Grenze International Journal of Engineering and Technology, June Issue

Text to Face Generation using DCGAN

Prof. Dr. Sandip Shinde[1], Atharva Joshi[2], Ansh Jaiswal[3], Sanskar Jain[4], Sahil Jain[5] and Mridul Sharma[6]
[1-6]Vishwakarma Institute of Technology, Department of Computer Science and Engineering, Pune 411037, Maharashtra, India
Email:{sandeep.shinde,atharva.joshi202,ansh.jaiswal20,sanskar.jain20,sahil.jain20,mridul.sharma20}@vit.edu

***Abstract*—The difficult issue of text-to-face (TTF) synthesis holds enormous promise for numerous computer vision applications. Due to the multiplicity of facial features and the parsing of high dimensional abstract natural language, textual descriptions of faces can be far more intricate and detailed than Text-to-Image (TTI) synthesis jobs. Due to lack of dataset, research work conducted on models performance in text to face generation is quite limited. In this paper We have proposed a DCGAN Text-To-Face model which produces images with 256x256 resolution. Experimental results show that the DCGAN model is able to generate high quality images using multiple layers of deep convolutional neural networks.**

***Index Terms*— GAN's, DCGAN, Machine Learning, Text-to-Image, Text-to-Face.**

I. Introduction

Generative Adversarial Networks are the type of neural network which learns the correlation of the input data to output and then tries to generate new data based on its training. There are multiple applications of GAN's such as forecasting data, generating new data for datasets, image colorization, increasing image resolution etc. One of the most used architectures of this neural network is called Deep Convolutional Generative Adversarial Network or DCGAN which is generally used for the task of Text-to-Image translation. In this project we implemented DCGAN for text-to-face translation and tried to generate realistic outputs.

II. Dataset

The dataset that we referred to is named as CelebA Dataset, which was available on kaggle. The dataset was created by authors of [17]. The dataset contains over 200k images of celebrities with 40 binary attribute annotations per image like "5 O'clock shadow", "Arched eyebrows", "Attractive",etc.

III. Literature Review

T2F is a subdomain of T2I and is still a challenging topic due to variation and complexities of faces. This challenge can be solved with the help of GAN. Since the introduction of the Generative Adversarial Nets architecture [1] there has been an increasing interest in the research in this field. Most of the work in this area of research lies within the domain of Text-to-Image as there exists state of the art architectures such as the DALL-E 2 [2] which uses CLIP latent for the image generation and produces astonishing results. One of the most prominent papers [3] in this subdomain refers to the usage of BERT [4] as the sentence encoder which we have implemented in this project, although we have used a pre-trained version from the HuggingFace Model Hub.

