Creating artificial galaxies using Generative Adversarial Networks.

Abstract

The study of extragalactic astronomy and the relationship between galaxy populations and their parameters is fast gaining deeper interest, therefore the need to create artificial images of galaxies has become more vital. To achieve this, we use an artificial neural network class known as "generative adversarial networks" (GANs) which can create realistic but fake images that are almost exactly like those in a training set. These images are typically small in GAN designs, therefore a variation called Deep Convolutional GANs (DCGANs) was utilized to produce arbitrary huge images of galaxies, where the training images showed some degree of periodicity.

The input dataset consists of 50,000 galaxy images obtained from Galaxy Zoo 2 (Willett et.al., 2013) fed into the DCGANs model. DCGAN uses convolutional and convolutional-transpose layers in the discriminator and generator networks respectively, replacing pooling layers with stride convolutions (Radford et al., 2015). To evaluate how good the outputs are, we got a professional expert to judge whether they were real or fake, and we ended up with 52 (81.2%) and 61 (95.3%) real out of 2 batches of 64 images in each. We calculated the Gini indices (GI) for them and discovered the Gini indices were significantly different between the real and fake images. This demonstrates that there is some residual in this process by which a computer could still identify the fake galaxies. We also evaluated the smoothness coefficient for the galaxy images and discovered the smoothness was the same for real and fake images.

1 Problem Statement

Astronomers stand to benefit substantially from the successful generation of plausible galaxy images. This will serve in data augmentation where a large amount of data is required for research. For example, Galaxy Zoo 2 (Willett et.al., 2013) is a crowdsourced astronomy project which enlists volunteers to help in the collection of galaxy images in large numbers. The task in this research will focus on how to produce artificial galaxy images, as close to the original galaxy images as possible and plausible enough to fool an astronomer, just as it has been used to produce photorealistic object images such as faces, birds, and interior or outdoor settings, image translation from a source domain to a target domain, and high-definition image generation from low-definition photographs, (Wang et al. 2017). This will be achieved using generative adversarial networks.

1.1 Introduction

One of the most exciting areas of machine learning today is generative adversarial networks (GANs). GANs are generative models that use artificial neural networks to produce synthetic data via a game of two adversaries: the generator, and the discriminator (Wang et al.,2017). The generator's goal is to learn the distribution of real data, whereas the discriminator's goal is to accurately identify whether the input data is from real data or from the generator. To win the game, the two players must constantly enhance their generation and discrimination abilities, respectively (Wu et al., 2018).

The discriminator is tasked with identifying which pixels in an output image were produced by the generator (Little et al., 2021). It receives batches of labeled real and generated data examples and outputs a single value for each example, the probability that it came from the real distribution, rather than the generator. If this value is close to 1, then it would be considered real; closer to zero would be classified as fake. The discriminator is a supervised classification model that gets penalized for misclassifying fake or real instances.

To match the growing research into galaxies and their parameters there is a need to create a lot of artificial galaxy images for astronomers to test out parameters. To create a batch of galaxy images, a set of latent points is produced and sent to the generator model. In a single GAN training cycle, genuine galaxy images from the problem domain are first chosen as a batch. The batch of real and generated images will be utilized to update the discriminator, minimizing the binary cross-entropy loss that is used in all binary classification problems. The discriminator model will be used to update the generator. The discriminator is shown generated images as if they were real (and not generated), and the error is transmitted back through the generator model. As a result, the generator model is modified to produce galaxy images that are more likely to fool the discriminator.

The GAN model takes the following steps:

1. The discriminator is trained with genuine images (and identifies them as genuine).
2. The discriminator is trained with generated images (and identifies them as generated).
3. The generator is trained with generated images (and recognizes them as genuine).

The discriminator learns how to distinguish between genuine and generated galaxy images in the first two steps. In the last step, the generator learns how to trick the discriminator by producing fake galaxy images that look like genuine ones.

Consequently, we have two feedback loops:

1. The discriminator is in a feedback loop with the galaxy images' known ground truth.
2. The discriminator and generator are connected, as adversaries in a feedback loop.

In the next section (1.1), we will give a background to the study surrounding the use of GANs for producing artificial galaxies. In section 1.2, we will look at related works in generative models. Section 2 gives the data description while in section 3 the methodology provides an understanding of GANs, DCGAN, its architecture, and training. In section 4, we will present our result followed by a discussion in section 5 which evaluates our research findings and considers potential future studies. Finally, the conclusion highlights the research outcomes in section 6.

