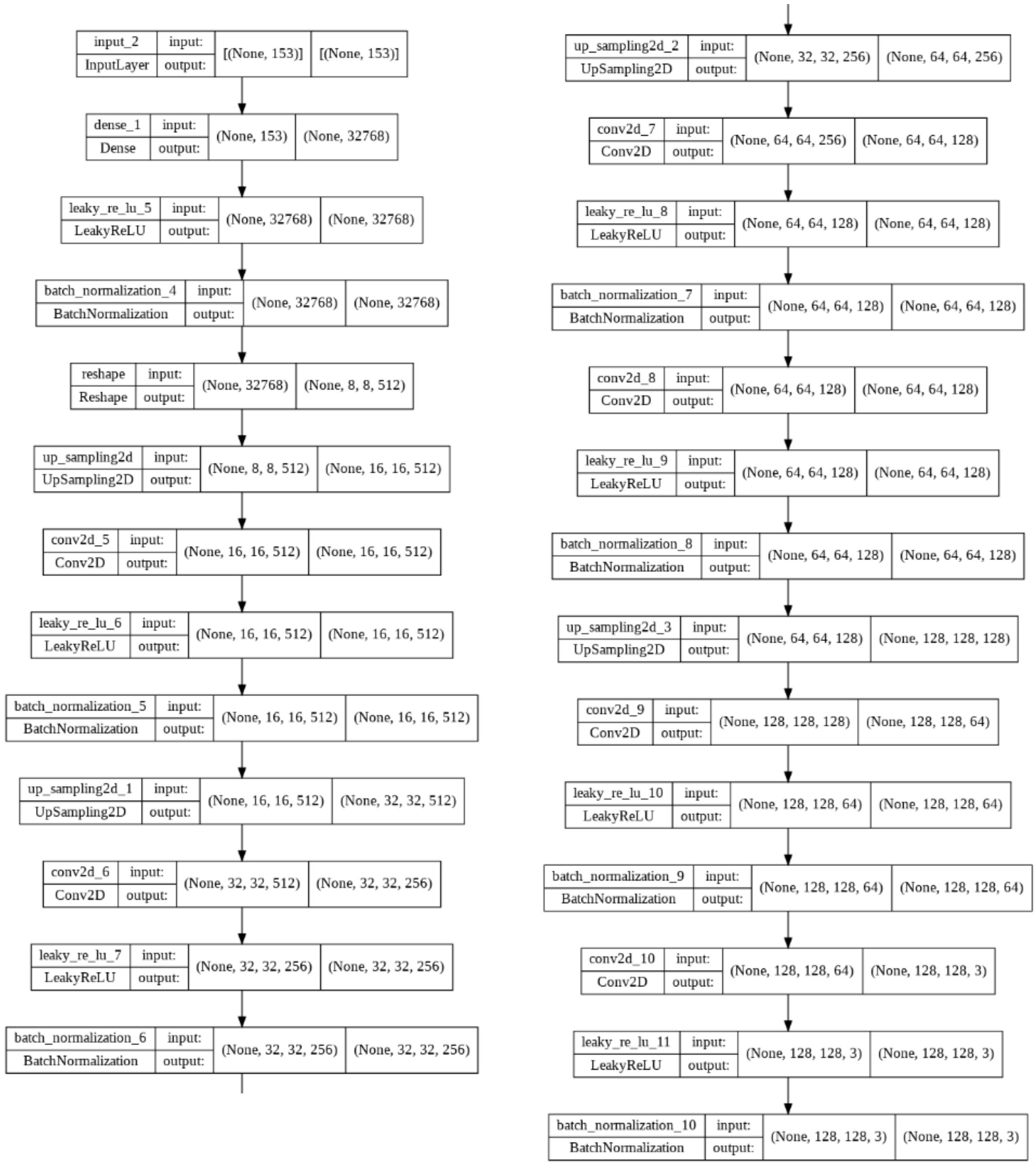

input_2 InputLayer input: output: [(None, 153)] [(None, 153)]
dense_1 Dense input: output: (None, 153) (None, 32768)
leaky_re_lu_5 LeakyReLU input: output: (None, 32768) (None, 32768)
batch_normalization_4 BatchNormalization input: output: (None, 32768) (None, 32768)
reshape Reshape input: output: (None, 32768) (None, 8, 8, 512)
up_sampling2d UpSampling2D input: output: (None, 8, 8, 512) (None, 16, 16, 512)
conv2d_5 Conv2D input: output: (None, 16, 16, 512) (None, 16, 16, 512)
leaky_re_lu_6 LeakyReLU input: output: (None, 16, 16, 512) (None, 16, 16, 512)
batch_normalization_5 BatchNormalization input: output: (None, 16, 16, 512) (None, 16, 16, 512)
up_sampling2d_1 UpSampling2D input: output: (None, 16, 16, 512) (None, 32, 32, 512)
conv2d_6 Conv2D input: output: (None, 32, 32, 512) (None, 32, 32, 256)
leaky_re_lu_7 LeakyReLU input: output: (None, 32, 32, 256) (None, 32, 32, 256)
batch_normalization_6 BatchNormalization input: output: (None, 32, 32, 256) (None, 32, 32, 256)
up_sampling2d_2 UpSampling2D input: output: (None, 32, 32, 256) (None, 64, 64, 256)
conv2d_7 Conv2D input: output: (None, 64, 64, 256) (None, 64, 64, 128)
leaky_re_lu_8 LeakyReLU input: output: (None, 64, 64, 128) (None, 64, 64, 128)