Grenze ID: 01.GIJET.9.2.42_1

The only publicly available dataset for the task is the CelebA Attributes dataset [5] which was released in 2016 and is widely used in various papers till date. The preprocessing algorithm which converts the 40 different binary attributes into N different captions was described in [6] and is also used in this project. We have limited the value of N=5 for the number of captions. The architecture of the project is inspired from [7] but we have changed the output resolution from 64x64 as described in the paper to 128x128 for our project. We've also changed the kernel size to (4,4) in our work. A generative adversarial proposed by Yijun Li et. al. [8] to fill the removed part in a picture of a face, consists of 1 generator and 2 discriminators viz. Local discriminator and global discriminator. The generator generates the part to be filled and discriminators use adversarial loss functions.This model architecture can not only fill the gap in the photo but also make it look realistic enough to not get caught when not paying close attention to minute details.Also since there is a lot of data available, it is difficult to classify images with high precision and accuracy. Earlier computer vision was used to define image generations, but later convolutional neural networks came into existence. CNNs are used for feature detection. Dataset acquisition was a big time-consuming problem and then came GAN.A simple GAN approach can be used for generating images from textual descriptions of flowers, or simple structured objects[9]. The provided sentence can be converted into tokens using natural language processing, which will be used as the input after processing. However for complex structures like faces a different architecture would be needed since a simple GAN architecture would not be able to learn accurately or completely.There also exists style based generators or style-GAN[10] can be considered as an improved GAN architecture which has better results than simple GAN. Two new metrics namely perceptual path length and linear separability can be used for estimating degree of latent space disentanglement. In recent times, image_segmentation has been involved everywhere including disease diagnosis to autonomous vehicle driving.[11]In computer vision, this image segmentation is one of the most important tasks, but it is relatively more complex than other vision tasks because it requires low-level spatial data. Deep learning, in particular, has had an incredible impact on the field of segmentation, offering a variety of successful models today. Deep learning-related generated adversarial networks (GANs) have shown remarkable results in image segmentation. In this study, the authors presented a systematic review analysis of recent publications of GAN models and their applications.[12]Generative Adversarial Networks (GANs) have become a research target in artificial intelligence. Inspired by two-player zero-sum games, GAN includes generators and discriminators trained using the idea of adversarial learning. The goal of GANs is to estimate the potential distribution of real data samples and generate new samples from that distribution. Since its inception, GANs have been extensively researched due to their great potential for applications including image and image processing, speech and language processing, etc.[13]explores the potential of GANs to create realistic images[14].The framework uses a high resolution stylegan2 face generator. BERT embedding is used for text embeddings provided as input to the styleGAN2 generator. The model outperforms other models in producing images that are closer to the ground. The model can generate images that are almost 57% closer to the real image. From both the quantitative and qualitative evaluative comparisons, the generated images exhibit good image quality and consistency with the input descriptions. [15]Deep convolutional generative Adversarial Networks (DCGAN) is used to generate new images that are not in the dataset. The DCGAN is a combination of two deep artificial neural networks called the Generator and Discriminator. for increasing efficiency, authors makes use of batch normalization of layers and parametric RELU having leaky RELU as activation function. This results into a sigmoid function for the two neural networks used respectively. This method is not optimized for video and can be used only for images. The Discriminator network uses conditions to generate the corresponding face and arrive at information that shall act as base values in determining synthetic images. Thus imposing on the generator to discriminate input images and generated images until they are real. This method uses two sub-encoders to separate distinct features of the human face. If the number of epochs are increased and neural networks are optimized better results can be achieved.Making images from text descriptions has become a popular area of research. Main goal is to make images that relate to the text description provided. Therefore, it can be concluded that with a larger data set and exposure to more facial attributes, this approach would produce even better results.

IV. METHODOLOGY

The dataset used for the task is the Celeb Faces Attribute Celeb-A dataset. It contains more than 2 lakh images of various celebrities with 40 binary attributes as annotations. These attributes are then processed with a custom algorithm to generate 5 different captions for a particular image. For preprocessing we have resized the image data into 128x128 resolution and normalized each pixel with mean and standard deviation of 0.5.
Since the dataset is quite large and due to the computational complexity and limited hardware, we decided to

take a small subset of 20,000 images which was balanced based on the binary attributes of the dataset. For text embeddings we used *all-mpnet-base-v2* pretrained Bert model which converted the sentence string into 786 dimensional semantic vector. We also introduced 100-dimensional noise inside the flow of training to avoid any overfitting issues in the model. Once we have all the data ready, using the data loader we feed the text embeddings and the noise to the generator to create fake images based on the training of it. We then backpropagate the loss and train the generator based on three loss functions. L1 loss, L2 loss and Binary cross entropy loss.

$$Total\_Loss = bce\_loss + 100*L2\_loss + 50*L1\_loss \quad (1)$$

Binary cross entropy loss is a commonly used loss function in binary classification problems. It measures the difference between the predicted probabilities and the true labels, and penalizes the model for making incorrect predictions. It is calculated as the negative logarithm of the predicted probability of the true label. L1 loss, also known as mean absolute error (MAE), measures the absolute difference between the predicted and true values. L2 loss, also known as mean squared error (MSE), measures the squared difference between the predicted and true values. Since L2 loss penalizes larger errors more strongly than L1 loss, we have kept more weights to L2 loss than L1 loss. The fake images along with their text embeddings are passed to the discriminator followed by a batch of real images and their counterparts. The discriminator is just trained using BCE loss, but the real, wrong, and fake loss are merged into a single total loss. We then trained the discriminator and generator for around 50 epochs with the training time of around 12 hours.