1.1 Background

A new breed of artificial intelligence technology, known in the tech community as deepfake, is emerging from a combination of deep learning and fake. The result of this technology is fakery that is more realistic than ever before (Westerlund, 2019). Even professional photographers, videographers, and artists have been fooled by the deepfake phenomenon. Deep learning builds generative models from data using artificial neural networks. By backpropagating the loss, or error, across the network and modifying the weights to find the best solution, a neural network is trained and learns iteratively (Bengio and LeCun, 2007). GANs are an approach to estimating generative models using adversarial nets that Goodfellow et al. first presented in 2014 for semi-supervised and unsupervised learning (Goodfellow et al., 2014). GANs train two Neural Network models: A generative model that captures the data distribution and a discriminative model that seeks to determine whether a sample is from the model distribution or the data distribution (Little et al., 2021). On the same dataset, GANs execute alternating training in a procedure that is analogous to a two-player minimax game by training the discriminator while holding the generator constant and vice versa. The generative model depends on feedback from the discriminative model to generate a data sample and begins with noise as input.

According to Little et al.:

The GAN plays a minimax game, where the whole network tries to optimize the function V (D, G). The GAN training operation is expressed by the value function in the equation:

$$\underset{G}{min}\,\underset{D}{max}V(D,G) = \mathbb{E}_{x\sim Pdata(x)}[logD(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim \boldsymbol{Pz}(\boldsymbol{z})}\left[\log\left(1 - D\left(G(\boldsymbol{z})\right)\right)\right].$$

G(z), which transforms the noise z we input into the image data, defines the generator.

The discriminator's predictions on the dataset should be as near to 1 as possible, and the generator's predictions should be as close to 0 as possible. We employ the log-likelihood of D(x) and 1-D(z) in the goal function to do this. The log makes sure that it is penalized more severely the closer it is to an inaccurate value.

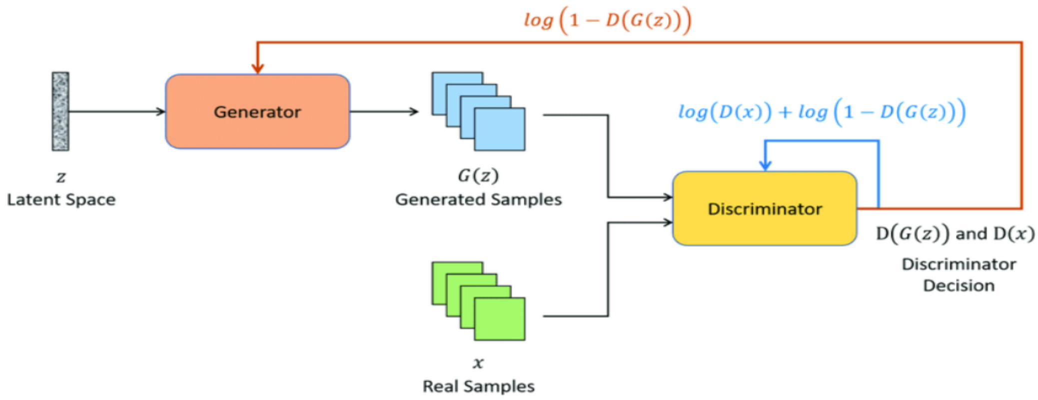

Figure 1: The GAN architecture (Vint et al., 2021).

The likelihood that the input x came from the actual dataset or not is output by the discriminator, which is represented by D(x). $\mathbb{E}$ is the expectation operator.

The discriminator receives the output G(z) from the generator along with a sample from the real distribution. Depending on whether it thinks a sample is real (1) or fake (0), the discriminator gives each sample a value. The generator is trained to minimize the function log (1 - D(G(z))) and is then used to examine the performance of the two models using these two outputs. In turn, this teaches the generator to create images that the discriminator cannot tell are fake (i.e., D(G(z)) ≈ 1). The discriminator is taught to maximize the function: log(D(x)) + log (1- D(G(z))) along with the generator, with the end goal of training the discriminator to maximize the likelihood of accurately recognizing the genuine samples, (D(x)), whilst also correctly identifying the synthetic samples (D(G(z))) (Little et al., 2021).

1.2 Literature review

The significance of developing generative models for astronomical images cannot be overstated, even though much of the deep learning literature on generative models have been concerned with natural images from everyday life settings (Lanusse et al., 2021). In November 2020 MBM, a South Korean cable channel, essentially duplicated one of its news anchors using GANs, allowing for continuous news transmission for 24 hours (Gelgel N, 2020). Bill Hader recalled the experience of meeting Tom Cruise for the first time at a table read during his appearance on the Late-Night Show with David Letterman. Hader's face gently changed into Cruise's as he impersonated Cruise in the deepfake, which was shared on YouTube (Delfino, 2022). In May 2019, the Dali Museum in Florida made an interactive deepfake creation of the late artist, Salvador Dali. Over 6,000 frames of videos from archival footage were used to bring Salvador to life and welcome visitors (Kidd & Rees, 2022). Coccomini et al. (2021) used 283 quality images of galaxies collected from Flickr scraper for one network and another 61636 images from the Galaxy Zoo dataset (Willett et.al., 2013) and trained for three days to generate synthetic images applying Lightweight GAN. Their model was implemented with a batch size of 3, using the Adam optimizer, and a learning rate of 0.0002.

