Designing a Generative Adverserial Network to Convert Portraits into Realistic Images and Evaluating its Performance on Different Art Styles

Zahra Moradi (2690281), Nisrine Mokadem (2692177), Mara Spadon (2688689), Katrina Slebos Perez (2714445) and Elisa Bermejo Casla (2665178)

Abstract

This report aims to investigate how different art styles influence the performance of a CycleGAN model that converts portraits to photo-like images. To examine this, a CycleGAN was designed which was then tested on three different datasets. Each dataset consisted of portraits from a different art style, namely, northern renaissance, post-impressionism and ukiyo. One of the major findings of the paper is that the model performs better when translating northern renaissance paintings than post-impressionist and ukiyo ones. The results indicate that the model has a higher performance when using realistic art as opposed to abstract art.

1 Introduction

New machine learning methods are being developed to work with images which are actively being used in diverse areas such as medicine [1] and facial recognition [2]. Another area of research that can be explored through the use of Artificial Intelligence, is visual arts [3]. In this field, diverse tasks have received the attention of researchers, from classification, identification, and generation of images to 3D modeling. This project will undertake the task of converting portraits to realistic images. That is, building a mapping from the input images that are portraits to the photo-like output images.

1.1 Generative Adversarial Networks

The main technique used in this research is Generative Adversarial Networks, or GANs [4]. This deep generative model consists of two neural networks that interact with each other called the generator and the discriminator. The aim of deep generative modeling is to produce new data examples that resemble the given dataset. The generator tries to improve the generated data examples in such a way that the discriminator will eventually classify it as real data. This works as follows. First, the generator starts by feeding random noise. Based on the feedback that it receives from the discriminator, the model updates itself. The cycle is repeated until the generated example gets classified as a real example. The discriminator is able to distinguish the real data from the fake data by comparing the generated fake data to the original dataset with real examples. There are multiple types of GANs, such as the vanilla GAN, conditional GAN, and the styleGAN, but we will focus solely on the CycleGAN [5]. The CycleGAN is mainly used for image transformation tasks, which is the topic of this project. The CycleGAN expands on the original GAN design as it trains two generator models and two discriminator models simultaneously. This it was decided to design a CycleGAN for this project, as it is a suitable method to obtain the desired outputs and flexible enough to adapt the networks to our needs.

1.2 Research Question

This paper investigates the question: *How does art style influence the performance of a CycleGAN that we designed to convert portraits into photo-like images?*

The purpose of this paper is to find out for which art style the CycleGAN has the best performance when converting the portraits to realistic images. We believe it would be interesting for both researchers in the field of AI, as well as for researchers in the field of art. To investigate this question, a CycleGAN was created that transforms paintings into images and vice versa. Secondly, its performance using different art styles was evaluated. The following three art styles were selected: northern renaissance, post-impressionism, and ukiyo. The hypothesis is that the performance when converting more realistic styles, such as northern renaissance, will be better than for those relatively abstract, such as post-impressionism and ukiyo. More specifically, the hypothesis is: The performance of the model will be the best for northern

renaissance ($P_{renaissance}$), and the performance for the post-impressionism ($P_{impressionism}$) will be better than the performance for ukiyo (P_{ukiyo}), but worse than for the northern renaissance. In other words, the hypothesis is:

$$P_{renaissance} > P_{impressionism} > P_{ukiyo}$$

2 Data Preparation

2.1 Datasets

The aim was to collect sufficient data to ensure that there is a big enough training set and data set for portraits, as well as a training set and a test set of real images. There are two training sets, one for portraits and one for real images. The training set for the portraits contains 1000 images of portraits, including both realistic as well as abstract portraits [6]. The training set for the real images is a subset of the CelebA dataset [7], which contains images of various celebrities from diverse backgrounds. The first 1000 images of the CelebA dataset were selected for the training set of the real images. There are four different test sets, one for the real images and one for each of the three art styles. The dataset for the real images contains 500 images that were selected from the CelebA dataset. These images did not overlap with the training set. For each art style, namely, northern renaissance, post-impressionism and ukiyo, an existing dataset was selected [8]. Each of these art style datasets contains approximately 150 portraits.