batch_normalization_7 BatchNormalization input: output: (None, 64, 64, 128) (None, 64, 64, 128)
conv2d_8 Conv2D input: output: (None, 64, 64, 128) (None, 64, 64, 128)
leaky_re_lu_9 LeakyReLU input: output: (None, 64, 64, 128) (None, 64, 64, 128)
batch_normalization_8 BatchNormalization input: output: (None, 64, 64, 128) (None, 64, 64, 128)
up_sampling2d_3 UpSampling2D input: output: (None, 64, 64, 128) (None, 128, 128, 128)
conv2d_9 Conv2D input: output: (None, 128, 128, 128) (None, 128, 128, 64)
leaky_re_lu_10 LeakyReLU input: output: (None, 128, 128, 64) (None, 128, 128, 64)
batch_normalization_9 BatchNormalization input: output: (None, 128, 128, 64) (None, 128, 128, 64)
conv2d_10 Conv2D input: output: (None, 128, 128, 64) (None, 128, 128, 3)
leaky_re_lu_11 LeakyReLU input: output: (None, 128, 128, 3) (None, 128, 128, 3)
batch_normalization_10 BatchNormalization input: output: (None, 128, 128, 3) (None, 128, 128, 3)

Figure 8: Model B Generator

3. Model C

The third model is a modification of the first model based on the architecture used in Mirza and Osindero’s Conditional GAN wherein the hidden layers comprised of Rectified Linear Unit (ReLu) activation, which rendered promising results [16]. Accordingly, the LeakyReLU layers in the discriminator and generator of Model A were changed to ReLU layers.

