Face Generation based on Race using Generative Adversarial Networks

P. N. Siva Jyothi [1], K. Kranthi Kumar[2], Pratisht Mathur[3], Mohammad Junaid[4], T. Vinay Reddy[5]

[1]Assistant Professor, Information Technology, Sreenidhi Institute of Science and Technology
[2]Associate Professor, Information Technology, Sreenidhi Institute of Science and Technology
[3-5]B.Tech Student, Information Technology, Sreenidhi Institute of Science and Technology

---

Abstract - *The field of image generation has seen immense amounts of improvement, with the advent of generative adversarial networks, an application of unsupervised learning. In this paper, the images are generated using Generative Adversarial Networks with the help of two sets of data, one containing the Asian faces and the other that holds Caucasian faces. Deep Convolutional Generative Adversarial Networks, that use convolutional neural networks for the generator, and the discriminator, have been utilized in order to accomplish the task of the generation due to advantages over conventional GANs such as less noise, greater stability of results.*

***Key Words*: Generative Adversarial Networks, Deep Convolutional Generative Adversarial Networks, Deep Learning.**

1. INTRODUCTION

Generative Adversarial Networks are a strong and effective approach to unsupervised learning. It is a deep learning-based generative model. This model finds its applications in various fields such as image synthesis and editing[1], audio generation[2], music generation[3], sound generation, predicting the next frames in videos[4,5,6], image to image translation[7,8], and also in academic areas such as medical imaging[9,10]. It also serves as an answer to various challenges faced in the field of semi-supervised learning [11, 12].

Generative Adversarial Networks contains two types of models: a generator and a discriminator, both of which are deep learning models. The generator tries to assume the distribution of the real data. It takes a fixed-length vector as input and then generates a sample. The discriminator, on the other hand, is generally a binary classifier, whose job is to classify data as real or fake. The real data is the data that belongs to the domain or distribution of the input data, and the fake data is the data that has been generated by the generator. In contrast to other generative models, Generative Adversarial Networks perform in an adversarial manner, a mechanism where the generator and the discriminator try to outperform each other. Training of the generator is done by maximizing the probability of the discriminator making a mistake. Optimization is based on a min-max optimization methodology. The termination of the process occurs when at a particular saddle point, that is minimum with respect to the generator and maximum with respect to the discriminator, thus reaching an equilibrium. Once the endpoint is reached, it is assumed that the generator has reached the distribution of the real data.

1.1 Deep Convolutional Generative Adversarial Networks

Deep Convolutional Generative Adversarial Networks (DCGANs) is an enhancement to the Generative Adversarial Networks. A DCGAN has two models, both of which are Convolutional Neural Networks, a generator, and a discriminator, one which generates data and the other, which checks the legitimacy of the data, whether real or synthetic. Similar to Generative Adversarial Networks, in DCGANs as well, the generator and the discriminator run in an adversarial manner, trying to better one another. It is an enhancement to GANs because the output generated is far superior with less noise, and the training is also stable and much more reliable. This gives DCGANs an upper hand over conventional GANs and has therefore been used in the project. The reason for the difference in the output comes because of the architecture that DCGANs have used. Modifying the conventional architecture of GANs, which uses multilayer perceptron in three layers, i.e., Convolution Layer, Pooling Layer, and Fully connected layer for the generator and the discriminator, DCGANs have replaced the pooling layers with strided convolutions in the generator and with fractional-strided convolutions in the discriminator. A process called Batch Normalization has been used to train the data in a more stable way and to avoid collapsing of the system. Fully connected layers are removed completely for deeper architectures, and activation functions such as ReLu and Leaky ReLu have been used in the Generator and Discriminator, which helps in better learning and provides higher resolution modeling.

2. RELATED WORKS

Image generation is a critical problem in the field of computer vision. Several advancements have been made with the help of deep learning models, for example, Variational Autoencoders[13], which help in formulating the problem with an approach that builds on probabilistic graphical models. In this, the lower bound maximization of the data takes place. Besides this, Autoregressive models, proposed by van den Oord et al. in the paper Pixel Recurrent Neural Networks 2016[14], based on modeling the conditional distribution of the pixel space, have been relatively successful generating synthetic images. Generative Adversarial Networks [15] have shown tremendous performance in the generation of synthetic images. Proposed by Goodfellow et al. 2014, this mechanism focused on image generation with the help of two deep learning models that compete with each other. For the generation of images, the use of convolutional neural networks as the model for the generator and the discriminator was also proposed [16]. Here, CNN, with certain special criteria, were employed so as to perform the task. Conditional image generation [17, 18, 19] was used in order to tackle problems such as the scaling of images, improving the stability of the generated images.

3. METHODOLOGY

In this paper, for the generation of images, Deep Convolutional Generative Adversarial Networks was used. It is an extension of the classic GAN, wherein the generator and the discriminator are Convolutional Neural Networks instead of a Multi-Layered Perceptron. The CNN uses batch normalization, has strided convolutions (discriminator) and fractional-strided convolutions (generator) in place of pooling layers. The use of ReLU activation in the generator for all layers except for the output layer, which uses Tanh and LeakyReLU activation in the discriminator for all layers, also happens [16].

