Anime Face Generation using Generative Adversial Networks in Deep Learning

Anjana M S, [1 a)] and Dr. Dhanya N M ,[2 b)]

[1] Masters in Artificial Intelligence, Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India

[2] Assistant Professor, Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India

[a)] Anjana M S: cb.en.p2aid20012@cb.students.amrita.edu
[b)]Dr. Dhanya N M: nm_dhanya@cb.amrita.edu

Article Info
Page Number: 335 – 342
Publication Issue:
Vol. 71 No. 3s (2022)

Article History
Article Received: 22 April 2022
Revised: 10 May 2022
Accepted: 15 June 2022
Publication: 19 July 2022

Abstract

Anime characters are used not only in books, but also in entertainment, awareness shows, video games, etc. In the recent times, there are many systems built for anime face generations. There are also various kinds of Artificial Intelligence approaches used to solve this, but the most famous ones are Generative Adversial Networks. The main reason is that the efficiency and performance of Generative Adversial Networks are found to be increasing as days go by. This paper is written to compare the quality of images, performance and efficiency and performance of two types of Generative Adversial Networks; namely Deep Convolutional Generative Adversial Network, and Style Generative Adversial Net-work2. Fréchet Inception Distance (FID) is taken as the evaluation metric for the systems implemented. For Deep Convolutional Generative Adversial Network, an FID score of 624.04 is found; whereas for Style Generative Adversial Net-work2, an FID score of 30.6 is found. From the resultant images and the Fréchet Inception Distance score, it is evident that Style Generative Adversial Network2 is the best model for anime face generation

Keywords: GAN, DCGAN, Style2GAN, Deep Learning, Machine Learning, Artificial Intelligence, Generation

INTRODUCTION

Deep Learning approaches are improving in efficiency and in speed very much. One of the main industries in which mankind is trying to bring the newly developed creativity that Deep learning systems can provide is the Comics and the Anime industry. Anime characters are used not only in books, but also in entertainment, awareness shows, video games, etc. This puts cartoonists at a huge pressure to create new kinds of characters every time. Thus, there is a need for generating different kinds of characters even for a single book. This is where Deep Learning, and Generative Adversial Networks in specific can help. The generator in the GAN can be trained to create fake characters, and the discriminator will give feedback for each image generated. There are various kinds of GANs through which this can be achieved. This project will be a comparative study on the various kinds of GANs and their performance to generate images. In the recent times, there are many systems built for anime face generations. There are also various kinds of Generative Adversial Networks that are used for this problem. The efficiency and performance of GANs are found to be increasing as days go by. This paper is written to compare the quality of images, performance and efficiency of two different types of GAN; namely Deep Convolutional GAN, StyleGAN2; using the anime face generation dataset. One of the main initial challenges faced to construct the system was the amount of computing power. The training time often took days, which is only possible with a supercomputer. This was corrected by connecting to Google Colaboratory instead of the physical system, and using the GPU runtime. Another one of the main challenges was the presence of outliers in the dataset. The outliers were present in the form of blurred images, and the ones in which the faces were not easily recognizable. This was corrected by removing the various outliers present in the original dataset, hereby improving the quality of the training dataset.

Caras et al.[10] improved the quality of the previously introduced StyleGAN and named it StyleGAN2 or Style GAN v2. This is found to be much more efficient than the previously developed StyleGAN, both in time and results generation. Jihye Back[11] used a StyleGAN2 to generate a cartoon face based on a human face. The model succeeds in creating a globally convincing image based on the image given. However, adjusting of layers is required to optimize the dataset. Radford et al.[3] introduced the DCGANs by combining CNNs with the traditional GANs. The results obtained in this process was found to be much better when com-pared to the traditional GANs. Bing Li et al.[6] proposed a new generator architecture to simultaneously transfer styles and transform local facial shapes into anime-like counterparts based on the style of a reference anime-face, while preserving the global structure of the source photo-face. Caras et al.[7] proposed a new GAN architecture called StyleGAN and used it to gen-erate anime faces. The results obtained were found to be far more convincing than those obtained from other kinds of GANs. Aswin et al.[16] used a Conditional GAN to evaluate the consequences of prepro-cessing methods and improve the model's ability to detect epiphytes in images that are produced as an output by Unmanned Aerial Vehicles (UAVs). Kumar et al.[17] used a Deep Convolutional GAN to generate synthetic data points and used the results to perform augmented data classification using a CNN. The usage of GAN proved to be much more efficient than using other models. Saradagi et al.[18] used evolutionary algorithms and processed the generator to classify the noise vectors, those which can be used to map a particular class without class labels. Patil et al.[19] used a GAN network for segmenting images of speed bumps. The resultant accuracy was found to be good, and showed the ability of GANs to help in the development of self-driving vehicles.

