Image Generation using Text Captions

Aryan Madaan 2020A1PS0222P

9th April 2023

1 Introduction

StyleGANs have been the foundation of modern text-to-face models for generating false images, and these models also include textual encoders that project textual descriptions into latent space. The Fréchet inception distance (FID) of the style-based generator is approximately 20% greater than that of the conventional generator [1], the StyleGAN model used is from the paper [2] and can generate high quality 1024 px * 1024 px images given enough training time.The codes that have been used were edited from this repository: Github Link

Text to image converters use artificial intelligence to convert natural language descriptions into images. AnyFace [3] is a model for generating images of any face and its occlusion from textual descriptions. Using a combination of generative adversarial networks and variational autoencoders, TediGAN [4] creates diverse and manipulable images from text descriptions. Both publications demonstrate the potential for text-to-image generation technology in a variety of industries, such as virtual and augmented reality, content creation, and the generation of training data for machine learning models.

2 Data Pre-Processing

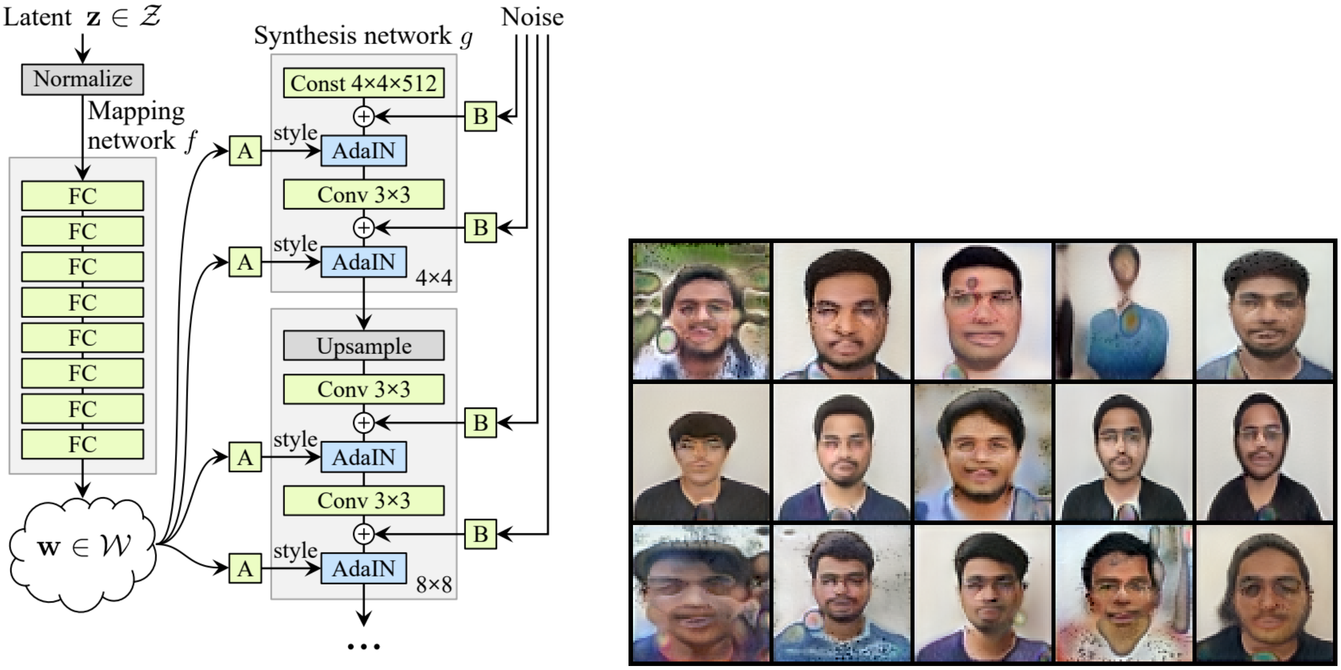
The data set provided to us contains 700 images with different extensions (".png",".jpeg",".jpg") with their multi caption-descriptions. These images were resized to 256 px * 256 px to reduce computational load and also all of them were converted to ".jpg". All captions were one hot encoded to create vectors. Using this vectors, multi-line textual descriptions were generator, but later not used. They can be used to further improve the model to generate images from multi-line text. Example of caption generated :- A middle aged male with a diamond shaped face and round jaw line. He has heavy upper black coloured lips and a narrow forehead. He has a wavy nose, square ear, his skin colour is brown and he has dimples on his cheeks. He has medium sized gray coloured eyes with a defect in the right eye. He has sharp and joint eyebrows. His face is shaved and he is half bald.

