Text to Face Synthesis using Generative Adversarial Networks (GANS)

Submitted in partial fulfillment of the requirements of the degree of

Bachelor of Engineering
(University of Mumbai)

by

Nitish Bhattacharjee (201902003)
Vineet Dabholkar (201902006)
Mihir Dichwalkar (201902009)
Roland Dsouza (201902010)

under the guidance of
Prof. FABIAN BARRETO

Department of Electronics and Telecommunication
Xavier Institute of Engineering

(2022-2023)

Text to Face Synthesis using Generative Adversarial Networks (GANS)

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering

in

Electronics and Telecommunication Engineering

by

Nitish Bhattacharjee (201902003)
Vineet Dabholkar (201902006)
Mihir Dichwalkar (201902009)
Roland Dsouza (201902010)

under the guidance of
Prof. FABIAN BARRETO

UNIVERSITY OF MUMBAI

Department of Electronics and Telecommunication Engineering
Xavier Institute of Engineering
Mahim(West), Mumbai-400016

(2022-2023)

CERTIFICATE OF APPROVAL

This is to certify that is a bonafide work of

Nitish Bhattacharjee	**(201902003)**
Vineet Dabholkar	**(201902006)**
Mihir Dichwalkar	**(201902009)**
Roland Dsouza	**(201902010)**

have satisfactorily carried out the Project work entitled **Text to Face Synthesis using Generative Adversarial Networks (GANS)** in partial fulfillment of Bachelor of Engineering in Electronics & Telecommunication as laid down by University of Mumbai during the academic year 2022-2023.

_____ _____

Prof. Fabian Barreto **(External Examiner)**
(Internal Guide)

_____ _____

Prof. Nitin Ahire **Dr. Y. D. Venkatesh**
(Head of the Department) (Principal)

Project Report Approval for B.E

This project report entitled **Text to Face Synthesis using Generative Adversarial Networks (GANS)** by **Nitish Bhattacharjee, Vineet Dabholkar, Mihir Dichwalkar** and **Roland Dsouza** is approved for the degree of **B.E. in Electronics and Telecommunication.**

_____

(Internal Examiner)

_____

(External Examiner)

_____

(Head of Department)

Date:

Place - Mahim, Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

Nitish Bhattacharjee
(201902003)

_____

Vineet Dabholkar
(201902006)

_____

Mihir Dichwalkar
(201902009)

_____

Roland Dsouza
(201902010)

Date:

Acknowledgement

The successful completion of this undertaking could not be possible without the support of many people whose names may not all be enumerated. We are deeply grateful for all the help that has made our project become a success and aided us in progressing. We give all our glory and honour to the God whose blessings made this endeavour a success.

We are also immensely gratified by the valuable support provided to us by our project guide Prof. Fabian Barreto. The blessings, help and guidance given by her from time to time shall carry us way ahead in the journey of life on which we are about to embark.

A deep sense of gratitude to Fr.(Dr.) John Rose, Director of Xavier Institute of Engineering, Mahim for his cordial support and granting permission to use laboratory whenever needed. We are indebted to our teaching and non-teaching staff for providing us with the valuable guidance, encouragement and advice in every single step of our project.

We are extremely grateful to our principle Dr.Y.D. Venkatesh for providing us with all the facilities required for completion of this project.

Last but not the least, we would like to express our gratitude towards our parents and siblings for supporting and co-operating with us while we were working on this project. Thank you for all encouragement and inspiration given by every person behind this project.

Abstract

Motivated by the unprecedented level of realism achieved by GANs, we focus on the challenging task of synthesizing facial images from text, which has significant potential for diverse computer vision applications. To achieve this, we surveyed literature on various state of the art models, including StyleGAN, DCGAN, and StackGAN.

Our primary objective is to address the limitations of previous models and develop a text-to-face generation model that can generate realistic and diverse face images from textual descriptions. However, implementing this is complicated due to the high dimensionality of natural language and the complexity of facial attributes.

The Deep Fusion Generative Adversarial Network (DF-GAN) is proposed to address and overcome these difficulties. It directly creates high-resolution images using a single stage without causing entanglement between the visual properties of multiple images generated by mulitple stages. The Deep Fusion Block (DFBlock) is the backbone of DF-GAN, which integrates text and visual information to enhance picture quality and semantic alignment. It incorporates the use of a Target-Aware Discriminator made up of One-Way Output and Matching-Aware Gradient Penalty (MA-GP) to improve text-image semantic consistency without the need for additional networks.