ARCHITECTURE OF SYSTEM

The architecture used for Deep Convolutional Generative Adversial Network is shown below.

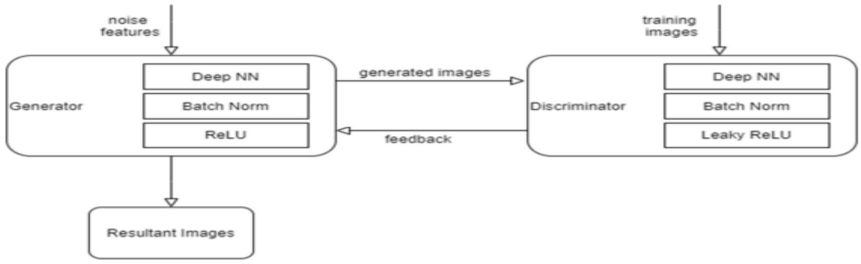

FIGURE 1: DCGAN Architecture

The architecture used for StyleGAN2is shown below.

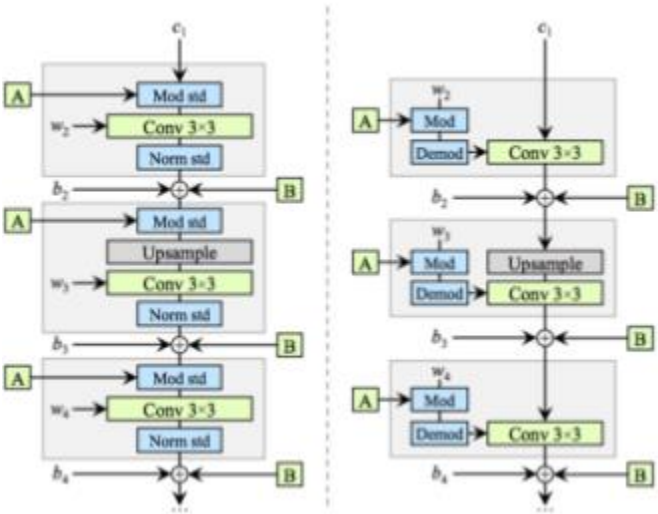

FIGURE 2: StyleGAN2 Architecture

DATASET

The dataset consists of high-quality anime faces scraped from www.getchu.com, and then cropped to fit only the face using lbp cascade algorithm. Images sizes vary from 90 * 90 to 120 * 120. For implementation purposes, the number of images used in the dataset is reduced to 20550. This number is obtained during the data cleaning stage while removing blurred pictures and erasing other corrupt files in the dataset.

PHASES

The following are the various phases in the process of generating new anime faces using GAN.

Phases for DCGAN

1. Data Cleaning and visualization

In this phase, the dataset is checked for any outliers, and the necessary action is taken. In the case of this dataset, the corrupt files and blurred images are removed. This makes the total number of images present in the dataset as 20550. A sample of around 20 images are visualized using the PIL, to check for further outliers.

2. Fixing Generator Architecture

The Generator is constructed with layers of CNN, Batch Normalization, and Leaky ReLU activation functions. The system constructed is then tested for errors.

3. Fixing Discriminator Architecture

The Discriminator is constructed with layers of CNN, Batch Normalization, and ReLU activation functions. The system constructed is then tested for errors.

4. Generating noise vectors

Noise vectors are used to create differences in the images generated by the generator. If they are not given, we would end up with multiple copies of the same image. Noise vectors are generated using the randint function, which is built-in in python.

5. Training

Training process is done for the whole system i.e., both the generator and the discriminator in this phase.

6. Output Visualization

The resultant images received are visualized using the Python imaging library, and the FID score, or the Fréchet inception distance is calculated for the generated images.

Phases for StyleGAN2

1. Data Cleaning and visualization

In this phase, the dataset is checked for any outliers, and the necessary action is taken. In the case of this dataset, the corrupt files and blurred images are removed. This makes the total number of images present in the dataset as 20550. A sample of around 20 images are visualized using the PIL, to check for further outliers.