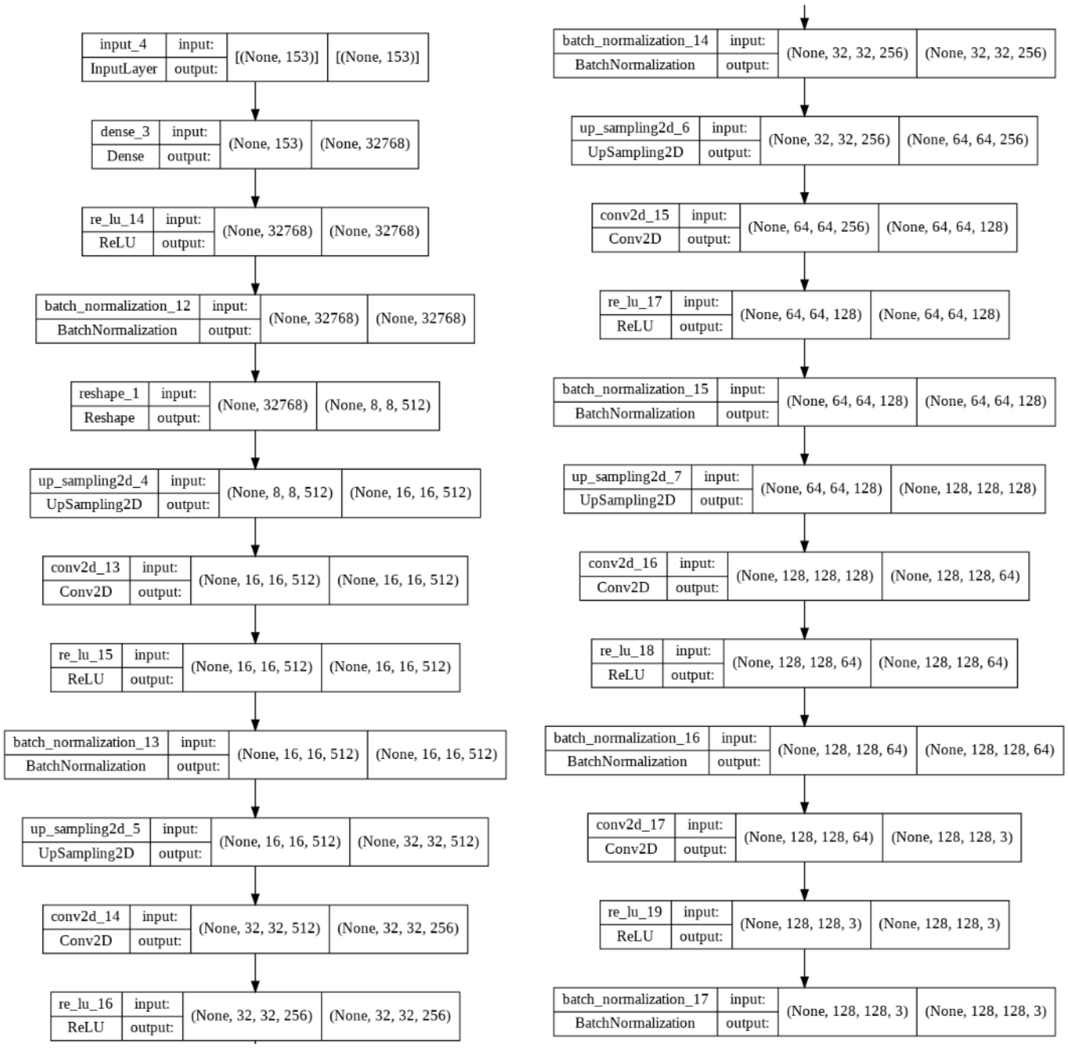

input_4 | input: | [(None, 153)] | [(None, 153)]
InputLayer | output:
dense_3 | input: | (None, 153) | (None, 32768)
Dense | output:
re_lu_14 | input: | (None, 32768) | (None, 32768)
ReLU | output:
batch_normalization_12 | input: | (None, 32768) | (None, 32768)
BatchNormalization | output:
reshape_1 | input: | (None, 32768) | (None, 8, 8, 512)
Reshape | output:
up_sampling2d_4 | input: | (None, 8, 8, 512) | (None, 16, 16, 512)
UpSampling2D | output:
conv2d_13 | input: | (None, 16, 16, 512) | (None, 16, 16, 512)
Conv2D | output:
re_lu_15 | input: | (None, 16, 16, 512) | (None, 16, 16, 512)
ReLU | output:
batch_normalization_13 | input: | (None, 16, 16, 512) | (None, 16, 16, 512)
BatchNormalization | output:
up_sampling2d_5 | input: | (None, 16, 16, 512) | (None, 32, 32, 512)
UpSampling2D | output:
conv2d_14 | input: | (None, 32, 32, 512) | (None, 32, 32, 256)
Conv2D | output:
re_lu_16 | input: | (None, 32, 32, 256) | (None, 32, 32, 256)
ReLU | output:
batch_normalization_14 | input: | (None, 32, 32, 256) | (None, 32, 32, 256)
BatchNormalization | output:
up_sampling2d_6 | input: | (None, 32, 32, 256) | (None, 64, 64, 256)
UpSampling2D | output:
conv2d_15 | input: | (None, 64, 64, 256) | (None, 64, 64, 128)
Conv2D | output:
re_lu_17 | input: | (None, 64, 64, 128) | (None, 64, 64, 128)
ReLU | output:
batch_normalization_15 | input: | (None, 64, 64, 128) | (None, 64, 64, 128)
BatchNormalization | output:
up_sampling2d_7 | input: | (None, 64, 64, 128) | (None, 128, 128, 128)
UpSampling2D | output:
conv2d_16 | input: | (None, 128, 128, 128) | (None, 128, 128, 64)
Conv2D | output:
re_lu_18 | input: | (None, 128, 128, 64) | (None, 128, 128, 64)
ReLU | output:
batch_normalization_16 | input: | (None, 128, 128, 64) | (None, 128, 128, 64)
BatchNormalization | output:
conv2d_17 | input: | (None, 128, 128, 64) | (None, 128, 128, 3)
Conv2D | output:
re_lu_19 | input: | (None, 128, 128, 3) | (None, 128, 128, 3)
ReLU | output:
batch_normalization_17 | input: | (None, 128, 128, 3) | (None, 128, 128, 3)
BatchNormalization | output:

Figure 9: Model C Discriminator

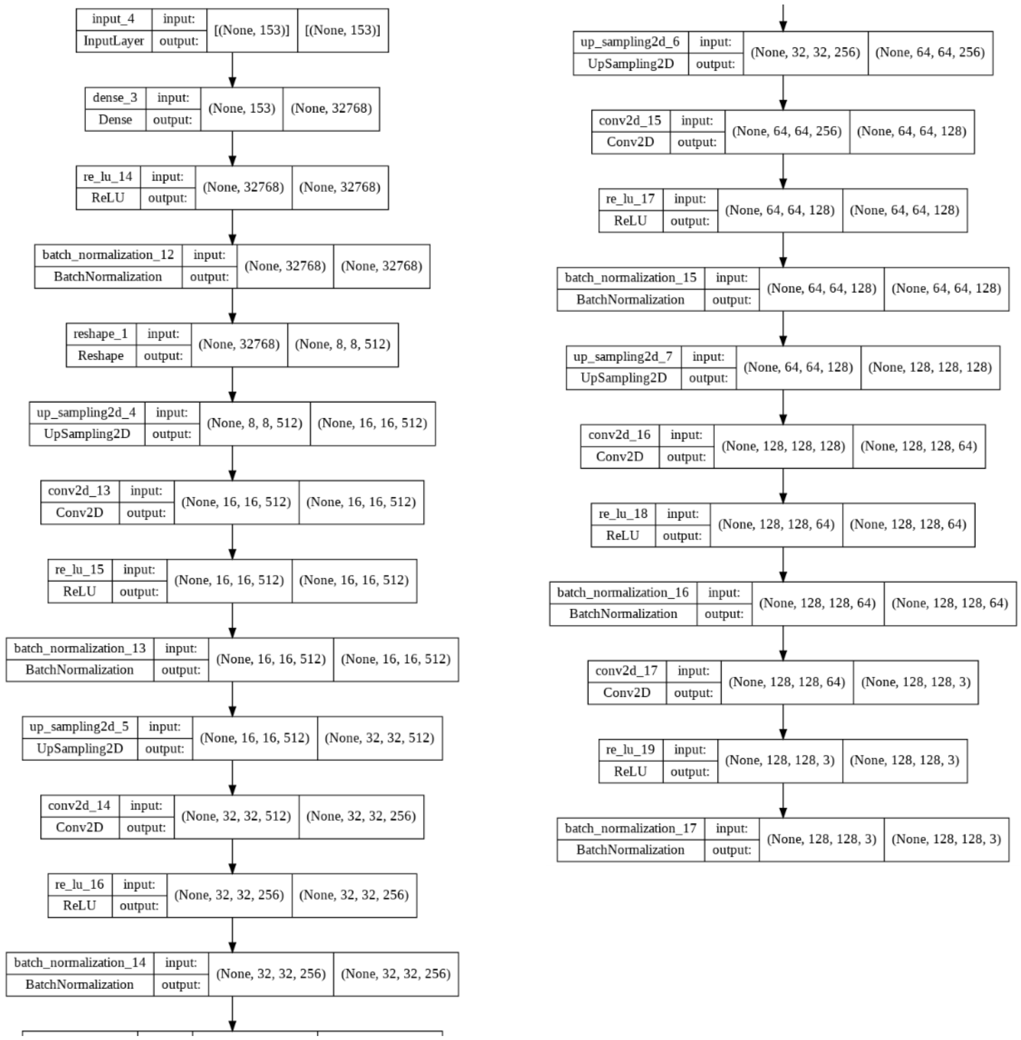


Figure 10: Model C Generator

i. Why do the chosen architectures work?