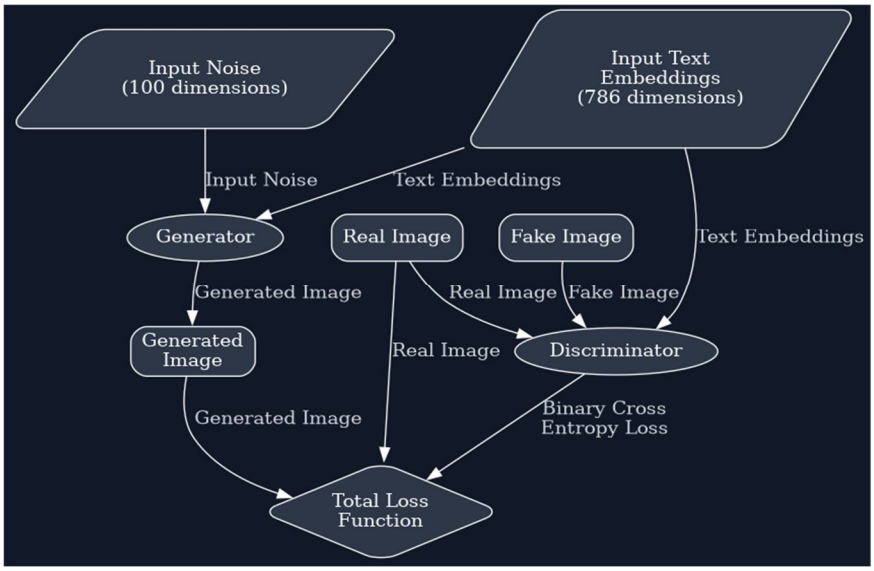

Fig 1. Flow diagram of the entire project

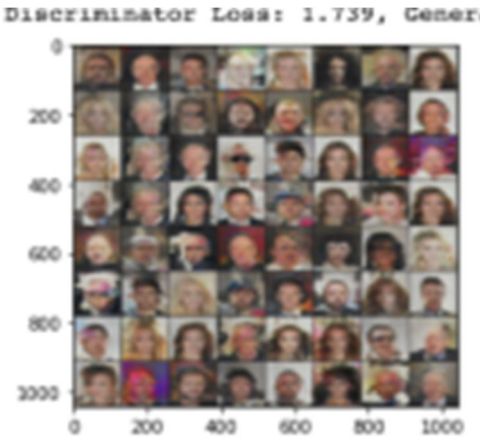

Fig 2. Training Process

V. System Architecture

The generator of the DCGAN consists of two different components: the text encoder and the up-sampling block. The text encoder consists of a Linear layer followed by Batch Normalization and Leaky ReLU. Its task is to project the 786 dimensions of text embeddings into 256 dimensions. Next, the up-sampling block consists of several *Convolutional Transpose-BatchNorm LeakyReLU* blocks which up-samples the entire vector into a 128x128x3 tensor. The last block is followed by tanh activation to get data in the range of -1 to 1.