Like the work of Coccomini et al., our research trained 50000 galaxy images from the Galaxy Zoo 2 dataset (Willett et.al., 2013) using the Adam optimizer and a learning rate of $2e^{-4}$. However, we implemented our model using DCGAN and a batch size of 64 to create artificial galaxies in 7 hours.

This research is motivated by Astronomers' need to test out algorithms that measure the parameters of galaxies such as their luminosity, and spectroscopic and photometric completeness (Blanton et al., 2005). The classification and study of galaxies in parametric research can be challenging and computationally intensive (Vikram et al., 2010), therefore a set of synthetic galaxies is ideal for this activity.

2 Data

The dataset for this research consists of 50000 images of galaxies with spectroscopic redshifts obtained from Galaxy Zoo 2(Willett et.al., 2013). The galaxy zoo is a large hub containing millions of galaxies drawn from astronomical surveys captured by telescopes. Below is a sample input of a batch of 64 real galaxy images in png.

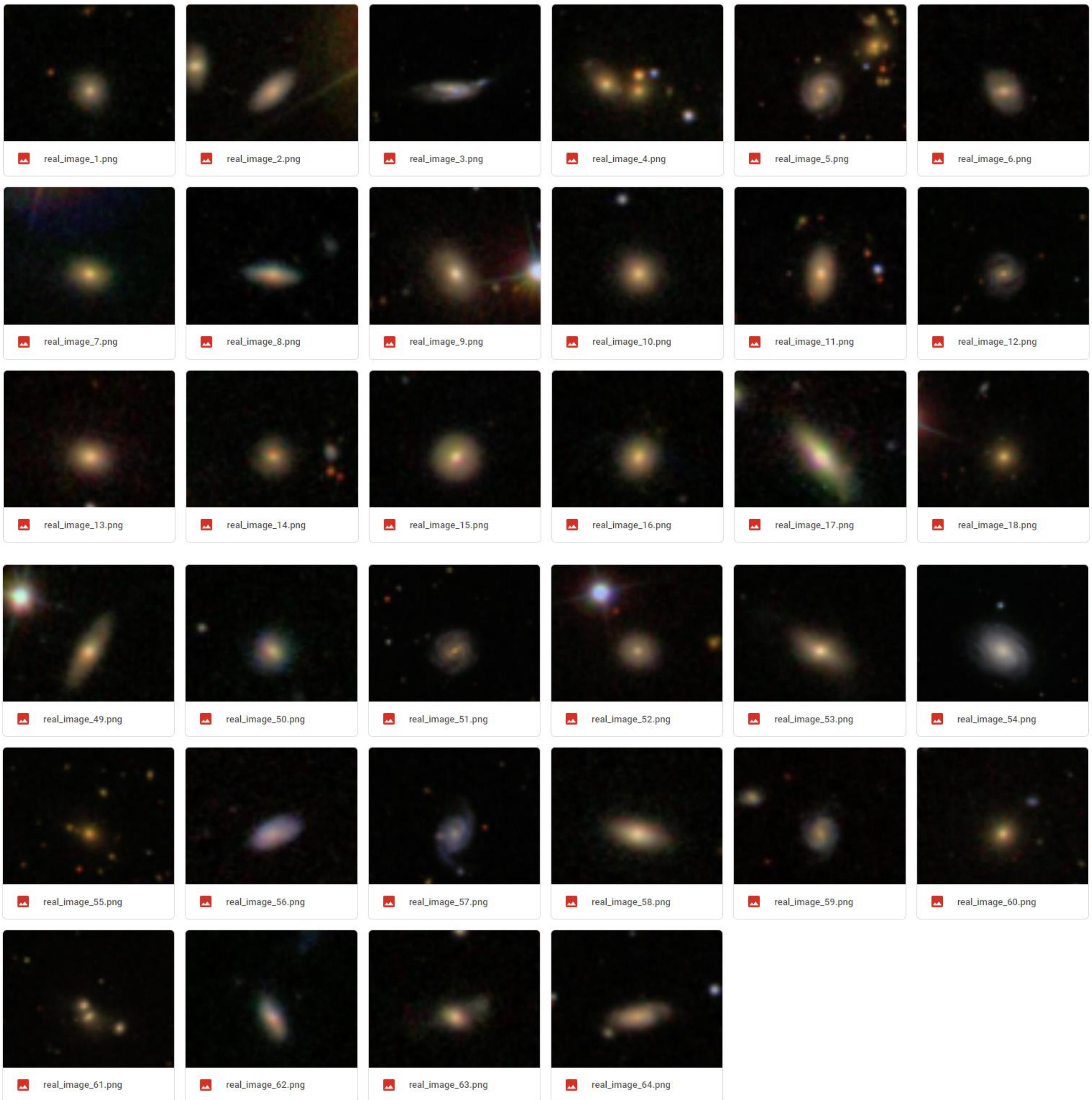

Figure 2: A section of input galaxy images.