2.2 Data Preprocessing

The only dataset that required preprocessing was the portraits training set. Some of the images were deleted from the original dataset due to their low quality. Other images contained full-body portraits. These images were cropped so that they would center on the faces. After cleaning the portraits dataset, the first 1000 images were selected for the portraits training set.

2.3 Data Inspection

The goal was to find diverse datasets but upon inspection of our data, we discovered that white people are overrepresented in the datasets that include portraits. Most portrait datasets that are available on the internet are heavily biased towards white people. Buolamwini and Gebru [9] have shown that facial analysis benchmark datasets that overrepresent white people may lead to a machine model that performs worse on people of color. Therefore, our model may have lower accuracy levels for generated portraits of people of color. On the other hand, we did succeed in composing a more diverse dataset of real faces, so we would expect the conversion from portraits to pictures to be somewhat less biased.

3 Method

In this section, we will explain how the CycleGAN model was designed and evaluated by discussing the model architecture, training configuration and performance measures.

3.1 Model Architecture

CycleGANs have laid the foundation for numerous image processing and translation projects in recent years [5]. Following this trend, we adapted the original CycleGAN structure by introducing two separate neural networks *DiscriminatorF* and *DiscriminatorP* as our discriminators. Furthermore, image augmentation techniques were applied to our training images as fake data to enhance the performance of each discriminator. The augmentation was done by randomly changing the original colors –more specifically, brightness, contrast, saturation, translation, and cutout– of the real images and the fake images that are generated in order to better generalize the trained model and enhance its robustness in the inference step [10]. The *DiscriminatorP* handles the discrimination of real face images and portrait paintings and the *DiscriminatorF* handles the discrimination of the augmented images.

Our framework operates in two phases. In the first phase, the generators handle the conversion from portraits to face photo-like images and then from photos back to portrait (cycled-portrait), the conversion from face images to portrait paintings and back to face images (cycled-face), and the conversions of the domain images to themselves (same-face and same-portrait). The second phase consists of using a two-objective discriminator to minimize the risk of over-fitting.

Since the dataset is limited to around 1000 images, there is a high risk of over-fitting the discriminator, leading to the continuous deterioration of the results. Successful results under such conditions are achieved by early stopping. However, since each epoch of the training phase

takes a long time for image generation tasks, a two-objective discriminator is proposed to avoid the negative effects of over-fitting the discriminator and the need for early stopping.

Let the domain of the portrait paintings and the real face images be Y_1 and Y_2 respectively, and the populations be y_1 and y_2. The first phase consists of two generators that share the same neural network structure and weights (G_1 corresponding to Y_1 and G_2 corresponding to Y_2), and a discriminator *DiscriminatorP* which is designed to detect the real images from the generated images. The conversion of the real images of faces can subsequently be denoted by $G_1(Y_1) \rightarrow Y_2(y_2)$. Since the two generators use weight-sharing throughout the procedure, G_1 is trained to generate images that have a similar style to Y_2 and resemble the shape of Y_1.

In the second phase the augmented real face images and augmented fake images (which are generated in phase 1) are passed through a newly introduced discriminator called *DiscriminatorF* and processed by *Head1* and *Head2*. With the assistance of the local identity loss and cycle consistency loss, the two discriminators ensure that the images are consistent before and after translations. This method helps to generate results that maintain a style similar to Y_1 and are close in appearance to input y_2.

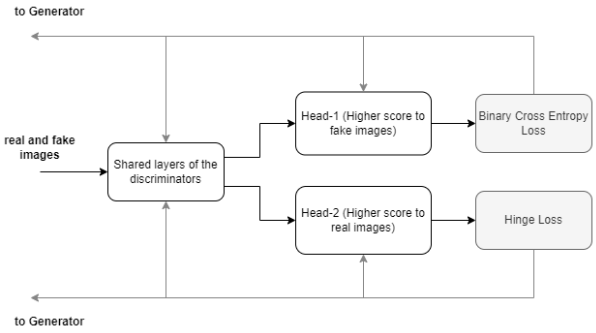

Figure 1: Schema of a two-objective discriminator.