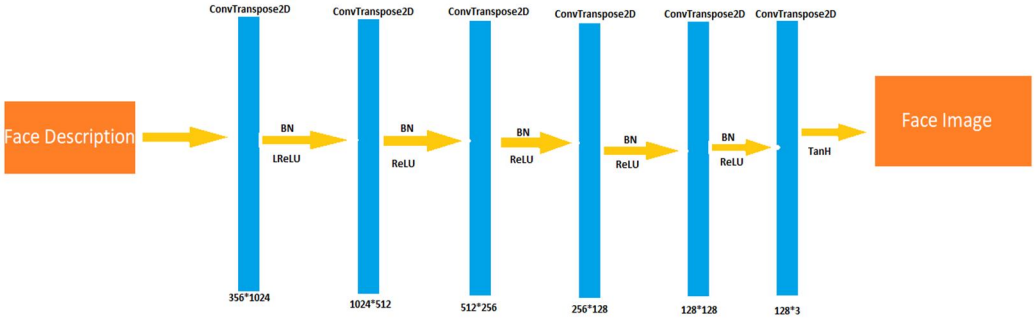

Fig 3. Architecture of Generator

The discriminator consists of three components: Image encoder, Text encoder and concatenation block. The image encoder consists of several blocks of Convolutional 2d layers which is generally followed by BatchNorm and LeakyReLU. The text encoder is similar to that of the generator which consists of Linear-BatchNorm-LeakyReLU block. The concatenation layer merges both text and image projections onto a single block. It consists of a Conv2d layer followed by sigmoid activation to get outputs in the range of 0 to 1.

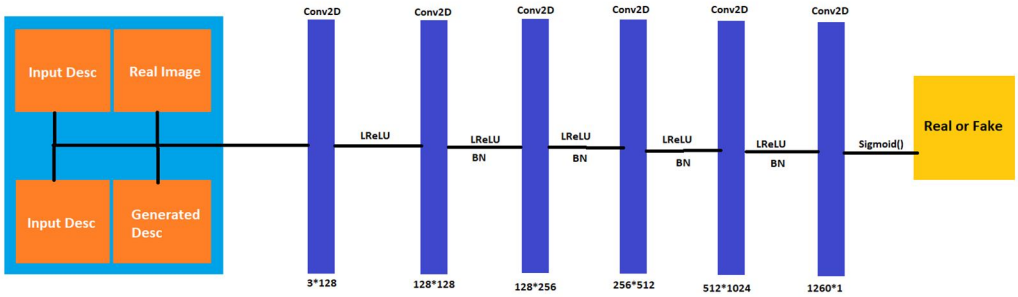

Fig 4. Architecture of Discriminator

VI. Results and Discussion

After both the models were trained for about 50 epochs, the loss of the generator decreased from around 25.926 to 21.094. Similarly, the discriminator total loss decreased from 1.67 to 1.576. As we can see the loss plateaued after about 30 epochs, but we kept training it further for some time. After inferring some number of outputs from the generator we can conclude that the model is somewhat biased towards female faces, maybe due to anomalies in the subset chosen. We can further implement better techniques to balance the gender for faces.

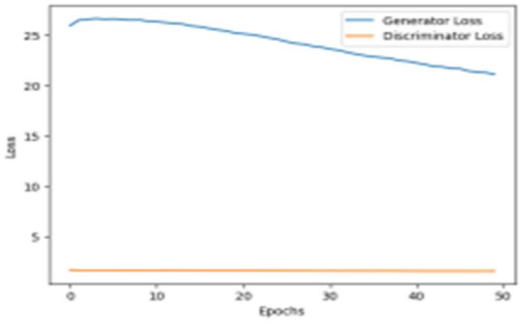

Fig 5. Loss values of both generator and discriminator

After checking the model with different sample inputs, we get the following results for 10 epochs ,20 epochs ,30 epochs ,40 epochs and 50 epochs respectively. Also we achieved an FID score of 90.47
Caption: The female has pretty high cheekbones and an oval face. She has brown hair. She has arched eyebrows and a pointy nose. She is smiling, seems attractive, young, has rosy cheeks and heavy makeup. She is wearing earrings and lipstick.

Fig. 6. Original image from dataset

Fig 7. Result after 10 epochs for caption1

Fig 8. Result after 20 epochs for caption1

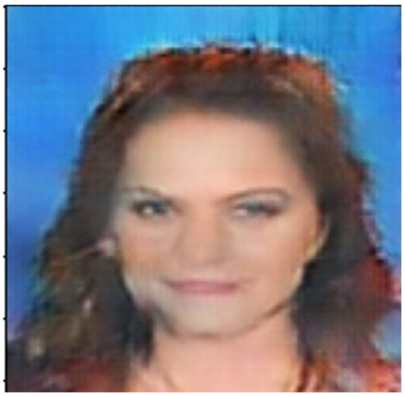

Fig 9. Result after 30 epochs for caption1

Fig 10. Result after 40 epochs for caption1

Fig 11. Result after 50 epochs for caption1

VII. CONCLUSION

Thus, in the project we successfully trained a DCGAN to successfully map the respective textual description to its respective face. Although the subset chosen was slightly biased towards female faces the results obtained