3 GANs Methodology

GANs are neural networks that may be trained to produce fake but plausible copies of known input data. The idea behind GANs is simple, however different variants of GANs have been used for specialized data synthesis tasks. The types of GANs include Vanilla GAN, Deep Convolutional GAN (DCGAN), Style GAN, Super Resolution GAN, and Conditional GAN (GeeksforGeeks, 2019).

3.1 Deep Convolutional GAN

In this research, we have chosen to use DCGAN – a class of Convolutional Neural networks (CNN) for our generative modeling. Our preference for DCGAN was borne out of its architectural limitations which show that they are excellent candidates for unsupervised learning (Radford et al., 2015).

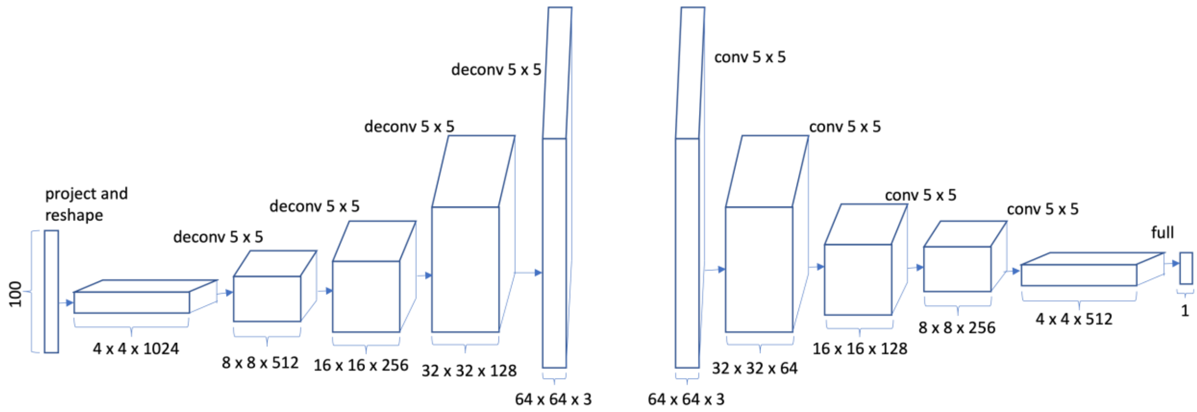

Figure 3: The DCGAN Architecture.

3.2 Generator network

A vector or a matrix of random values, known as a latent tensor, is utilized as the generator's input and serves as the image's seed. A latent tensor of shape (128, 128, 3) will be transformed into an image tensor of shape (64 x 64 x 3) by the generator. We will utilize the PyTorch framework's ConvTranspose2d layer, which functions as a transposed convolution (also referred to as a deconvolution) (Hany and Walters, 2019).

3.3 Discriminator network

An image is fed into the discriminator, which attempts to categorize it as "genuine" or "generated". It is comparable to other neural networks in this regard.

A CNN, which generates a single number for each image, will be used. To gradually shrink the size of the output feature map, we will use a stride of 2.

3.4 DCGAN Architecture

Brownlee (2019) suggested the following methods for the DCGAN architecture:

1. First, we use upsampling with 2-strided transpose convolutional layers (no pooling).
2. We use LeakyReLU for the discriminator hidden layers, and sigmoid, which outputs between (0) and (1) for binary classification, as the output layer. Since this is a binary task, Binary Cross entropy is chosen as the cost function. To keep the output image scale between -1 and 1, we use the Tanh function in the generator output and ReLU for the hidden layers.
3. Batch normalization and Gaussian weight initialization are applied to achieve a mean of zero (0) and a standard deviation of 0.02.
4. We implement Adam optimizer with learning rate = 0.0002, betas = (0.5, 0.999).

Now, we combine both models into one adversarial network, with the output being the output of the discriminator and the input being a 100-dimensional vector.

3.5 Training the GAN

We load the data and divide it into many batches to feed into our model, then we initialize our GAN network based on the DCGAN methods described above, and we execute the training loop for the required 150 epochs. To ensure that created images from the same class do not all look the same, we generate some random noise and remove some images from our dataset. The generator is then used to produce a vector, say X, that contains both fake and real images. Then, we make another vector, say Y, that corresponds to X and contains the "right responses," with generated images tagged 0 and the genuine images labeled 0.9 (labeled 0.9 instead of 1 using one-sided label smoothing, which improves the GAN train) (Brownlee, 2019). Here, we update the discriminator since we need to alternate between training the generator and the discriminator.

4 Results

Synthetic images of galaxies were generated using DCGAN. After training the GAN, below are some samples of galaxy images generated by the model.

Figure 4: A sample of generated galaxy images.

