Project Report

Zhexu Li
Department of Electrical and Computer Engineering
University of California, San Diego
9500 Gilman Dr, San Diego, CA 92093
zhl411@ucsd.edu

Abstract

GANs are widely used for generating human photos, while the demand for fast generation of semi-realistic character portraits is strong in many industries, there is rarely any research existing for this topic. In this project I generated semi-realistic portraits for fantasy characters using StyleGAN on different datasets consists of a combination of real human photos and fictional portraits.

1 Introduction

1.1 Research problem aiming to solve

Generative Adversarial Networks (GAN) are widely used for generating human portraits, but most studies either focused on generating photo realistic human images using real photos, or creating anime characters using anime avatars. There are not many existing works about creating portraits of fantasy characters using both human photos and fictional fantasy portraits. This project aims to generate semi-realistic portraits of fictional characters using styleGAN, a novel derivative of GAN.

1.2 Importance of the problem

Generative Adversarial Networks, especially its derivatives (CycleGAN, StyleGAN, etc) are widely used for generating or transforming human images. With the rapid architecture advancements in recent years, GANs was quickly adopted by digital artists and "AI artists" [3], and have found its application in concept art [1], anime [2], gaming, and filming industry, and even in the blockchain [3]. There is a strong demand for semi-realistic fantasy character creation in those industries [1], but currently there are very few research in this area. A GAN which allows automatic and fast creation of semi-realistic fantasy characters could have big potential in multiple industries, and may even revolutionize the current digital art creation workflow.

1.3 Related Works

StyleGAN was first proposed by NVIDIA researchers Tero Karras et al. in 2018, and an improved version, StyleGAN 2, was later published by those researchers in 2020 [4][5]. It was built upon the Progressive Growing GAN architecture introduced by the same group of researchers (Tero Karras et al.) in 2017 [6], but with improved architecture especially for the generator, which not only overcomes the lack of control over style in traditional GAN models [9], but also allows generation of very high resolution images. Ever since its release, StyleGAN was widely used for creating all kinds of images, ranges from human faces to cats and even landscapes. A famous website ThisPersonDoesNotExist.com, which generates random fake human faces, was created by Uber researcher Phillip Wang using StyleGAN. The website quickly gained mainstream attraction because of its ability to generate shockingly realistic images. A recent interesting project carried out by Derek Philip Au utilized styleGAN to generate fake ceramic vessels which provides inspirations for ceramics professionals [8].

Overall, despite StyleGAN is widely used for generate many kinds of images, models are either trained using purely real human photos, or used for generating fake objects, and currently there is no existing research about generating fantasy character portraits using a combination of real human photos and semi-realistic portraits.

2 Description of Methods

2.1 Generative Adversarial Networks

Generative Adversarial Networks (GAN) was first introduced by Ian Goodfellow et al. in 2014, the main idea behind a GAN is that it consists of two Neural Networks: one Generator (G) and one Discriminator (D) [7]. The Generator Network G generates samples G(z) based on Gaussian noise z in order to simulate a latent distribution of input dataset. The Discriminator Network D learns to distinguish a generated sample G(z) from a real one. The goal of the training process can be described in the following form:

$$\min_G \max_D V(D,G) = \min_G \max_D \mathbb{E}_{x\sim p_{data(x)}}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1 - D(G(z)))]. \quad (1)$$

Which is basically a min max game between the Generator and the Discriminator, the Discriminator is trained to classify the generated images from the real images, while the Generator is trained to fool the Discriminator. In practice, this approach often suffers from Gradient Saturation, which the Generator network is quickly saturated and the Gradients are too small to continue the training. In order to overcome this problem, a modified non-saturating version of the loss function was introduced:

$$max_G L(G) = \mathbb{E}_{z\sim p_z(z)} logD(G(z)) \quad (2)$$

And this non-saturating GAN loss is often used in practice. But apart from vanishing gradients, the vanilla GAN architecture also suffers from many other problems, including mode collapse and difficulties in convergence. And more importantly, it can not control the style of the image generated, which makes it not ideal for our purpose.