Furthermore, our objective function monitors different aspects of the neural networks. The *DiscriminatorF* loss is used to preserve the texture information of the source domain y_1 in the target domain y_2. The cycle consistency loss is used to prevent a contradiction between learned mapping G and F. The identity loss helps to preserve the stability in the generated outputs and the local adversarial loss obtained from the *DiscriminatorP*. This preservation ensures that the model is trained to generate data that is indistinguishable from real data for both image domains y_1 and y_2. We will now explore the process in each phase extensively.

3.1.1 Phase One

The generator is a neural network that contains an encoder-decoder structure. The image is gradually down-sampled through the encoder and up-sampled through the decoder using a convolutional neural network with LeakyRelu activation [12] between the layers. To preserve the details of the input in the translation process, we modified a UNet [13] which is commonly used in image translation tasks. The skip connection concatenates the encoding layers before down-sampling and after up-sampling with the generator layers. The generators G_1 and G_2 will then translate the images of each domain Y_1 and Y_2 to each other, and from those back to themselves. This recycling will strengthen the generators' weights in obtaining results that are as close as the target y_2 in appearance. The details of the first phase are displayed in Figure 2.

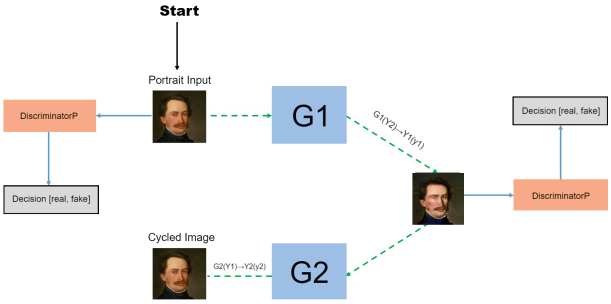

Figure 2: Pipeline of our framework in the first phase: we train the generators $G1(Y2) \rightarrow Y1(y1)$, $G_1(Y_1) \rightarrow Y_2(y_2)$ and the cycled-images of $G_1 \rightarrow Y_1$ and $G_2 \rightarrow Y_2$.

3.1.2 Phase Two

In this phase, the augmented images are processed by *DiscriminatorF*. The objective is to update *DiscriminatorF* and G_1. The augmentation is applied to both the real sample y_1 and the generated output $G_1(Y_1) \rightarrow Y_2(y_2)$. The model then back-propagates the gradients through the augmentation when G_1 is updated. This ensures that the generator outputs distinct w.r.t. images [14]. *DiscriminatorF* aids G_1 to generate images that do not deviate from the corresponding shape in the portraits such that the original face structure is retained. The corresponding loss is calculated by the hinge loss method first, which supervises the margin boundary of the

real and fake images, detected by *DiscriminatorF* (equation 1).

Subsequently, both the cross-entropy loss (equation 2) and the hinge loss (equation 1) are calculated. The total loss of these augmented images is calculated as the mean of the two losses (equation 3). In these formulas, $D_f(x)$ refers to the *DiscriminatorF* which takes $x \in Aug(y_2)$ as input and outputs a scalar between 0 and 1. $G_1(z)$ refers to the generator that maps a sample $z \in y_2$ from the distributions Y_2 to the input space $Aug(y_2)$.

$$L_{D_f(hinge)} = max(0, 1 - D_f(x) * D_f(Aug(G_1(z))) \tag{1}$$

$$\begin{aligned} L_{D_f(BCE)} = -(D_f(x) * log(D_f(Aug(G_1(z))) \\ +(1 - D_f(x)) * log(1 - D_f(Aug(G_1(z)))) \end{aligned} \tag{2}$$

$$L_{D_f(total)} = (L_{D_f(hinge)} + L_{D_f(BCE)}) * \frac{1}{2} \tag{3}$$

The details of the second phase are displayed in Figure 3. The motivation behind using hinge loss, in this case, is to ensure that the gradient maintains a non-zero stability and to decrease vanishing gradients at fake samples [11]. The result is then scaled by a hyperparameter of $\lambda = 0.4$. This is to make certain that the value is not too close or larger than 1 and to regularize the value when calculating the total loss of G_1 and the corresponding gradient. This hyperparameter was chosen from the set of $\lambda = 0.1, 0.2, 0.3, ..., 0.9$. The effect of these 9 different values on the overall loss of the G_1 model was analyzed in training after 5 epochs.