The proposed model has several future scopes and could potentially facilitate more realistic simulations for training purposes. It has significant applications in the domain of public safety for the purpose of generating images of suspects or missing persons based on textual descriptions to improve the accuracy and speed of investigations. Moreover, the size of existing datasets can be increased by generating synthetic images from text descriptions.

In conclusion, DF-GAN represents a significant advancement in text-to-face generation and has the potential to open up new possibilities for computer vision applications in various fields.

Keywords: Generative Adversarial Networks, Text-to-Image Generation, Natural Language Processing, Computer Vision, Deep Learning, Neural Networks, Convolutional Neural Networks, Face Synthesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Background

Within less than a decade, the technology of GANs has witnessed rapid growth and has made a huge impact in the field of AI. With continuous advancements in GAN technology, it is possible to synthesize fake images or manipulate real images in a manner that they are nearly indistinguishable to the human eye. This has wide ranging applications in several fields ranging from generating deep fakes to innovative product designs. Currently there are various models available for Text-to-Image [22] applications but there is very limited work done in the domain of Text-To-Face generation. Also, there is a need to get more dataset images for senior citizens and children as deep learning is data hungry and images for the same are very limited. Moreover, a text-to-face GAN could be a powerful tool for generating realistic facial expressions from non-image data, opening up new possibilities for generating lifelike images

1.2 Objective

- To generate high resolution, lifelike facial images using user given text descriptions without any disentanglements using GANs.

- To use the generated facial images for applications such as Criminal Investigation, Plastic surgery preview, Portrait drawing, Increase size of datasets, etc.

1.3 Outline

This dissertation report consists of the following chapters. The contents of the chapters are as follows,

- **Chapter 2: Literature Survey**
  - The domain of Generative Adversarial Networks is surveyed.
  - Several concepts and models related to image generation and manipulation are reviewed & their limitations are analyzed.
  - A problem statement is formulated.
  - The potential scope is reviewed.
- **Chapter 3: Methodology**
  - Previous system of AttnGAN was studied and discussed
  - The Deep Fusion GAN model for generation of face images from text is proposed & it's architecture is detailed.
- **Chapter 4: Implementation**
  - The dataset used for training is studied
  - Different software & system requirements are specified
  - Model training details using different hyperparameters are specified
  - The descriptive algorithm for the working model is discussed.
- **Chapter 5: Results and Discussion**
  - Preliminary results for previous system AttnGAN and finalized results for our proposed model (DFGAN) are presented
  - A detailed comparative analysis of the obtained results is discussed
- **Chapter 6: Conclusion and Future Scope**

1.4 Time Plan

Table 1.1: Time Plan of the project

Timeplan										
	July	August	September	October	November	December	January	February	March	April
Group Formation										
Topic Selection										
Topic Presentation										
Allocation of BE Project										
Literature Survey										
BE Project Progress Evaluation-I										
Implementation-I										
BE Project Progress Evaluation-II										
Synopsis Report										
Troubleshooting										
BE Project Progress Evaluation-III										
Implementation-II										
Troubleshooting										
Final Report (Black Book)										

Chapter 2

Literature Review

Generative Adversarial Networks also known as GANs were discovered by computer scientist Ian Goodfellow in 2014. GAN is an approach to generative modeling that employs competitive neural networks called Generator and Discriminator to generate a new set of data based on training data that is inseparable from it. The generator is analogous to a forger generating fake data and the discriminator is the detective whose main role is to differentiate input data as genuine or fake. Discriminative models in GANs play the role of a classifier. On the other hand, generative models generate realistic images from random samples such as random noise [1].

Within less than a decade, the technology of GANs has witnessed rapid growth and has made a huge impact in the field of AI. With the continuous advancements in GAN technology, it is now possible to synthesize fake images or manipulate real images in such a manner that they are nearly indistinguishable to the human eye. In this paper, we explore GANs and their application in Face Synthesis by reviewing some state-of-the-art models that include GANimation, StyleCLIP, JOJOGAN, InterFaceGAN, and GANs N Roses.

2.1 Existing systems survey

2.1.1 'Generative Adversarial Nets", I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, 2014 [6]

GANs are the generative models which can produce new data instances that resemble our training data. The function of the Generator is to produce samples that are similar to those from the problem domain while that of the Discriminator is to identify whether the given sample is generated by the generator (fake) or from the problem domain (real).