3.1 Implementation

The first step in the implementation is the loading and pre-processing of the dataset. In this case, the pre-processing involved reduction in the size of the images to the dimensions of 96*96 pixels. After this, the models for the generator and the discriminator were created. These were designed based on the concepts used in DCGANs[16]. The main goal of the generator is the creation of a vector space that corresponds to the domain of the data used. Whereas, for the discriminator, the main goal is to maximize the probability of classifying the data correctly. For the optimization of both the components, loss functions are defined. After this, this training process begins. The training process occurs until the completion of all epochs. It must be made sure that the training occurs in a manner such that one component does not overpower the other. At the beginning of the training process, the sample generated looks like random noise, but as the number of epochs increases, the quality of the generations improves significantly. After this, the model execution occurs, and the images generated are saved.

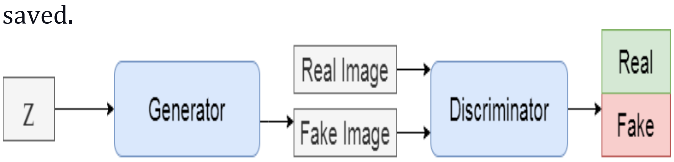

Fig -1: Generative Adversarial Networks

Reproduced from the paper 'How Generative Adversarial Networks and Their Variants Work: An Overview' **[20]** *z* denotes the latent space given to the generator

3.2 Dataset

Two distinct datasets are used in this paper. A dataset containing Asian faces with 804 images and another dataset that containing Caucasian faces with 380 distinct images were employed. The Caucasian faces dataset was obtained from UK Parliament Portraits by Chris McAndrew.

4. RESULTS AND DISCUSSION

The sample of images generated after the first iteration is not at all refined as they consist of random, as can be seen in Fig -2

Fig -2: Images Generated after first iteration

As the number of iterations increase, the quality of the output generated also increases. This can be observed in Fig -3

Fig -3: Images Generated after Seventeen iterations

In the above image, it can be observed that the sample generated is slowly taking the form of the input data distribution.

Fig -4: Images Generated after Fifty iterations

Above image shows the sample of images generated after fifty iterations. This shows the faces taking shape and slightly matching the input data.

5. CONCLUSION

This paper provides a way in which data that pertains to one racial group can be created with the help of Generative Adversarial Networks. It can also be concluded that the changes in the types of loss functions used can lead to better results. The quality and the quantity of the dataset plays a major role in getting a better result. Generative Adversarial Networks can be used in techniques such as data augmentation, which intern can be useful in the creation of larger dataset samples for research purposes.

ACKNOWLEDGEMENT

We would like to thank Dr. V.V.S.S.S Balaram, The Head of the Department, Information Technology, Sreenidhi Institute of Science and Technology for his valuable advice and technical guidance.

REFERENCES

[1] X. Wu, K. Xu, and P. Hall, "A survey of image synthesis and editing with generative adversarial networks," Tsinghua Science and Technology, vol. 22, no. 6, pp. 660–674, 2017.
[2] N. Torres-Reyes and S. Latifi, "Audio enhancement and synthesis using generative adversarial networks: A survey," International Journal of Computer Applications, vol. 182, no. 35, pp. 27–31, 2019.
[3] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. arXiv preprint arXiv:1705.10843, 2017.
[4] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. arXiv preprint arXiv:1707.04993, 2017.
[5] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In Advances In Neural Information Processing Systems, pages 613–621, 2016.
[6] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. arXiv preprint arXiv:1705.00053, 2017.
[7] Sagie Benaim and Lior Wolf. One-sided unsupervised domain mapping. arXiv preprint arXiv:1706.00826, 2017.
[8] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192, 2017.
[9] Wei Dai, Joseph Doyle, Xiaodan Liang, Hao Zhang, Nanqing Dong, Yuan Li, and Eric P Xing. Scan: Structure correcting adversarial network for chest x-rays organ segmentation. arXiv preprint arXiv:1703.08770, 2017.
[10] Morteza Mardani, Enhao Gong, Joseph Y Cheng, Shreyas Vasanawala, Greg Zaharchuk, Marcus Alley, Neil Thakur, Song Han, William Dally, John M Pauly, et al. Deep generative adversarial networks for compressed sensing automates mri. arXiv preprint arXiv:1706.00051, 2017.
[11] Chongxuan Li, Kun Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. arXiv preprint arXiv:1703.02291, 2017.
[12] Emily Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with contextconditional generative adversarial networks. arXiv preprint arXiv:1611.06430, 2016.
[13]P. Kingma and M. Welling. 2014. Auto-encoding variational bayes. ICLR (2014).
[14]van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. 2016. Pixel recurrent neural networks. ICML (2016).
[15]I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. Generative Adversarial Networks. NIPS (2014).
[16]A. Radford, L. Metz, and S. Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Network. ICLR (2016).
[17]T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Redford, and X. Chen. 2016.
Improved techniques for training GANs. NIPS (2016).
[18]A. Brock, J. Donahue, and K. Simonyan. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. ICLR (2019).
[19]X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang. 2018. Improving the Improved Training of Wasserstein GANs: A Consistency Term and Its Dual Effect. ICLR (2018).
[20]Hong, Yongjun and Hwang, Uiwon and Yoo, Jaeyoon and Yoon, Sungroh. 2019. How Generative Adversarial Networks and Their Variants Work. ACM Computing Surveys(2019).