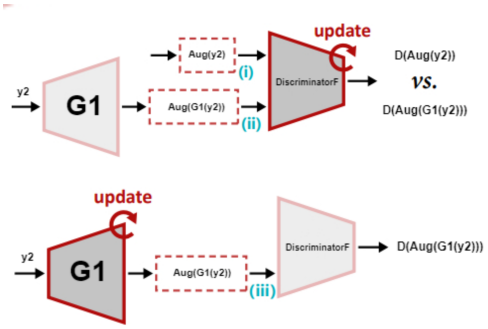

Figure 3: Pipeline of our framework in the second phase: we train a two-objective discriminator *DiscriminatorF* on the augmented images. The corresponding losses then aid in the final loss and gradient calculations of G_1 and *DiscriminatorP*.

Subsequently, $\lambda = 0.4$ was chosen since it resulted in the least loss of all the values tested. The reduced amount of epochs used for hyperparameter selection is due to time limitations. Afterwards, the loss that was obtained in equation 3 is added to the G_1 loss. The G_1 loss is then used in the calculations of the total CycleGAN loss (equation 8) to improve the performance of the model.

Lastly, the *DiscriminatorP* monitors the outputs of the two generators and distinguishes between "fake" and "real" images to enhance the performance of the two generators. Two adversarial losses (adversarial losses of pairs of the generator/discriminator for each domain), an identity loss, and a cycle-consistency loss were subsequently established. For the first adversarial loss, the mean between the the binary cross-entropy loss of G_1 and the loss from equation 3 was calculated (equation 4). For the second adversarial loss, the binary cross-entropy loss of G_2 was calculated to accomplish results closest to y_2 (equation 5).

$$\begin{aligned} L_{(G_1, D_{p_1}, y_1, y_2)} = ((log(D_{p_1}(y_2)) + \\ log(1 - D_{p_1}(G_1(y_2)))) + L_{D_f(total)}) * \lambda \end{aligned} \tag{4}$$

$$\begin{aligned} L_{(G_2, D_{p_2}, y_1, y_2)} = (log(D_{p_2}(y_1)) + \\ log(1 - D_{p_2}(G_2(y_1)))) * \lambda \end{aligned} \tag{5}$$

For the identity loss, the mean of the distance (the mean of the overall losses) between the real images and the cycled images for each domain was calculated (equation 6). For the cycle-consistency loss, the mean of the distance between the real images and the cycled images for each domain was calculated using equation 7. The final global loss is then the average across these four losses scaled by the hyperparameter $\lambda = 0.4$ (equation 8). In these formulas G_1 is the portrait to face generator, G_2 is the face to portrait generator, D_{p_1} is the face recognizer, D_{p_2} is the portrait recognizer, y_1 is the image of real faces and y_2 is the image of portrait paintings [15]. It is important to note that some sub-results are again scaled by the hyperparameter $\lambda = 0.4$ to ensure a better loss consistency.

$$L_{(identity)} = (|G_1(y_2) - y_2| + |G_2(y_1) - y_1|) * \lambda \tag{6}$$

$$\begin{aligned} L_{(cycle-consistency)} = (|G_2(G_1(y_2)) - y_1| + \\ |G_1(G_2(y_1)) - y_2|) * \lambda \end{aligned} \tag{7}$$

$$\begin{aligned} L_{GAN(total)} = (L_{(G_1, D_{p_1}, y_1, y_2)} + L_{(G_2, D_{p_2}, y_1, y_2)} \\ + L_{(identity)} + L_{(cycle-consistency)}) * \frac{1}{4} * \lambda \end{aligned} \tag{8}$$

3.2 Training Configuration

We used the Adam solver optimization [16] with a batch size of 1 for the training of the model. All

networks are trained from scratch with an initial learning rate of *2e-4*, and a polynomial decay strategy. The hyperparameter λ was set empirically and tuned in different steps to ensure consistent results. The weights of the model were saved at epochs 20, 40, and 90. The performance of the model was monitored using its total loss decrease.

3.3 Performance measures

3.3.1 Inception Score (IS)

The inception score [17] is a popular evaluation metric for generative models and aims to measure how closely the generator can simulate real inputs. This is an alternative approach to having a human grade the quality of the images. The score monitors how distinctively the image looks like the target value and the variation of the generated outputs. The final score obtained is between 0 and a high value.