Model A	This model was intended as a baseline. The architecture is fairly simple without any complex concepts. It is intended to give an idea of cGANs and serve as a basis for models B and C.
Model B	The input layer was altered to work based on the data processing and data cleaning performed on the Best-Artworks-of-All-Time Dataset. The architecture for this model was inspired from the ARTGAN model proposed by W. R. Tan, C. S. Chan, H. E. Aguirre, and K. Tanaka. Hence, the combination of convolutional and leaky relu layers works because it has been tested on the above mentioned model. Additionally, it follows the architecture of general GANs to produce good results.
Model C	Model C is an altered version of Model B with a replacement of LeakyReLU layers with ReLu inspired by the cGAN model proposed by Mirza and Osindero. The model was made simpler by removing one set Conv2d, ReLU and BatchNormalization layers since based on our research, certain simpler models performed better than ones with too many hidden layers. This is because after a certain threshold, the images produced bythe generator tend to look identical as they can successfully fool the discriminator. Hence it works, but with different results due to the activation function employed.

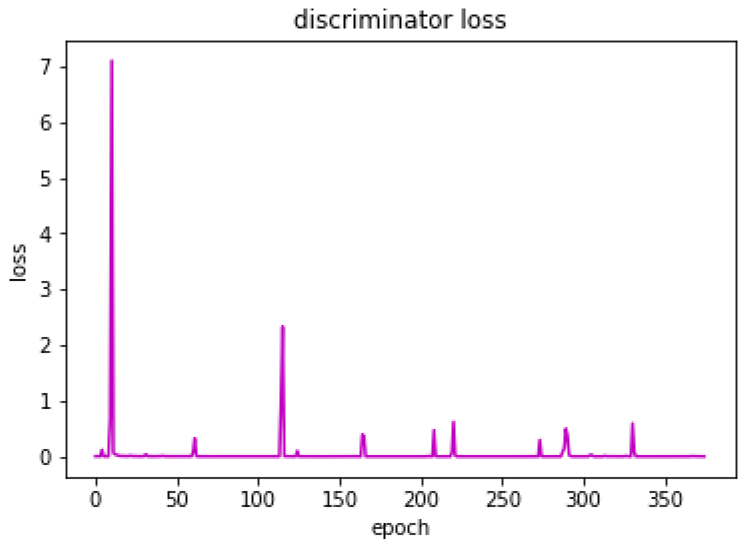
VI. Validation

In the development of GANs, validation often takes the form of investigating the produced images/objects. In our code, we split our dataset into a training and validation split. We then used the validation set to calculate the FID and KID, both often used to measure the performance of GANs and based on the InceptionV3 model. We also generated and saved images after every epoch for every class and from that, we can get a pretty good idea of the model's performance (human perception).