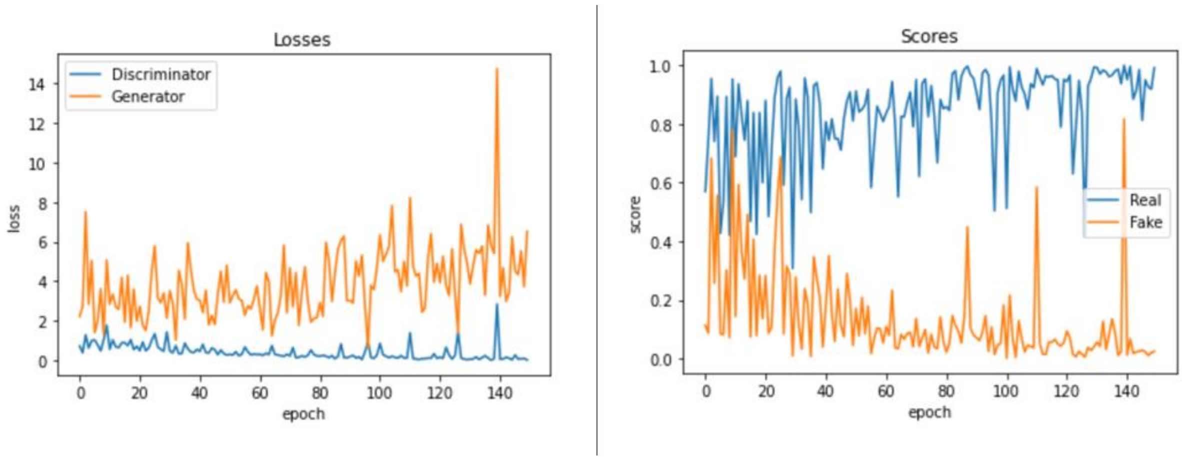

Figure 5: Generator and Discriminator Loss and scores during training.

Table 1: Generator and Discriminator losses.

Epoch	Generator loss	Discriminator loss
1	2.24	0.73
6	1.41	1.04
7	2.02	0.79
9	1.32	1.65
10	5.07	1.79
65	1.26	0.76
66	2.03	0.31
97	0.66	0.97
98	3.81	0.16
127	1.41	1.46
128	6.87	0.13

We can infer from Figure 5 and Table 1 that the generator loss began at 2.24 while the discriminator loss started very low, at 0.73. This is not surprising as the generator would initially fail at fooling the discriminator before it learns to produce plausible galaxy images. Table 1 reveals that when the generator successfully fools the discriminator, the discriminator quickly learns at classifying the generated images correctly. The generator in turn also learns again and improves at deceiving the discriminator, and the minimax game continues until the discriminator fails to distinguish between genuine and fake images.

4.1 Evaluation of GANs

In a variety of problem domains, GANs have shown to be remarkably adept at producing both big and high-quality synthetic images. Instead of being trained directly, the generator models are trained by a second model - the discriminator, that learns to differentiate real images from generated images (Goodfellow et al. 2014). As a result, the generator model lacks any objective function or objective measure (Salimans et al., 2016). Since GANs lack an objective function, it is challenging to compare the effectiveness of various models. To measure the quality of data generated by GAN models, different qualitative and quantitative metrics can be adopted.

Qualitative metrics are measures that are non-numerical and frequently include subjective human evaluation or comparison. According to Alqahtani et al. (2019), the most widely used qualitative evaluation methods are:

1. Nearest Neighbors.

2. Rapid Scene categorization.

3. Rating and Preference Judgment.

4. Evaluating Mode Drop and Mode Collapse.

5. Investigating and Visualizing the Internals of Networks.

Visual evaluation of the synthesized sample is used in most research projects in the picture domain; however, visual methods are arbitrary and may even be deceptive (Gerhard et al, 2013). Rating and Preference Judgment was used for this research. To achieve this, we got a professional expert to judge whether the galaxies in two batches of synthetic images were real or fake.

This however comes with a lot of shortcomings. First off, using human vision to assess the quality of generated images is much more expensive and biased (Denton et al., 2015a). Even so, it is challenging to duplicate and does not accurately represent the capabilities of models. Second, the variability of human inspections necessitates averaging across a wide number of participants. Thirdly, a sample-based evaluation may be skewed toward overfit models, making it a poor gauge of a log-likelihood-based excellent density model (Theis et al., 2015).

Quantitative metrics for evaluating GAN involve the calculation of precise numerical scores to measure the effectiveness of generated images. There are numerous quantitative methods that have been suggested and used by various scientists, the most popular are the Inception Score (Salimans et al., 2016), the Frechet inception distance (Heusel et al., 2016), and Average log-likelihood (Goodfellow et al., 2014). For this research, we used Gini Index (Singh and Zwiggelaar, 2004), and smoothness.