3 Models Developed

3.1 StyleGAN

Explanation of StyleGAN model :- At each convolution layer, adaptive instance normalisation (AdaIN) regulates the generator. After each convolution, Gaussian noise is added before evaluating the nonlinearity. Here "A" represents a learned affine transform, whereas "B" applies learned per-channel scaling factors to the noise input. The mapping network f has eight layers, while the synthesis network g has eighteen layers, two for each resolution. The final layer's output is converted to RGB using a distinct 1 1 convolution. The StyleGAN generator was trained on the complete dataset to generate 64 px * 64 px images due to computational limitations. There all the images were again resized to 64 px * 64 px. Only

20000 training iterations could be completed given the time and the resource constraint. The output of the StyleGAN is presented below :-



(a) StyleGAN Model [2]

(b) Output of StyleGAN

Figure 1: Explanation of StyleGAN layers and its output

As show below the GAN has started to recognize the features of the faces like the face shape and hair, but still fails to generate eyes, ears and nose properly.

3.2 The Perceptual Model

VGG16 is the Perceptual Model used in our circumstance. We utilised this model to generate the GAN's latent space (Input to GAN to produce an Image). First, train the VGG model to output the correct latent space for image generation with correct features. I was only able to train the VGG model on the first 50 images for 5000 iterations due to computational limitations. Due to fewer training iterations, the output of images produced after training is subpar.

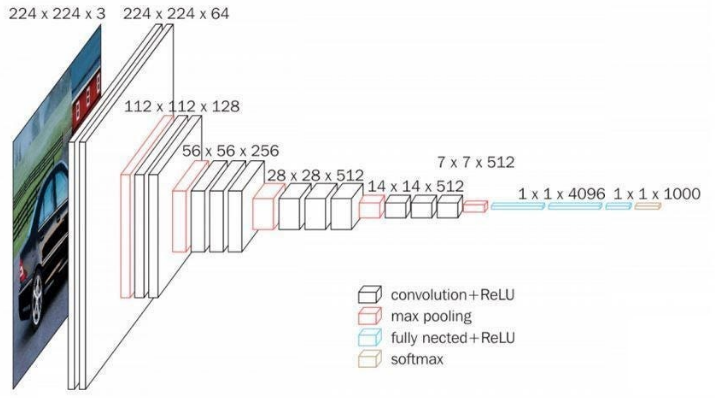


Figure 2: VGG16 Architecture

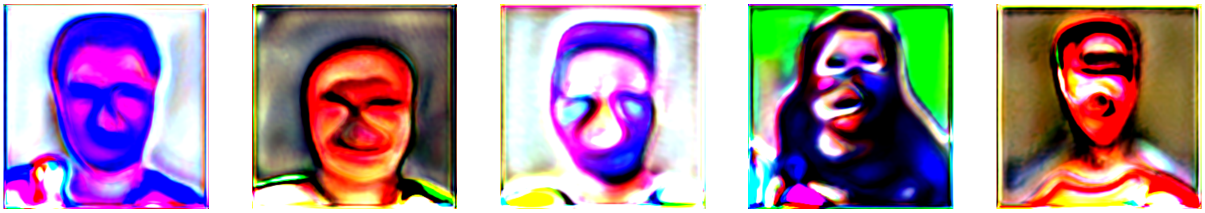


Figure 3: Output of Images Generated Using Perceptual Model Latent Space

3.3 Text Encoder

Usually, noise is passed to a GAN to produce an image, since our application requires us to produce specific images from text descriptions, I trained a straightforward neural network on One Hot Encoded caption vectors to generate the latent space required to generate the images. However, since only 50 latent space vectors were generated, there is a high probability of over-fitting. Different architectures and activation functions (Tanh, Sigmoid, ReLU, and LeakyReLU) were evaluated for the encoder, and the optimal model was:- Input(89) — Hidden1(1024) — LeakyReLU — Hidden2(1024) — LeakyReLU — Output(512)

4 Results and Further Improvements

4.1 Frechet Inception Distance

FID, or the Frechet Inception Distance score, is a metric that measures the distance between feature vectors calculated for real and generated images. The score summarises the degree of similarity between the two groups in terms of statistics on visual features of the original images computed using the image classification model Inception v3 [5]. Lower scores indicate that the two image groups are more similar or have more similar statistics, with a score of 0.0 indicating that the two image groups are identical. The FID score is used to evaluate the quality of images produced by generative adversarial networks, and it has been demonstrated that a correlation exists between lesser scores and higher quality images. Generally, an FID score of 5.06 relates to images produced by StyleGANS trained on CelebA-HQ Database [6].