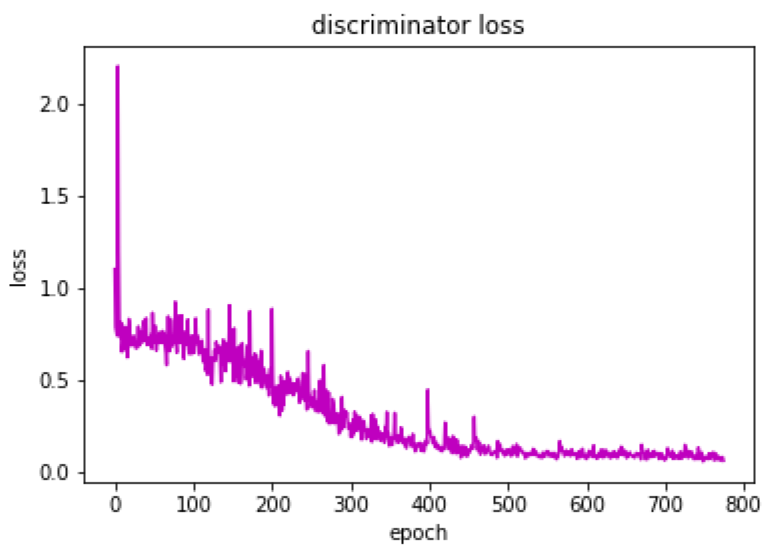
VII. Results

i. Discriminator loss

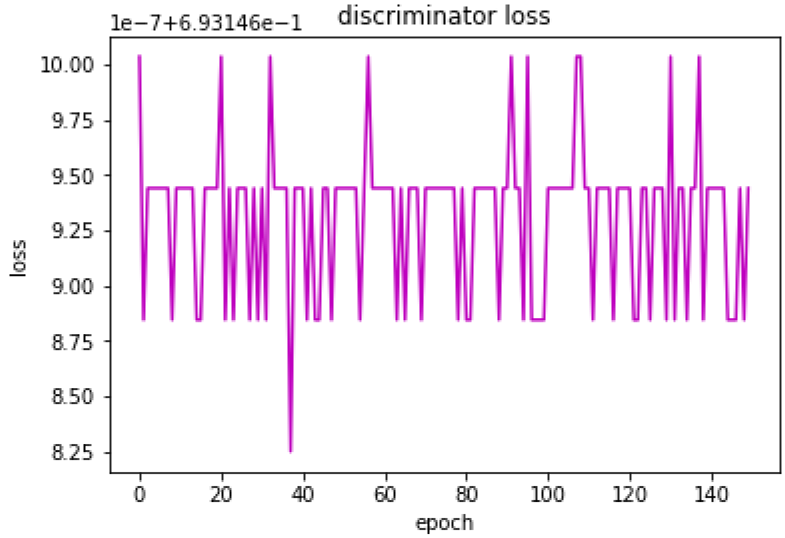
1. Model A



2. Model B



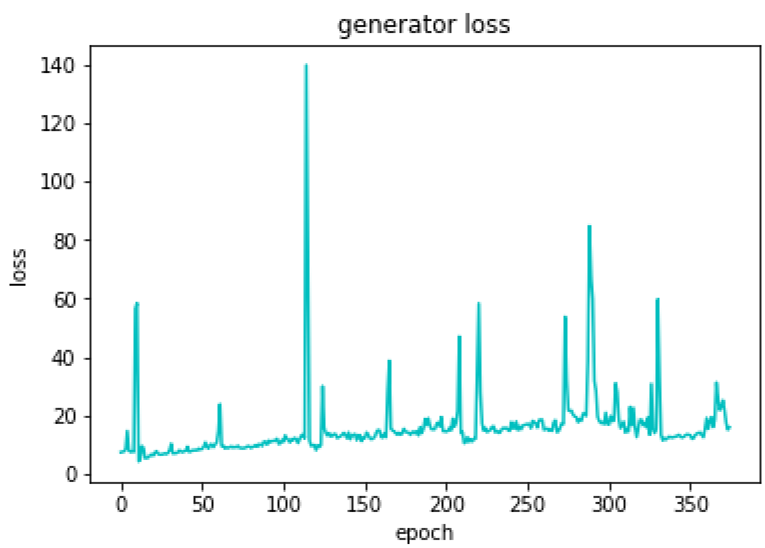
3. Model C



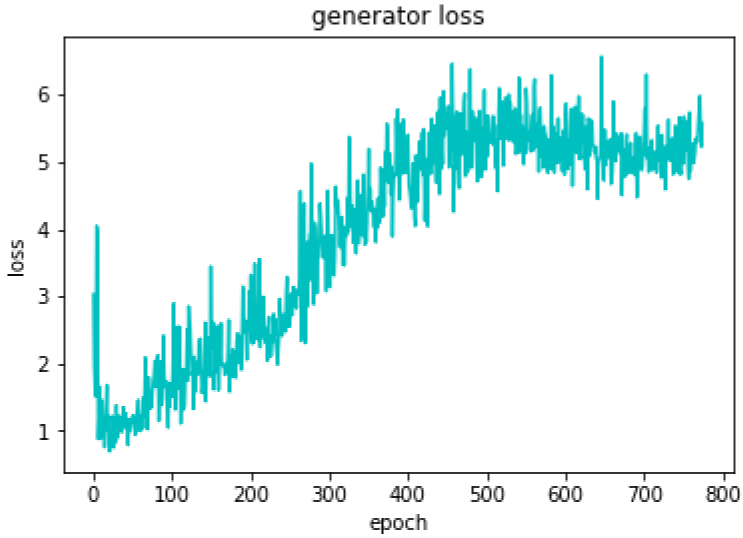
Model A seems to have resulted in a model going through steep peaks and long valleys. Model C's behavior was similar in some ways but its range was much smaller. Model B had shrinking loss over time, meaning that the discriminator was slowly coming to dominate over the generator.