2.2 Style GAN

StyleGAN is an improved architecture built upon style transfer literature and the structure of Progressive Growing GAN. It introduces a redesigned generator, inspired by the mapping network, which is capable of control the image synthesis process. The improved generator also overcomes the problem that the generator network is not able to create feature combinations that are missing from the training set, which persists in traditional GANs. To apply "styling" to the network, the pixel norm in the vanilla generator was replaced by adaptive instance normalization (AdaIN) [10], which is defined as:

$$AdaIN(x,y) = \sigma(y)(\frac{x-\mu(x)}{\sigma(x)}) + \mu(y) \quad (3)$$

AdaIN normalizes the input features using instance normalization $\frac{x-\mu(x)}{\sigma(x)}$, scales the normalized features by $\sigma(y)$, and shifts it with $\mu(y)$.

Mixing regulation is also employed to encourage the style to localize. The process of "style mixing" basically utilizes two latent codes to generate different proportions of the image. Stochastic variation is then introduced to generate style-based noise to the image in a local level. The overall generator network structure (figure 1) is much more complex than the traditional GAN generator structure, which makes it harder to train, but the results obtained by the generator are much better than the traditional one, since it addresses a lot of persisting problems in the traditional generator.

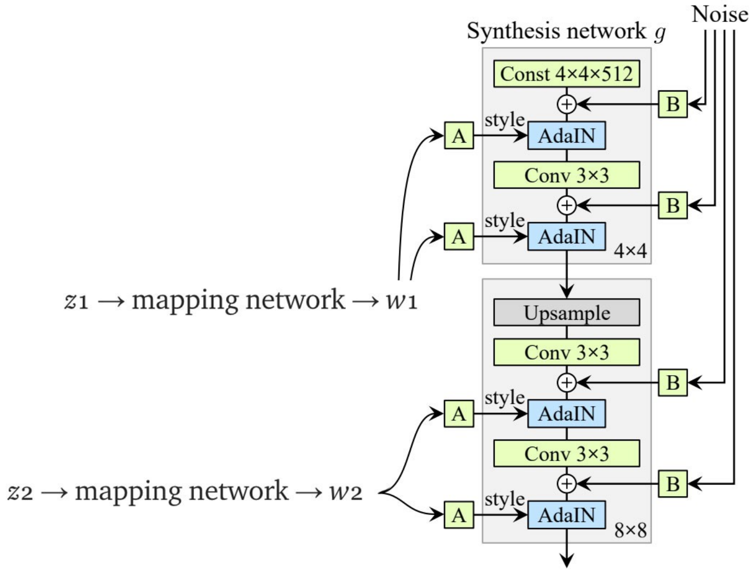

Figure 1: Structure of StyleGAN Generator, from the StyleGAN paper [4].

2.3 Truncation

It's very common in practice that certain areas that are underrepresented in the training set are usually difficult for the generator network to learn accurately. Truncation is a technique which limits the generator to draw latent vectors from a truncated space, and the process basically exchanges variation for better image quality. For StyleGAN, the truncation process is conducted in the latent space W:

$$W^{'} = \overline{W} + \psi(W - \overline{W}) \tag{4}$$

Where $W^{'}$ is the truncated latent space, $0 < \psi < 1$ is the "style scale". The truncation trick is very useful for generating samples after the training process.

3 Description of Approach

3.1 Raw Data Collection

The datasets I'm creating consist of high quality semi-realistic portraits of fictional characters and real human photos. The portraits are collected from popular art websites including Artstation, Pinterest, and DeviantArt. Because none of those websites provides accessible API, the data collection process is conducted using a a custom scraping scrpt modified from a popular opensource commandline tool gallery-dl. The program is now functional, but because of access limitations, it can only collect around 200 images per hour, especially in DeviantArt which has a strict access limit policy. The images are collected by categories based on characters or races.

3.2 Data Preprocessing

After collecting all the portraits, I will first filter out undesired images (non realistic sketches or anime portraits) and validate they are correctly categorized, by manually inspecting every image. This is a very time consuming process, and I have been trying to automate this process.