2. Importing Generator and Discriminator architecture from NVDIA

The generator and discriminator architecture are taken from the NVDIA github repository. This is a public repository with instances for StyleGAN, StyleGAN2, Bayesian GAN, etc.

3. Seed Vector Generation

Seed vectors are used to initialize how similar all of the images generated should be. If the seed value is given to be low, images with similar features will be generated. On the other hand, if seed vectors are given to be large, images with less similar features will be generated.

4. Training

Training process is done for the whole system i.e., both the generator and the discriminator in this phase.

5. Output Visualization

The resultant images received are visualized using the Python imaging library, and the FID score, or the Fréchet inception distance is calculated for the generated images.

RESULTS

The results obtained for both the systems are discussed. The FID score, or the Fréchet Inception Distance score for each kind of GAN is tabulated below.

TABLE 1

Model	**FID Score**
DCGAN	624.04
StyleGAN2	30.6

At the end of the implementation of DCGAN and the generation of resultant images, the FID score is found to be 624.04, which is not a good score. Thus, we can conclude that DCGAN is not the most optimal system. A sample of the images present in the dataset is shown

FIGURE 3: Sample images in dataset

The resultant images that are obtained by the anime face generation GAN by using Deep Convolutional GAN is as follows.

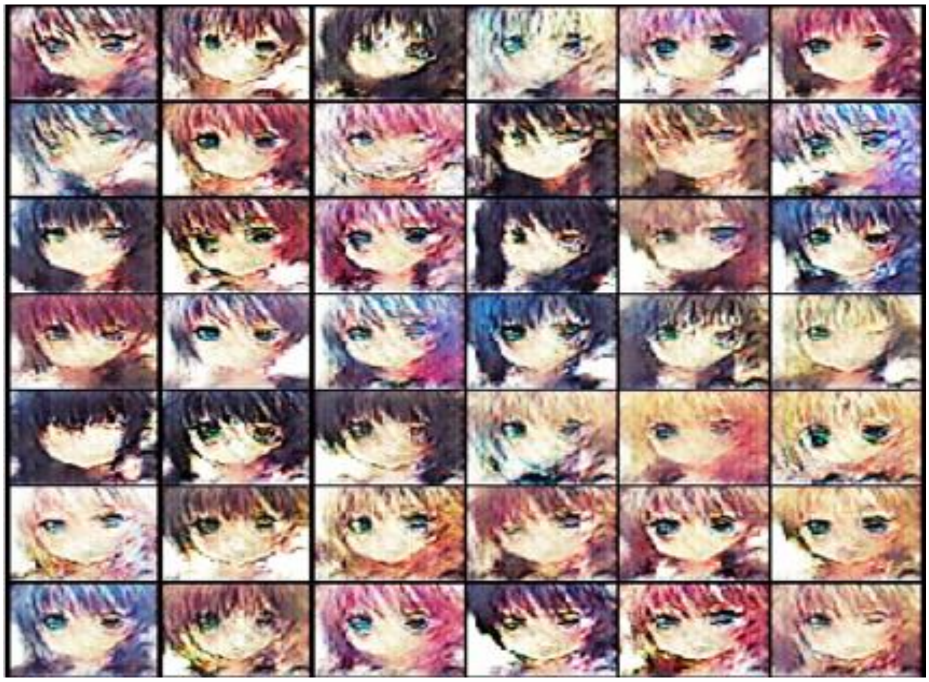

FIGURE 4: DCGAN resultant images

As it is shown in the above image, the resultant images are not globally convincing faces, as there are different eye colors, different eye shapes, and the whole image itself is fairly blurred.

At the end of the implementation of StyleGAN2 and the generation of resultant im-ages, the FID score is found to be 30.6, which is a good score. Thus, we can conclude that StyleGAN2 can be popularly used for anime face generation, as it is an optimal system.

The resultant images that are obtained by the anime face generation GAN by using StyleGAN2 is as follows.

FIGURE 5: StyleGAN2 resultant images

The above images are found to be much more globally convincing than the ones generated from DCGAN. Thus, StyleGAN2 is found to be an optimal system.