were decent. We have created a dynamic interface to interact with the model and get results instantly. In addition, we use the Fréchet Initial Distance (FID) score to measure the similarity between the generated image and the real image. In general, lower the FID score, greater is the similarity.Typically, a good FID score lies between 50 and 100. We achieved an FID score of 90.47, which means that the generated images are more similar to the real ones, but there is still room for improvement.

FUTURE SCOPE

The Inception score of our model is around 2.728 which signifies the quality of generated images is quite low. So, this project can be further implemented with different architectures such as StackGAN, AttnGAN etc. to get better and higher resolution outputs.
As discussed earlier we took a subset of around 20,000 images from the entire dataset which is merely 10% of the entirety, we can train it using the entire dataset to get better outputs. Lastly, we can introduce dropouts inside the network and analyze the results.

REFERENCES

[1] J. Goodfellow et al., "Generative adversarial nets," Adv. Neural Inf. Process. Syst., vol. 3, no. January, pp. 2672–2680, 2014.
[2] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol "Hierarchical Text-Conditional Image Generation with CLIP Latents" arXiv:2204.06125v1 [cs.CV] 13 Apr 2022
[3] D. M. A. Ayanthi And Sarasi Munasinghe "Text-To-Face Generation with Stylegan2" David C. Wyld Et Al. (Eds) Fcst, Cmit, Se, Sipm, Saim, Snlp – 2022
[4] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, no. Mlm, pp. 4171–4186, 2019.
[5] Ziwei Liu Ping Luo Xiaogang Wang Xiaoou Tang Large-scale CelebFaces Attributes (CelebA) Dataset Multimedia Laboratory, The Chinese University of Hong Kong
[6] Osaid Rehman Nasir, Shailesh Kumar Jha, Manraj Singh Grover, Yi Yu, Ajit Kumar, Rajiv Ratn Shah "Text2FaceGAN: Face Generation from Fine Grained Textual Descriptions" arXiv:1911.11378
[7] Alec Radford & Luke Metz "UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS" Arxiv:1511.06434v2 [Cs.LG] 7 Jan 2016
[8] Yijun Li et. al. Generative Face Completion
[9] Singh, Akanksha & Anekar, Sonam & Shenoy, Ritika & Patil, Sainath. (2022). Text to Image using Deep Learning. International Journal of Engineering and Technical Research.
[10] Tero Karras, Samuli Laine, Timo aila. A Style-Based Generator Architecture for Generative Adversarial Networks.arXiv:1812.04948v3 [cs.NE] 29 Mar 2019
[11] Alankrita Aggarwal, Mamta Mittal, Gopi Battineni (2020). Generative adversarial network: An overview of theory and applications.
[12] Kunfeng Wang, Chao Gou, Yanjie Duan, Lin Yilun (2017). Generative Adversarial Networks: Introduction and Outlook.
[13] David C. Wyld et al. (Eds): FCST, CMIT, SE, SIPM, SAIM, SNLP - 2022 pp. 49-64, 2022. CS & IT - CSCP 2022
[14] Ayanthi, Akila & Munasinghe, Sarasi. (2022). Text-to-Face Generation with StyleGAN2. 49-64. 10.5121/csit.2022.120805.
[15] Palaniappan, Devaki. (2020). Face Generation using Deep Convolutional Generative Adversarial Neural Network. Bioscience Biotechnology Research Communications.13.20-23.10.21786/bbrc/13.1/5.
[16] Ayanthi, Akila & Munasinghe, Sarasi. (2022). Text-to-Face Generation with StyleGAN2. 49-64. 10.5121/csit.2022.120805.
[17] 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.