The StyleGAN has very strict requirement for the dataset. The input dataset should have the same format, the same size, and ideally the same color space. So after the initial filtering process, the images will be converted to the JPG format with RGB colors, which could save some space while preserving the most details. Because we only care about head portraits, I will crop portrait faces using a custom head detection algorithm derived from blogger ultraist's open source face detector for paintings, and crop photo faces using a OpenCV based face detection algorithm. The detector is not perfect, it has been producing a lot of false negatives, so there is still space to improve, but the process is totally automatic and the algorithm runs fast. Once the faces portraits are processed, they will be scaled to 512 x 512 px in order to meet the StyleGAN dataset requirement. After the rescaling, I will further filter out images with size < 40 kilobytes which indicates low quality. Then I will run a duplicate detection algorithm (based on image hashing techniques) to remove near duplicates.

3.3 Training and Evaluation

After preprocessing, I should have several datasets from different categories ready for training. Because the differences between fantasy characters and races (elf, orc, human, etc) are huge, in order to obtain the best results, I will train models seperately on every one of these categories. I will also apply transfer training, because the resulting dataset is often too small to train from scratch. In fact, based on my experiments, for every 20 raw images collected, there is usually only 1 valid for training after the preprocessing process. I will use StyleGAN - 2 for training, because it provides a number of improvements over the original StyleGAN, including the optimized AdaIN normalizer. One advantage of StyleGAN 2 is that it allows a variety of loss functions and optimizers to choose from. I will use the Adam optimizer, the non-saturated GAN loss with r1 regulation (which I discussed above in section 2) for training, because based on my experience this combination converges relatively fast, and usually provides the best result. Image augmentation will be included and adjusted based on the characteristics of the dataset. The training process will be conducted on Google Colab, which should provide a rather stable environment for training.

Once the model finished training after a certain amount of epoch, I will use the generator network with truncation factor (discussed in section 2.3) = 0.5 to create images. This truncation factor was selected because it seems to strike a balance between variety and quality, based on my experience. The resolution of generated images always equal to the resolution of the input images, which will be 512 x 512 px for this project. The generated images will be evaluated using Fréchet Inception Distance (FID) and Inception Score (IS), because these two metrics are the standard metrics for assessing the quality of GAN, according to [11]. Inception V3 model is used for the evaluation.

3.4 Novelty of Approach

This project utilizes a combination of real human faces and semi-realistic fantasy character face portraits to train a StyleGAN in order to generate high quality fantasy character faces, and currently there is no similar research existing. The dataset will contain newly collected samples which requires a lot of efforts to collect and process. The project will reveal how different mixture of real photos and semi-realistic portraits affects the performance of the StyleGAN, and how it behaves in different settings and topics, which may lead to some interesting discoveries.

3.5 Advantages and Limitations

One major advantage of my approach is that I have built an highly automatic pipeline for StyleGAN training. The pipeline requires very few inputs, and it's able to generate high quailty (512 x 512 px) semi-realistic fantasy character portraits. The training speed is relatively fast and for learning a certain domain it requires very few input images thanks to transfer learning, it took only 52 epoch to converge for the first proof of concept experiment which has a tiny training set of 58 images, and the results were promising.
On the other hand, though the data collection process is automatic, it could be quite slow depeding on the website it's scraping, and the data preprocessing filters out too many images which makes the resulting training set too small to train effectively. And the lack of data often leads to the lack of variation in the generated image, which was especially obvious in the first experiment. The

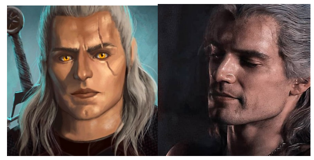

Figure 2: Sample training images for the first experiment, left is portrait, right is photo.

Figure 3: Training loss curves of generator and discriminator for the first experiment.

image augmentation process deployed to fix the lack of training set sometimes cause problems in learning. And while the StyleGAN generator network brought improvements over the traditional GAN generators, it's also way more complex, and require more computational resources to train [12].

4 Experiments

The experiments were carried out in 3 stages. In the first stage I conducted a proof of concept experiment using a small handpicked dataset of a single character. In the second stage I trained a styleGAN on a different character and with a medium sized dataset collected by the proposed pipeline with limited manual input. In the third stage I trained a StyleGAN on a fictional race instead of a specific character, and the whole process was conducted almost fully automatically, with minimal mannual input mainly for hyperparameter tuning.