If this outcome is high, it means that the generator can create many distinct images. For instance, in our case, the resulting faces are not too similar to another. From a mathematical point of view, the score is calculated as shown by equation 9. That is, measuring the KL-divergence between the two probability distributions p and q; p being the posterior probability of a label on image x that is produced by the generated from a z distribution, and q being the marginal class distribution [17]. It was decided not to calculate the IS score individually since the Frechet Inception Distance (FID) is more accurate. However, the *Inception V3 model* in keras [18] was used for calculating the FID score, which is explained in the next section.

$$IS_{score} = exp[E_{z\sim p}(z)[D(p(y|g(z)|p(y)]] \quad (9)$$

3.3.2 Frechet Inception Distance (FID)

Frechet Inception Distance is an improved evaluation approach to analyzing generative models. This score is calculated by the differences in the two Gaussian distributions of the results and the target (portrait paintings and real face images). From a mathematical point of view, FID uses the activation from the Inception v3 model [18] and summarizes them as a multivariate Gaussian distribution by calculating the mean and co-variance of the images. The FID then calculates the distance between these two distributions, which can also be referred to as the Wasserstein-2 distance [19].

The detailed calculations of FID [17] are displayed in equation 10 where Mu_1,Cov_1 and Mu_2,Cov_2 denote the mean and covariance of the true face images and generated images features respectively, $|Mu_1 - Mu_2|^2$ refers to the sum squared difference between the two mean vectors and T_r denotes the sum of elements on the main diagonal.

$$FID_{(score)} = (|Mu_1 - Mu_2|)^2 + T_r(Cov_1 + Cov_2 - 2 * \sqrt{(Cov_1 * Cov_2)}) \quad (10)$$

A lower FID score indicates that the images that were generated have better quality whereas a higher score indicates lower-quality images. Additionally, in our case, a lower FID scores indicates that the images are similar to the CelebA dataset.

4 Results

In this section, the FID score and a visualization of the generated images for each art style will be reported. For each art style, the model was tested with a total of 90 epochs. In addition, we used a simple cycleGAN [5] as a baseline for the performance of G_1 (the generator of this model will be referred as $BaseG$). The FID scores for this baseline model for the three test datasets of art styles together with the test dataset of the celebrity faces were calculated. The baseline model was also trained for 90 epochs.

4.1 FID Scores

Table 1 reports the FID scores that were obtained for each art style during testing. The results are also be visualized in Figure 4. As seen on the bar chart, the FID score is the lowest for the northern renaissance, and the highest for Ukiyo. Moreover, FID scores for all three styles are lower for $G1$ than for $BaseG$, which indicates a better performance.

Model	**A**rt style	**F**ID
G1	Northern renaissance	25.750954
G1	Post-impressionism	35.490105
G1	Ukiyo	36.507885
BaseG	Northern renaissance	38.619938
BaseG	Post-impressionism	39.68667
BaseG	Ukiyo	43.22856

Table 1: FID scores obtained for the art styles compared to the base line, with 90 epochs.

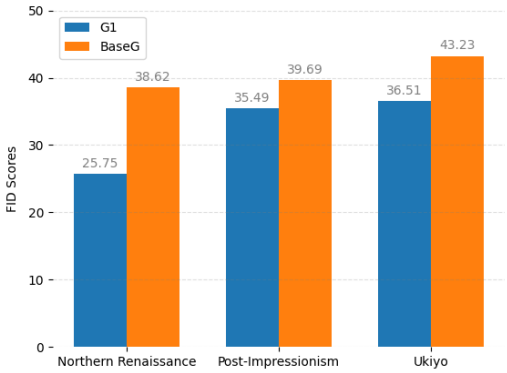

Figure 4: Bar chart of FID Scores by art style and model ($G1$ in blue, $BaseG$ in orange).

4.2 Generated Images

In this subsection, some example images that were generated from each art style will be showcased. Figure 5 shows the images that were generated for the northern renaissance art style. The images generated from post-impressionist portraits are displayed in Figure 6. The images that were generated from the ukiyo art style are displayed in Figure 7. In all of the figures, the original painting is shown on the left, and the generated image is shown on the right.