CONCLUSION

The most often used GANs for anime face generation is implemented and the result-ant images are compared. For DCGAN, In the resultant images, a few are found to be convincing and those images present different hair styles, hair colors or face orienta-tions. However, in some of the faces generated, there are different eye colors, eye shapes, etc; thus, making the results less globally convincing. FID score is found to be 624.04; which is a very high score. So, we can conclude that results will be guaranteed while using DCGANs, but the quality of the resultant images will be compromised. While coming to the StyleGAN2 architecture, the resultant images are found to be of high quality, with a smaller number of blurred images. The generated images are found to be globally convincing, with the features controlled by the seed values. The FID score is found to be 30.6, which is a very good score when compared to DCGAN. Thus, it is accepted that StyleGAN2 is the most optimal system among the GANs implemented. For further enhancements, Anycost GANs and StyleGAN3 can be implemented and further checked for their efficiency

REFERENCES

1. Towards the Automatic Anime Characters Creation with Generative Adversarial Networks by Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, Zhihao Fang in arXiv.org > cs > arXiv:1708.05509
2. Anycost GANs for Interactive Image Synthesis and Editing by Ji Lin, Richard Zhang, Frieder Ganz, Song Han, Jun-Yan Zhu in arXiv.org > cs > arXiv:2103.03243v1
3. Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.
4. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.
5. Yu-Jing Lin , Chiou-Shann Fuh. Generating Anime Faces From Human Faces With Ad-versial Networks. In csie.ntu.edu.tw
6. Bing Li, Yuanlue Zhu, Yitong Wang, Chia-Wen Lin, Bernard Ghanem, Linlin She. Ani-GAN: Style-Guided Generative Adversarial Networks for Unsupervised Anime Face Gen-eration in arXiv.org > cs > arXiv:2102.12593
7. Tero Karras, Samuli Laine, Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks in arXiv:1812.04948v3 [cs.ne], March 2019
8. Yuanbo Xiangli, Yubin Deng, Bo Dai, Chen Change Loy, Dahua Lin. Real Or Not Real, that is the Question in arXiv:2002.05512v1 [cs.LG] February 2020

9. Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks in arXiv:1703.10593v7 [cs.CV] August 2020

10. Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila. Ana-lyzing and Improving the Image Quality of StyleGAN in arXiv:1912.04958v2 [cs.CV] March 2020

11. Jihye Back. Fine-tuning StyleGAN2 for Cartoon Face Generation in arXiv:2106.12445v1 [cs.CV] June 2021

12. Sergei Belousov. MobileStyleGAN: A Lightweight Convolutional Neural Network for High-Fidelity Image Synthesis in arXiv:2104.04767v2 [cs.CV] Sep 2021

13. Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, Timo Aila. Alias-Free Generative Adversarial Networks in arXiv:2106.12423v4 [cs.CV] October 2021

14. Yuri Viazovetskyi, Vladimir Ivashkin, Evgeny Kashin. StyleGAN2 Distillation for Feed-forward Image Manipulation in arXiv:2003.03581v2 [cs.CV] October 2020

15. Rameen Abdal, Yipeng Qin, Peter Wonka. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? In arXiv:1904.03189v2 [cs.CV] September 2019

16. S. Aswin, V.V. Sajithvariyar, Ramesh Sivanpillai, V. Sowmya, Gregory K Brown, A. Shashank and K. P. Soman. Effect of Annotation and Loss Function on Epiphyte Identifica-tion using Conditional Generative Adversarial Network in International Conference on Ad-vances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 2021

17. Kumar K, Soumya V, Gopalakrishnan E A, Soman K P. Classificaiton of Class-Imbalanced Diabetic Retinopathy Images using Synthetic Data creation by Generative Models in Interna-tional Conference on Intelligent and Sustainable Systems, ICISS 2021

18. Ankit B Saradagi, Jeyakumar G. Evolutionary Algorithm based Encoder Decoder Network Design for Semantic Inpainting and Noise Vector Mapping in Generative Adversarial Net-work in Sixth International Conference on Inventive Computation Technologies [ICICT 2021]

19. Sandip Omprakash Patil, Sajith Variyar V V, and Soman K P. Speed Bump Segmentation an Application of Conditional Generative Adversarial Network for Self-driving Vehicles in Fourth International Conference on Computing Methodologies and Communication (ICCMC 2020)

20. Dataset: https://github.com/bchao1/Anime-Face-Dataset

21. https://github.com/NVlabs/stylegan2