ii. Generator loss

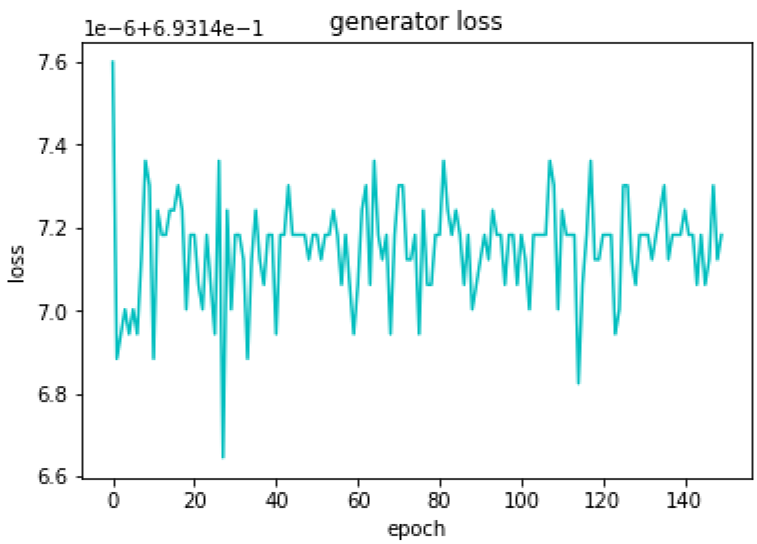
1. Model A



2. Model B



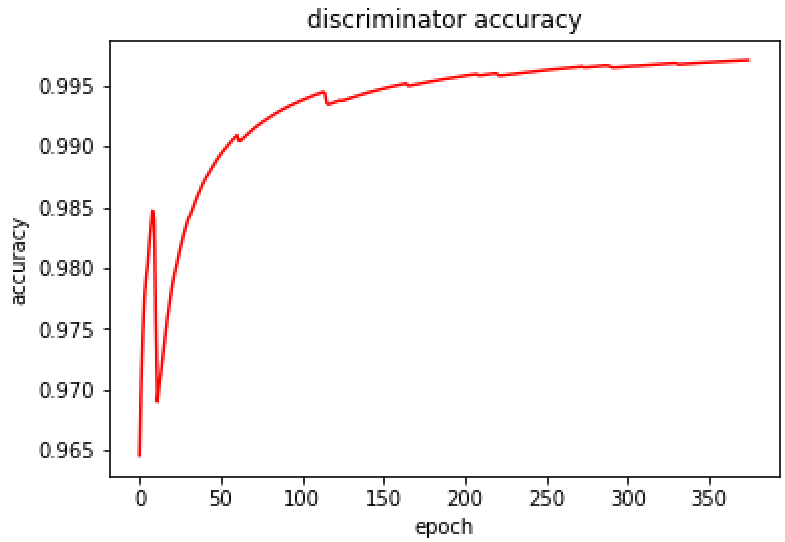
3. Model C



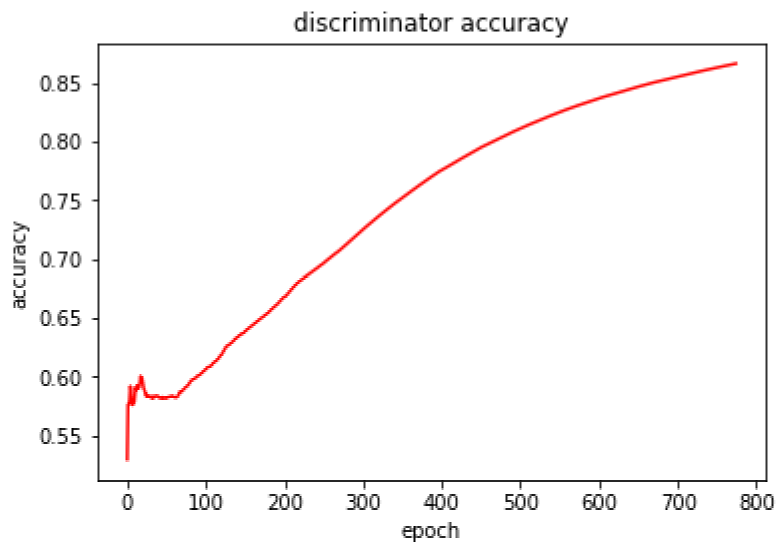
Model A and C have similar performance yet again with ups and downs; however, Model A had seriously long flats but were showing an overall trend (this may mean promising results with more epochs). Model B was again the outlier with a positive climb.