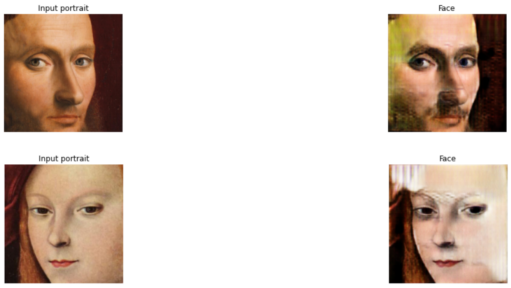

Figure 5: Images generated from northern renaissance portraits.

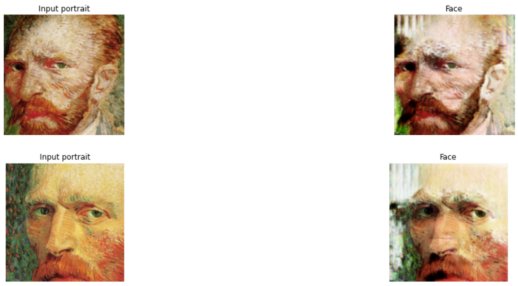

Figure 6: Images generated from post-impressionism portraits.

Figure 7: Images generated from ukiyo portaits.

4.3 Evaluation

For the evaluation, we will focus on the Frechet Inception Distance [18] (FID score) as a performance metric to assess the performance of our CycleGAN model. The FID scores for all three art styles can be found in Table 1. The value of the FID score assesses the quality of an image that was created by a generative adversarial model and it should be as low as possible to obtain the best results. Additionally, we will evaluate the visual outputs that are displayed in Figures 5, 6 and 7.

If we compare the FID score for the different art styles, we are able to observe that the Northern Renaissance art style has the lowest FID score of 25.75, which indicates that its generated images are the most realistic. If we look at Figure 5, we can indeed see some noticeable improvements. Firstly, the highlights that are created on points of the face such as the nose, under the eyes, and forehead are characteristics of a real-life image. Secondly, the eyes have more contrast in them through the darkened colors, which makes the eyes look more realistic. We can also conclude that the colors of the skin tone and hair have been adapted to look more like that of a photograph rather than a painting.

When evaluating the FID score of the post-impressionism art style, see Table 1, we can see that it is in between the northern renaissance and ukiyo, with a score of 35.49, with a score of 35.49. As post-impressionism is a less realistic art style than northern renaissance, this is to be expected. In Figure 6, it is clear that the output is an improvement on the original portrait, but it does not look as realistic as the northern renaissance output. One of the most notable improvements is the fact that the skin tone has a more natural color and the texture has been smoothed to a certain extent to remove any of the harshly painted lines. Additionally, the color and contrast of the eyes and hair are also more

image-like, as we saw was the case for the northern renaissance output.

Lastly, we will evaluate how the ukiyo art style affects the performance of our generative adversarial network. Its FID score is the highest out of all the art styles, around 36.51, which means that it is the worst score. This is also noticeable from the output in Figure 7. The one improvement that we notice is that the skin tone has more picture-like colors and highlights.

All of this points to the fact that our CycleGAN has an increased performance when converting realistic portraits rather than impressionistic or abstract portraits. We can make sense of this as the CycleGAN compares the portraits to the realistic images dataset and tries to recognize the most important facial features in order to convert the portraits to images. Since realistic portraits match more closely to a real-life image, the CycleGAN will have better results. On the other hand, it would be harder for the generative adversarial network to recognize these facial features in impressionistic and abstract portraits.

Furthermore, It is evident from the results in table 1 that the G_1 model has a better general performance than the $BaseG$ [5] in all three art styles. This is because of the optimizations that were applied - namely the addition of the two-objective discriminator 1 and the enhanced adversarial loss function 8. The G_1 model is then able to hide the information about the structure of face images inside the portrait paintings in a more efficient way than the $BaseG$ model.

5 Conclusion

In this work, we proposed an improved CycleGAN to convert portrait paintings to real face images. A model was developed to enhance the performance of the baseline CycleGAN [5] and reduce the distortion in the resulting face images. Subsequently, we introduced the use of the two-objective discriminator, the identity loss and the cycle-consistency loss to assist the generator in preserving the face structure and to ensure the resulted images keep the original portrait formation. The results of the experiment indicate that our CycleGAN performs better when converting portrait paintings to realistic faces than the simple CycleGAN used as baseline.