4.2 Gini index

The Gini index (GI) calculates at pixel level the likelihood that a given variable will be incorrectly classified when it is randomly selected (Gini, 1936). It is a widely used economic metric that assesses how equally wealthy a country is. GI ranges from 0 to 1, where 0 indicates that all the elements belong to a single class or that there is only one class, and 1 indicates that the elements are spread randomly across the classes.

Equally distributed elements are indicated by a Gini Index of 0.5 in some classifications.

The basis for use of the GI was borne out of its capacity to gauge the homogeneity and uniformity of the pixels within a space to assess the quality of created galaxy images (Habba et al, 2018). GI is calculated as:

$$Gini = 1 - \sum_{i=1}^{n}(p_i)^2$$

Where pi is the likelihood of an image being mapped to a given class.

The GI for each real and generated galaxy image was evaluated in a batch of 64 and a sample is shown in figure 6.

Figure 6: Gini indices of Real and generated images.

We can see from figure 6 that quite a few generated galaxy images have Gini indices of 0.5 and above. The GI of real images, mostly above 0.5 were higher than those of generated images, except for a few generated images which had a higher GI than the real images. We will therefore perform a hypothesis test on the significant difference between the GI of the real and generated galaxy images.

4.3 Hypothesis Test

The two hypotheses for our paired samples t-test are as follows:

- H_o: μ1 = μ2 (the mean of real galaxy Gini indices and the mean of generated galaxy Gini indices are equal).
- H_a: μ1 ≠ μ2 (the mean of real galaxy Gini indices and the mean of generated galaxy Gini indices are not equal).

  Gini mean value (real) = 0.5908

  Gini mean value (generated) = 0.4699

  p-value = 3.905728096857875e-18

  Therefore, since the p-value of the test (3.905728096857875e-18) is less than 0.05, we reject the null hypothesis.

This implies we have sufficient evidence to conclude that the Gini mean value of the real galaxies is not equal to the Gini mean value of generated galaxies. However, since we rejected the null hypothesis, this then means that we have not quite faked the images that the computer will be able to recognize the fake images.

4.4 Smoothness

The smoothness (or clumpiness) is one of the three model-independent parameters (Concentration, Asymmetry, and Smoothness), that is simple yet considered powerful at understanding the star formation properties of galaxies (Conselice, 2003). The smoothness, S is calculated as:

$$S = \frac{I-B}{I}$$

where I is the original galaxy image,

and B is the blurred image by the factor 0.3 × r (η = 0.2)

The smoothness for each real and generated galaxy image was calculated in a batch of 64 and a sample is shown in figure 7.

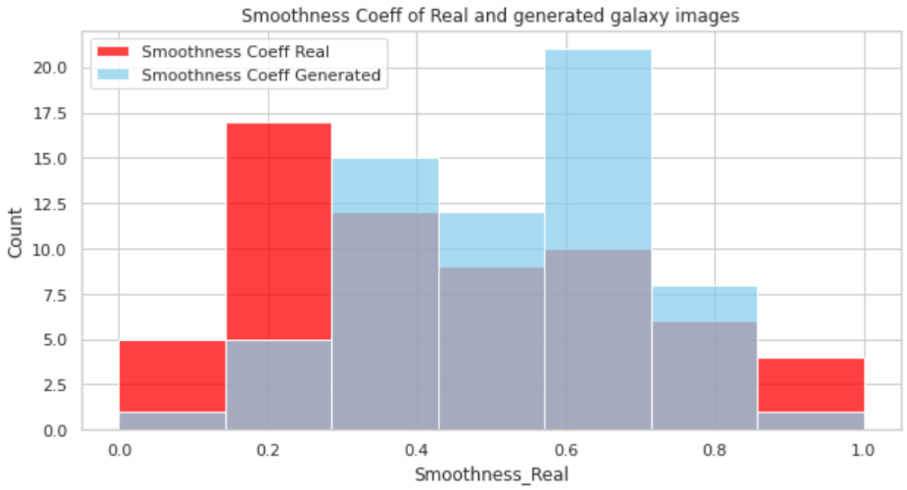

Figure 7: Smoothness of Real and generated images.

The above histogram shows the smoothness coefficient of real and generated galaxies overlapping all through from 0 to 1.

Therefore, we will carry out a hypothesis test to compare the smoothness of the real with the smoothness of the generated galaxy images.

4.5 Hypothesis Test for smoothness

The two hypotheses for our paired samples t-test are as follows:

- H_0: µ1 = µ2 (the mean of real galaxy Smoothness is equal to the mean of generated smoothness).
- H_a: µ1 ≠ µ2 (the mean of real galaxy smoothness is not equal to the mean of generated galaxy smoothness).

  Smoothness real mean value: 0.4466

  Smoothness generated mean value: 0.5180

  p-value 0.06380153230215922

- Since the p-value of the test (0.06380153230215922) is greater than 0.05, we accept the null hypothesis.

We can infer from the above that the artificial galaxy images have the same smoothness as the real galaxies.