4.1 First Experiment: Proof of Concept

The first experiment was a proof of concept experiment designed for validating the proposed pipeline. I collected around 200 raw images of Geralt of Rivia, a world famous fictional character from a popular Polish novel series "The Witcher". This character was selected because his portraits are widely available on the Internet, and it has adaptation movies which provides real human images. The image preprecessing yield only about 20 images (around 12%), and after some hand picking, the resulting dataset consists of 58 images (29 movies photo, 29 portraits), samples are shown in figure 2. Images were augmented through horizontal flipping and mirroring. The model was transform learned from an 512 x 512 px anime face model trained by Nagatomi, and trained for 52 epoch (52k images) until it was deemed converge based on the training loss curves shown in figure 3. The training process took about 4.5 hours using a Tesla T4 GPU on Google Colab, which is the standard setting for later experiments.

Images were then generated using truncation = 0.5, samples are shown in figure 4. The generated images achieven an IS of 1.6042732 evaluated on the Inception V3 model. FID was not evaluated for

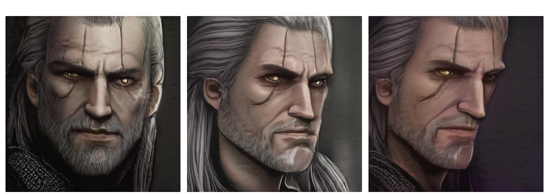

Figure 4: Sample images generated by the first experiment.

Figure 5: Sample training images for the second experiment, left is portrait, right is photo.

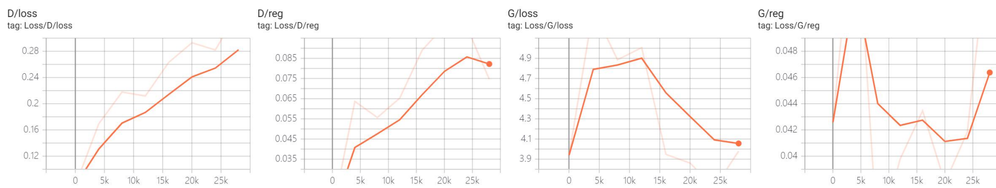

Figure 6: Training loss curves of generator and discriminator for the second experiment.

this experiment because it's not meaningful to evaluate FID on such a small dataset. And we can see the generated images captures some characteristics of Geralt pretty well, including his hairs, beard and scars. It also seems to achieve a good balance between portraits and photos. Interestingly, the scar which is supposed to appear on Geralt's left face appeared on both sides in the generated image, which was likely caused by the heavy image augmentation (mirroring). There was also noticeable artifacts in the generated eyes. And the variation of generated images were very limited, likely because of the small training set and the truncation setting. Overall, despite some limitations, the result was pretty impressive especially considering the extremely small size of the training set.

4.2 Second Experiment: Medium Sized Dataset

The second experiment was conducted on a medium sized dataset of Wonder Woman (Diana Prince), which is also a world famous fictional character with a lot of fan portraits and adaptation movies. I collect 1071 raw images of Wonder Woman from DeviantArt and Artstation using the scraper implemented, and the data preprocessing pipeline yield 129 images (12%). The number of filtered images was lower than expected, so I manually select 71 more images, and ran the deplicate detection algorithm to make sure there was no near duplicate. The resulting dataset contains 200 images (72 photo, 128 portraits) rescaled to a resolution of 512 x 512 px, samples are shown in figure 5. Images were augmented using techniques similar to the first experiment (mirroring, horizontal flipping) because the portrait of Wonder Woman is roughly symmetric. The model was then transfer learned from the same model mentioned in the first experiment to keep the consistency, and it mainly converged after 28 epoch (28k images) of training. The training loss curves are shown in figure 6. The training process took about 2.5 hours using the setting similar to the first experiment.