The research question of this paper was: *How does art style influence the performance of a CycleGAN that we designed to convert portraits into photo-like images?*. With regards to this, the analysis demonstrated that the performance of the model was most notable for northern renaissance, then post-impressionism, and lastly ukiyo. This conclusion is evident from the measured FID scores, which were 25.75, 35.49, and 36.51 respectively. These results are due to the fact that realistic portraits illustrate realistic facial features most clearly, which is why the performance of the CycleGAN increases for realistic art styles. This statement matches the hypothesis that we stated in the introduction to the report.

6 Discussion

In this section, the main limitations of the current study will be discussed and interesting areas of research for future works will be suggested.

6.1 Limitations

One of the main limitations of our research stems from the fact that we are dealing with a computationally heavy task, but our resources are limited both in terms of time and computational power. This resulted in the presence of some random noise in our generated images, which could have been minimized after a larger number of epochs. Although this issue was partially remediated through the use of Google Collab, we could have run more epochs or perhaps worked with bigger datasets in order to improve our results.

Another limitation (See also 2.3 Data Inspection) is the fact that our dataset was heavily biased. Most of the people in the portraits are white, which means our model will most likely perform poorly when dealing with images of people that belong to other races. This was hard to resolve as most historical paintings portray white people.

Lastly, one could argue that the evaluation of the visual output might be biased as it was only evaluated by the authors. Ideally, the evaluation would be improved by having unbiased observers provide feedback.

6.2 Future Work

An interesting avenue for future research would reflect on one of the limitations of our current study, namely, that it is biased. Our portrait dataset of

Northern Renaissance and post-impressionism consist primarily of white people. This is probably because they are fairly old western paintings and it was customary back then to mainly paint white people. It would be worth researching how using images from different races would affect the performance of our model. Another approach could be to try to resolve this problem for our current dataset by increasing the proportion of black people for example.

Lastly, due to the fact that image translating tasks are incredibly time-consuming and require high GPU usage we were not able to continue the training and obtain greater results. Thus, while the overall structure of the portrait faces is relatively maintained, the final real-like faces generated look distorted and can be distinguished from a realistic photo.

On the contrary, the $G2$ model shows promising improvement in the same stages of training such that the generated portraits are considered to be closer to a real portrait painting in both the face structure and the appearance of the image as a whole. The paired translations of $G2(Y1) \rightarrow Y2(y2)$ are displayed in the Appendix A.1 Figure 8. Additionally, the FID scores of $G1$ and $G2$ can be observed in Appendix A.2 Table 2. It would be interesting for future work to explore this conversion of image to portrait, instead of the portrait to image conversion we covered in this research paper.

References

[1] I. Castiglioni, L. Rundo, M. Codari, G. Di Leo, C. Salvatore, M. Interlenghi, F. Sardanelli, "AI applications to medical images: From machine learning to deep learning," *Physica Medica*, vol. 83, 9-24, 2021. doi: 10.1016/j.ejmp.2021.02.006

[2] N. H. Barnouti, S. S. Mahmood Al-dabbagh, W. E. Matti, "Face Recognition: A Literature Review," *International Journal of Applied Information Systems*, vol. 11, no. 4, pp. 21-31, 2016. doi: 10.5120/ijais2016451597

[3] I. Santos, L. Castro, N. Rodriguez-Fernandez, A. Torrente-Patino, A. Carballal, "Artificial neural networks and deep learning in the visual arts: A review," *Neural Computing and Applications,* vol. 33(1), 121-157, 2021. doi: 10.1007/s00521-020-05565-4

[4] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A. A. Bharath, *"Generative Adversarial Networks: An Overview," IEEE Signal Processing Magazine,* vol. 35, no. 1, pp. 53-65, Jan. 2018, doi: 10.1109/MSP.2017.2765202

[5] J. Zhu, T. Park, P. Isola, A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", *IEEE International Conference on Computer Vision (ICCV),* 2017, doi: 10.48550/arXiv.1703.10593

[6] "Secure Cloud Storage that protects your privacy," Sync. [Online]. Available: https://ln4.sync.com/dl/d7addacf0/b978wvm4-9dndxvh6-hc4ss39y-5hpck6si/view/default/11901442090008 [Accessed: 31-Mar-2022].