iii. Discriminator accuracy

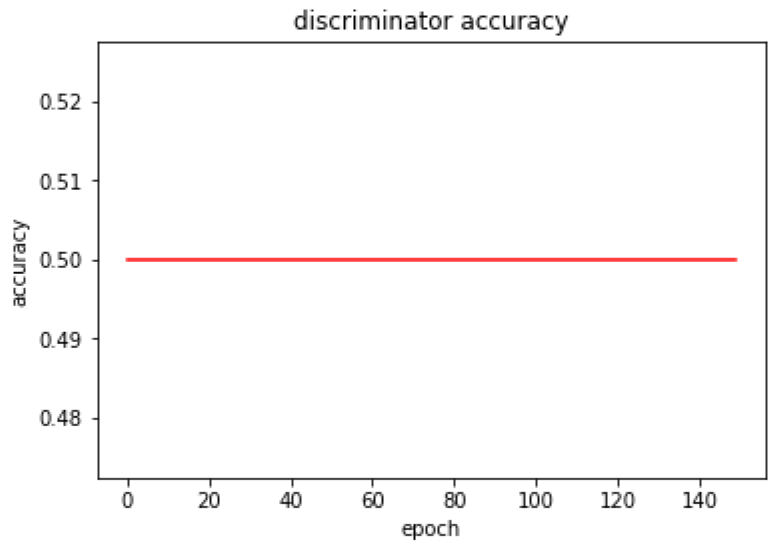
1. Model A



2. Model B



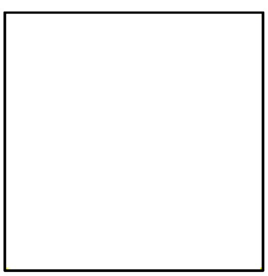
3. Model C



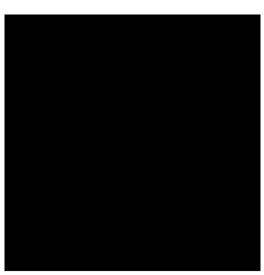
In Model C, the discriminator seems to only produce one kind of prediction or was pretty much as good as random. Model A and B saw the most upward improvement of discriminator accuracy. However, with high discriminator accuracy, we can imagine that the generator isn't doing very well in tricking the discriminator.

Images Generated:

1. Model A
   a. First Batch

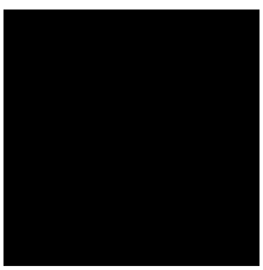
   b. Last Batch

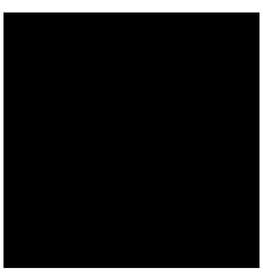
2. Model B
    a. First Batch

    b. Last Batch

3. Model C
    a. First Batch



    b. Last Batch



Upon analyzing the images generated, it is evident that Model B produces the best images. This could be attributed to two reasons: firstly, model B has run for longer epochs and it constitutes a better architecture that is prepared based on research. Contrastingly, Models A and C are basic compared to Model B, hence they produce poor images.

iv. FID and KID

1. Frechet Inception Distance (FID)

   FID represents the gaussian distance between the generated distribution and the actual data distribution. The lower the number, the better the model is.

2. Kernel-Inception distance (KID)

   KID measures the difference between the two probability distributions, the original data, and the generated data using samples drawn independently from each distribution.

Model	FID	KID
A	449.514	11.67314
B	*data was overwritten and couldn't be recovered in time	
C	570.711	5.461453

VIII. Discussion and Future Work

Based on the performance and analysis of our models, there is a scope for improvement by applying several constructive measures. Firstly, since the model is trained on the new dataset, it requires a lot of time to generate the epochs and produce results. As a result, the idea may be applied to a pre-trained model that may produce better results. Secondly, we can apply more pre-processing and data cleaning on the dataset to use as input to train the model since the model may perform better with distinctive genres and styles, as discussed previously. Additionally, we can try to evaluate how the randomization of the tensor affects the originality and creativity of the inputs produced. We can further contact experts in algorithmic art and GANs to receive insight on how to optimize the working of our models. More advanced models and bigger datasets with longer computation time would enable us to have better results with better images.

IX. Youtube Link

https://youtu.be/SyEtEyU9jmQ