Image generation were carried out using truncation = 0.5, samples are shown in figure 7. The IS for generated images is 1.2491823, and the FID is 80.38091. Thanks to its bigger dataset (>3 times larger than the first experiment), the model converged way faster (28 epoch compared to 52 epoch), and has noticeably higher variety. The quality of generated images was quite impressive, the key features of the character (her crown, eyes and hair style) are well captured by the model. On the other

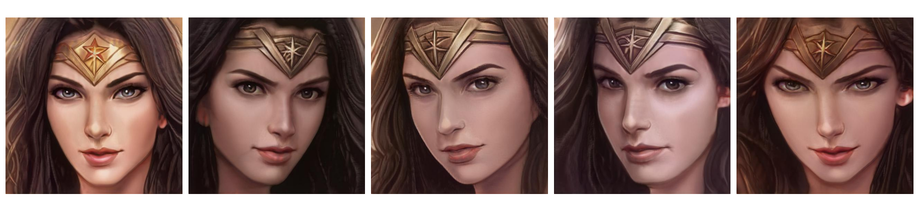

Figure 7: Sample images generated by the second experiment.

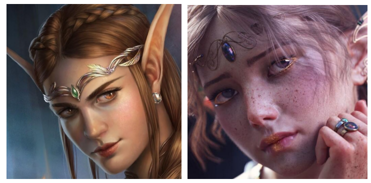

Figure 8: Sample training images for the third experiment, left is portrait, right is photo.

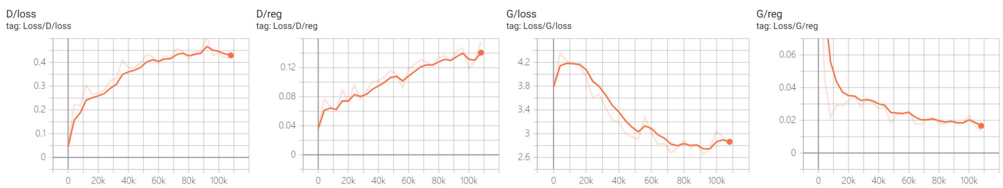

Figure 9: Training loss curves of generator and discriminator for the third experiment.

hand, we can see the images are more cartoon-ish compared to the first experiment, probably because the percentage of real photos was lower in this dataset (37% compared to 50%). The IS was also lower than the first model, possibly due to artifacts in her hair and crown which were hard to learn. Still, from a viewer's perspective, these images looks way better than the first experiment, and the proposed pipeline was working properly.

4.3 Third Experiment: Large Elves Dataset

The prior two experiments have proven the feasibility of my pipeline, so instead of focusing on specific characters, in this experiment I train a model on a fictional race: elf, a humanoid race very commonly seem in a lot of fantasy settings. Training for a race requires a lot more images than training for a specific character, so I collected 6777 raw images of elf scraped from DeviantArt, Pinterest, and Artstation. The image preprocessing pipeline yield 1212 cropped portraits, and in order to avoid manual input, there was no hand-picking, I directly used the images produced by the preprocessing. The resulting dataset contains 1212 images with 512 x 512 px resolution, samples are shown in figure 8. The proportion of real photos in the dataset was not verified. In order to maintain the consistency, the model was constructed using the same setting in the second experiment, with the same transfer learned model and the same augmentation techniques (mirroring and horizontal flipping). The model was trained for 108 epoch (108k images, 10 hours) until it roughly converged. The training loss curves are shown in figure 9, note the model seemed to converge at around 100 epoch (100k images), but I was not satisfied with the results and trained it for 8 epoch more, and stopped because no quality improvement was observed.
Images were generated using truncation = 0.5, samples are shown in figure 10. The generated images achieved an IS of 1.2523042, and a FID of 73.80367. While the dataset was way bigger than the first two experiments, it took longer training to converge (108 epoch), but this was also expected because learning a race is more complex than learning a single character. The resulting model captured iconic features of elves (pointy ear, small nose) and the quality of the images generated was impressive. The variety of the generator is way higher than the first and second model, and the IS and FID are also better. On the other hand, we observed more artifacts compared to earlier models, especially in hair and ear portion, likely because of the the high variety of these complex features. And interestingly, almost all images generated were female, probably due to the overwhelmingly large portion of female portraits in the dataset, and also the feminine natural of the elf race. Overall, this experiment proved that my proposed approach is capable of generating fantasy character portraits almost automatically using a combination of real human photos and semi-realistic portraits.