[7] S. Yang, P. Luo, C. C. Loy, and X. Tang, "From Facial Parts Responses to Face Detection: A Deep Learning Approach", in IEEE International Conference on Computer Vision (ICCV), 2015. doi: 10.1109/ICCV.2015.419

[8] J. Yang, "Portrait Painting Dataset For Different Movements", 2021, V1, doi: 10.17632/289kxpnp57.1

[9] J. Buolamwini and T. Gebru, "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification," *Proceedings of Machine Learning Research*, vol 81, no. 1, 2018. [Online serial]. Available: http://proceedings.mlr.press/v81/buolamwini18a.html. [Accessed March 17, 2022].

[10] A. Mahmoud, B. Michael, "What Else Can Fool Deep Learning? Addressing Color Constancy Errors on Deep Neural Network Performance," *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 2019, pp. 243-252. doi: 10.48550/arXiv.1912.06960

[11] N. Tu, L. Trung, V. Hung, P. Dinh, "Dual Discriminator Generative Adversarial Nets," *Advances*

in Neural Information Processing Systems 30 2017, arXiv:1709.03831. doi: 10.48550/arXiv.1709.03831

[12] X. Bing, W. Naiyan, C. Tianqi , L. Mu, "Empirical Evaluation of Rectified Activations in Convolutional Network," 2015, arXiv:1505.00853. doi: 10.48550/arXiv.1505.00853

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedial image segmentation," *Medical Image Computing and Computer-Assisted Intervention,* 2015, vol. 9351, pp. 234–241, arXiv:1505.0459, doi:10.1007/978-3-319-24574-4$_2$8

[14] J. Alexia, M. Ioannis, "Connections between Support Vector Machines, Wasserstein distance and gradient-penalty GANs," 2019, arXiv:1910.06922v1. doi: 10.48550/arXiv.1910.06922

[15] G. Ian, P.A. Jean, M. Mehdi,X. Bing, W.F. David,O. Sherjil, C. Aaron, B. Yoshua, "Generative Adversarial Nets," *Advances in Neural Information Processing Systems 27,"* 2014, arXiv:1406.2661. doi: 10.48550/arXiv.1406.2661

[16] D.P. Kingma, J. Ba, "Adam: A method for stochastic optimization," 2014, arXiv:1412.6980. doi: 10.48550/arXiv.1412.6980

[17] C. Min Jin, F. David, "Effectively Unbiased FID and Inception Score and Where to Find Them," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,"* 2020, pp. 6070-6079, arXiv:1911.07023. doi: 10.48550/arXiv.1911.07023

[18] S. Christian,V. Vincent,L. Sergey, S. Jon, W. Zbigniew, "Rethinking the Inception Architecture for Computer Vision," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,"* 2016, pp. 2818-2826, arXiv:1512.00567. doi: 10.1109/CVPR.2016.308.

[19] H. Martin, R. Hubert, U. Thomas, N. Bernhard, H. Sepp, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," *Advances in Neural Information Processing Systems 30 (NIPS 2017),* 2017, arXiv:1706.08500. Available:doi: 10.48550/arXiv.1706.08500

A Appendix

A.1 Image to Portrait Output

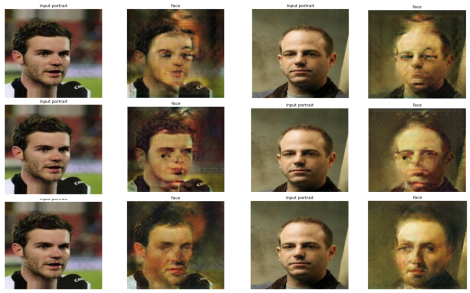

Figure 8: Two sets of celebrity faces (left) → portrait (right).

A.2 Table Comparing G1 and G2

Model	FID
G1 (20 epochs)	18.080236
G1 (40 epochs)	16.308403
G1 (90 epochs)	16.075895
G2 (20 epochs)	13.212725
G2 (40 epochs)	11.479938
G2 (90 epochs)	10.200821

Table 2: FID scores in different stages of training for both G_1 and G_2 generator models.