5 Discussion

We have shown a framework for generating images of galaxies that looked exactly like real using the DCGAN. During a visual evaluation of two batches each of 64 generated galaxy images (figure 8), the astronomer identified 12 images as fake in one and 3 images as fake in the second batch representing 88.3% success. Even though DCGAN, a strong variant of GAN was implemented with parameters considered most suitable, it is not surprising to have few generated images in a batch that fail at fooling the astronomer.

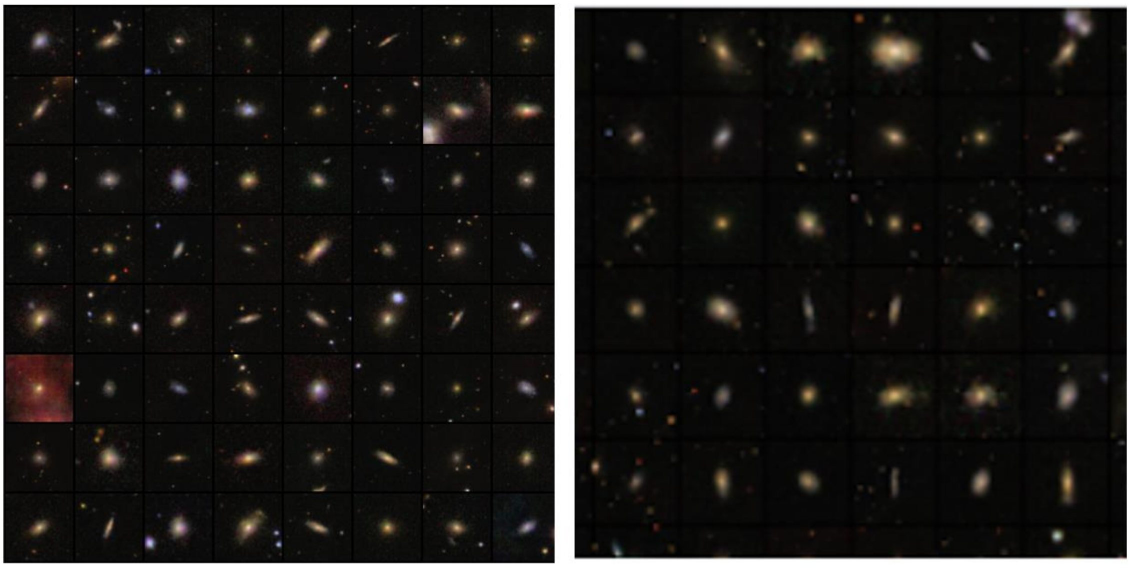

Figure 8: A Sample of 128 generated images.

This is because neural networks in some instances are prone to small errors. A few flaws like the presence of artifacts, awkward shape, or strange color of a galaxy or its background easily give away the quality of generated images when observed visually. This imperfection can also be observed in big GANs projects like the popular website thispersondoesnotexist.com which has been developed using StyleGAN to generate random samples of non-existing human faces with every click (Karras et al., 2020).

Given more time, we would like to improve the quality and speed of our GAN model using Free AdverSarial Training or FastGAN (Zhong et al. 2020). FastGAN is an efficient algorithm capable of achieving better image generation quality in less training time by using 2-4 GPUs.

The concentration and asymmetry of galaxy images would also be measured to obtain their stellar light distributions.

To accomplish the project's greater goal of creating tools that support tacit intent in architectural design, more work still must be done. In the future, rather than convert the galaxy images to png format, we would try to carry out the experiment in fits format. Unlike other formats which are compressed, fits images are lossless. Another future work suggested in this regard is the Galaxy2Galaxy, a library of datasets, utilities, and models which is still under development and would be used to produce generative models for astronomical images. Galaxy2Galaxy will aim to accelerate research in deep learning models for processing celestial images (Lanusse et al., 2021). This research has gone ahead of previous work creating artificial galaxies using GANs in less time and evaluating image quality with the Gini index and smoothness.

6 Conclusion

In this research, we have shown the ability of DCGAN to achieve highly plausible galaxy images with large data.

- The expert rated 113 generated galaxy images out of 128 as real, representing 88.3% success.
- The Gini mean values for real and generated galaxies were evaluated as 0.5908 and 0.4699 respectively.
- The smoothness mean values for real and generated images were 0.4466 and 0.5180 respectively, estimated as same based on hypothesis test.
- From 150 epochs of training, over 90% of real scores achieved were above 0.9 while over 90% of fake scores were below 0.2. The closer the real scores to 1, and the closer the fake scores to 0, the better and more successful the model.

References.

Alqahtani, H., Kavakli-Thorne, M., Kumar, G., & SBSSTC, F. (2019). An analysis of evaluation metrics of gans. In *International Conference on Information Technology and Applications (ICITA)* (Vol. 7).