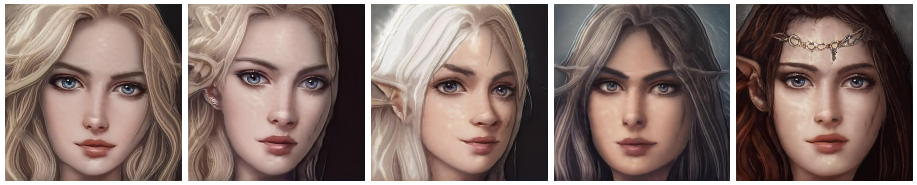

Figure 10: Sample images generated by the third experiment.

5 Conclusion

In conclusion, the above experiments have shown that my approach is capable of generating semi-realistic portraits of specific fantasy characters and fantasy races, with minimal manual input involved. The resulting models captured key features of the topic pretty well, and generally had good variety, thanks to the improved generator network of the StyleGAN. The generated images stroked a good balance between the photo-realistic style and fantasy portrait style, the quality of those images were amazing. The experiments also shown that the StyleGAN architecture benefited hugely from the increased size and variety of the dataset, with faster converge speed for training, and lesser artifacts in the generated images. And learning for a race was definitely a more complex task than learning for a specific character, requiring a significant more amount of data, and a longer training time.
On the other hand, the experiments also revealed some flaws in my pipeline. Though it was able to automatically collect and preprocess images, the speed of the process relied heavily on the website it was scraping, and the yield of preprocessing was too low (about 20%) which was very wasteful of the slowly scraped images. The open source face detection algorithm could be a big factor behind the low yield, it was not specifically designed for semi-realistic portraits in my dataset. A better and faster face detection algorithm adapted for my purpose could improve the yield significantly, but it was beyond the scope of this project.
Overall, despite some limitations, my experiments have proven the feasibility of my approach for its given task, and revealed some valuable insights about how the StyleGAN behaves using different settings. The quality of the resulting images was quite impressive.

5.1 Grace Day

Please be aware this report uses 3 grace days (submitted 3/20), Thanks!

6 References

[1] Max Schulz (2020). "Using GAN in CG: Concept Art Workflows". *80 LV*, `https://80.lv/articles/using-gan-in-cg-concept-art-workflows/`

[2] Yuanlue Zhu et al. (2021). "AniGAN: Style-Guided Generative Adversarial Networks for Unsupervised Anime Face Generation", *IEEE Transactions on Multimedia*. 2021.

[3] Brian Droitcour (2021). "GANs and NFTs". *Art in America*, `https://www.artnews.com/list/art-in-america/features/gans-and-nfts-1234594335`

[4] T. Karras, S. Laine and T. Aila (2018), "A Style-Based Generator Architecture for Generative Adversarial Networks" in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4217-4228, 2018. doi: 10.1109/TPAMI.2018.2970919

[5] Karras, Tero et al (2020). “Analyzing and Improving the Image Quality of StyleGAN.” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR) (2020): 8107-8116.

[6] Karras, Tero et al (2017). “Progressive Growing of GANs for Improved Quality, Stability, and Variation.” *ArXiv* abs/1710.10196 (2017): n. pag.

[7] Goodfellow, I. J. (2014, June 10). Generative Adversarial Networks. arXiv.Org. https://arxiv.org/abs/1406.2661

[8] Derek Philip Au (2021). "This Vessel Does Not Exist". https://thisvesseldoesnotexist.com/#/

[9] Jason Brownlee (2019). "A Gentle Introduction to StyleGAN the Style Generative Adversarial Network". *Machine Learning Mastery*. https://machinelearningmastery.com/introduction-to-style-generative-adversarial-network-stylegan/

[10] Huang, X. (2017, March 20). Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. arXiv.Org. https://arxiv.org/abs/1703.06868

[11] Wikipedia contributors. (2022, March 10). Fréchet inception distance. Wikipedia. https://en.wikipedia.org/wiki/Fr%C3%A9chet_inception_distance

[12] Park, S. (2022, January 7). What is StyleGAN? An overview of the key concepts of StyleGAN. Medium. https://medium.com/analytics-vidhya/what-is-stylegan-an-overview-of-the-key-concepts-of-stylegan-3c1031775fb