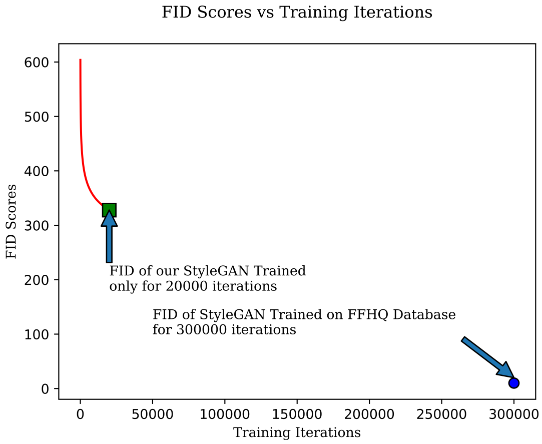


Figure 4: FID vs Training Iterations

4.2 Training Details

After training the StyleGAN for 20,000 iterations, a FID score of 320 was achieved. Given enough training time (3,00,000 iterations), the StyleGAN would be able produce good qualtiy images, and will be able to reach an FID score in between 50 to 100. Although, the aim was to reach a FID score of 5, but since the size of training data provided is less as well as the quality of the images is not up to the mark, 50 is a commendable score.

4.3 Suggested Improvements

Once the text encoder is trained, you can directly provide the caption vectors to generate images. The perpetual model is only used to train the text encoder and is not used during image generation. The Text to Face Generator was evaluated with 100 captions; however, the results were subpar due to computational limitations on training the perpetual model.

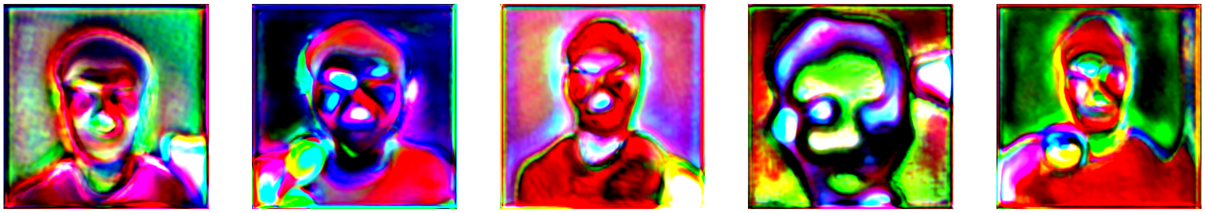


Figure 5: Images generated from Captions

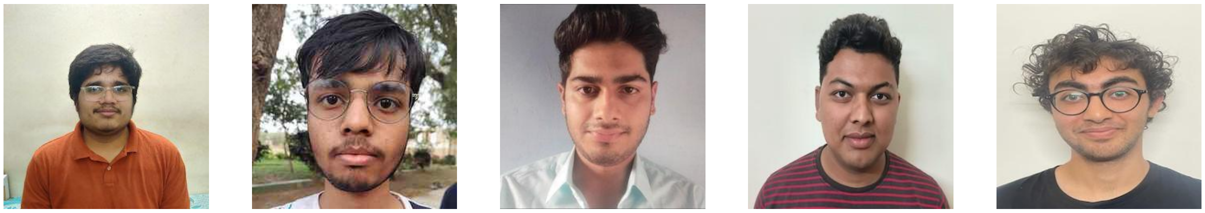


Figure 6: Training Images provided with the same Captions

Training this model for an extended period of time with the multi-line text to latent space generator as opposed to the One Hot encoded vector to latent space generator can further enhance it. Our training dataset should also be expanded to at least 20,000 images, and FID scores should be calculated for at least 10,000 images to obtain an accurate estimate of GAN's quality [1].

5 Note

Please download the model weights from here :- Model Weight (This file is too large to be uploaded to Nalanda). Please place it in the checkpoint folder for the model to work.

References

[1] Martin Heusel et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium.* 2018.

[2] Tero Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *CoRR* abs/1812.04948 (2018).

[3] Jianxin Sun et al. *AnyFace: Free-style Text-to-Face Synthesis and Manipulation.* 2022.

[4] Weihao Xia et al. "TediGAN: Text-Guided Diverse Face Image Generation and Manipulation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2021.

[5] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *CoRR* abs/1512.00567 (2015).

[6] Tero Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* June 2019.