Bengio, Y., & LeCun, Y. (2007). Scaling learning algorithms towards AI. *Large-scale kernel machines*, *34*(5), 1-41.

Blanton, M. R., Lupton, R. H., Schlegel, D. J., Strauss, M. A., Brinkmann, J., Fukugita, M., & Loveday, J. (2005). The properties and luminosity function of extremely low luminosity galaxies. *The Astrophysical Journal*, *631*(1), 208.

Brownlee, J. (2019). *Generative adversarial networks with python: deep learning generative models for image synthesis and image translation*. Machine Learning Mastery.

Coccomini, D., Messina, N., Gennaro, C., & Falchi, F. (2021). Generative Adversarial Networks for Astronomical Images Generation. *arXiv preprint arXiv:2111.11578*.

Conselice, C. J. (2003). The relationship between stellar light distributions of galaxies and their formation histories. *The Astrophysical Journal Supplement Series*, *147*(1), 1.

Denton, E. L., Chintala, S., & Fergus, R. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. *Advances in neural information processing systems*, *28*.

Delfino, R. (2022). Deepfakes on Trial: a Call to Expand the Trial Judge'S Gatekeeping Role to Protect Legal Proceedings from Technological Fakery. *Available at SSRN 4032094*.

GeeksforGeeks (2019). Generative adversarial network (GAN). Available online: Generative Adversarial Network (GAN) - GeeksforGeeks [Accessed 06/08/2022].

Gelgel, N. M. R. A. (2020). Will technology take over journalism?. *power*, *153*, 164.

Gerhard, H. E., Wichmann, F. A., & Bethge, M. (2013). How sensitive is the human visual system to the local statistics of natural images? *PLoS computational biology*, *9*(1), e1002873.

Gini, C. (1936). On the measure of concentration with special reference to income and statistics. *Colorado College Publication, General Series*, *208*(1), 73-79.

Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Habba, M., Ameur, M., & Jabrane, Y. (2018). A novel Gini index-based evaluation criterion for image segmentation. *Optik*, *168*, 446-457.

Hany, J., & Walters, G. (2019). *Hands-On Generative Adversarial Networks with PyTorch 1. x: Implement next-generation neural networks to build powerful GAN models using Python*. Packt Publishing Ltd.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two-time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, *30*.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8110-8119).

Kidd, J., & Rees, A. J. (2022). A MUSEUM OF DEEPFAKES?. *Emerging Technologies and Museums: Mediating Difficult Heritage*, 218.

Lanusse, F., Mandelbaum, R., Ravanbakhsh, S., Li, C. L., Freeman, P., & Póczos, B. (2021). Deep generative models for galaxy image simulations. *Monthly Notices of the Royal Astronomical Society*, *504*(4), 5543-5555.

Little, C., Elliot, M., Allmendinger, R., & Samani, S. S. (2021). Generative Adversarial Networks for Synthetic Data Generation: A Comparative Study. *arXiv preprint arXiv:2112.01925*.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, *29*.

Singh, H., & Zwiggelaar, R. (2004). G-Images: Towards Multilevel Unsupervised Image Segmentation. In *ICVGIP* (pp. 473-478).

Theis, L., Oord, A. V. D., & Bethge, M. (2015). A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.

Vikram, V., Wadadekar, Y., Kembhavi, A. K., & Vijayagovindan, G. V. (2010). Pymorph: automated galaxy structural parameter estimation using python. *Monthly Notices of the Royal Astronomical Society*, *409*(4), 1379-1392.

Vint, D., Anderson, M., Yang, Y., Ilioudis, C., Di Caterina, G., & Clemente, C. (2021). Automatic Target Recognition for Low-Resolution Foliage Penetrating SAR Images Using CNNs and GANs. *Remote Sensing*, *13*(4), 596.

Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., & Wang, F. Y. (2017). Generative adversarial networks: introduction and outlook. *IEEE/CAA Journal of Automatica Sinica*, *4*(4), 588-598.

Wang, Z., & Bovik, A. C. (2009). Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE signal processing magazine*, *26*(1), 98-117.

Westerlund, M. (2019). The emergence of deepfake technology: A review. *Technology Innovation Management Review*, *9*(11).

Willett, K. W., Lintott, C. J., Bamford, S. P., Masters, K. L., Simmons, B. D., Casteels, K. R., ... & Thomas, D. (2013). Galaxy Zoo 2: detailed morphological classifications for 304 122 galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, *435*(4), 2835-2860.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., ... & Fu, Y. (2018). Incremental classifier learning with generative adversarial networks. *arXiv preprint arXiv:1802.00853*.

Zhong, J., Liu, X., & Hsieh, C. J. (2020). Improving the speed and quality of gan by adversarial training. *arXiv preprint arXiv:2008.03364*. Zhong, J., Liu, X., & Hsieh, C. J. (2020). Improving the speed and quality of gan by adversarial training. *arXiv preprint arXiv:2008.03364*.