The term adversarial in this network arises from the fact both these networks are engaged in a competitive minimax game wherein each network tries to outperform the other. The Discriminator maximizes the probability of assigning accurate labels to the examples while the Generator tries to minimize the probability of the Discriminator correctly classifying its generated examples as fake. The GAN is trained by first freezing the Generator parameters, performing a forward pass, and then back-propagating the error to update the Discriminator's parameters. The Discriminator is now frozen while the forward pass and backpropagation is performed to update the Generator parameters. Since the inception of GANs [2] a variety of researchers have made significant contributions to the development of improved GANs, including C-GAN, DC-GAN,BI-GAN,SR-GAN,INFO-GAN and CYCLE-GAN.

2.1.2 'Generative Adversarial Networks : A Survey", Tanuj Kakkar, Upasna Singh, 2021 [16]

2.1.2.1 'Conditional Generative Adversarial Networks'

In CGAN images are generated by the model by applying a conditional setting. This means that both the Generator and Discriminator are conditioned over additional information, such as class labels or data [9]. It is able to generate the data point with the target label. Unlike the vanilla(conventional) GAN, Conditional GAN can control data generation. By providing additional information, convergence will also be faster. Applications of CGANs are Image-to-Image translation, Text-to-image synthesis.

2.1.2.2 'Deep Convolutional Generative Adversarial Networks'

Deep Convolutional Generative Adversarial Network (DCGAN) uses convolutional and transpose convolutional layers in the generator and discriminator, respectively. It is capable of high-resolution image generation from a restricted set of images. The layers of DC GANs are not fully connected and use batch normalization in both the generator and discriminator. Here, all the pooling functions are replaced with strided convolutions [14]. DCGAN has had

great success in generating new images. It helps to outpaint the image recursively up to a greater extent, by obtaining larger extended realistic images.DCGANs are introduced to tackle the problem of mode collapse [19] which occurs when the Generator gets over biased towards a particular set of outputs and is not able to generate varied images from the dataset.

2.1.2.3 'Information Maximizing Generative Adversarial Networks'

One drawback of vanilla GANs is that we have no control over the images that they produce, so in Info-GAN [1] to control the types of images produced, we need to feed some specific information along with the random noise to the Generator in order to force it to utilize that information when creating the fake images. This additional information that we are feeding should be related to the type of specific features we want the images to have. It can learn disentangled representations in an unsupervised manner by adding a regularization term, lambda to the loss function.

2.1.2.4 'Super Resolution Generative Adversarial Network'

The function of this architecture is to recover fine textures from the image when we upscale it so that its quality doesn't degrade. Super-resolution GANs applies a Deep-network in combination with an adversarial network to produce higher resolution images. It uses a deep convolutional network with residual blocks. The Objective function contains an adversarial loss and a feature loss. During training, a high-resolution image is downsampled to a low-resolution image, the GAN Generator upsamples the low-resolution images to super-resolution images. We use a discriminator to distinguish the high-resolution images and backpropagate the GAN loss to train the Discriminator and the Generator. The model finally uses a perceptual loss function to ensure variety in images.

2.1.3 'AttnGAN',Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, Xiaodong He, 2017 [21]

AttnGAN helps to generate photorealistic images of face using textual description. It consists of an Attention Network which helps to capture and generate word level information or features along with sentence level info. The Text Encoder used is a Bi-directional LSTM. It uses 3 Feature extraction layers/stages which enhance the images. 64x64x3 $\rightarrow$ 128x128x3 $\rightarrow$ 256x256x3. A DAMSM (Deep Attention Multimodal Similarity Model) checks similarities between the image and word level and sentence level info. Given a Text Description the model outputs a high quality Image. It captures word level and sentence level information The Attention mechanism helps to get accurate and fine-grained high quality images.It will try to find out how relevant the words were for drawing a specific subregion of the image and synthesize fine- grained details at different sub-regions of the image by paying attention to the

relevant words in the natural language description.

2.1.4 'TediGAN: Text-Guided Diverse Face Image Generation and Manipulation', Weihao Xia1 Yujiu Yang Jing-Hao Xue Baoyuan Wu, 2021 [20]

TediGAN, a novel framework for multi-modal image generation and manipulation with textual descriptions. The proposed method consists of three components: StyleGAN [7] inversion module, visual-linguistic similarity learning, and instance-level optimization.The inversion module maps real images to the latent space of a well-trained StyleGAN. The visual-linguistic similarity learns the text-image matching by mapping the image and text into a common embedding space. The instancelevel optimization is for identity preservation in manipulation. To facilitate text-guided multimodal synthesis, we propose the Multi-Modal CelebA-HQ, a large-scale dataset consisting of real face images and corresponding semantic segmentation map, sketch, and textual descriptions.

2.1.5 'GANimation',Albert Pumarola, Antonio Agudo, Aleix M. Martinez, Alberto Sanfeliu, Francesc Moreno-Noguer, 2018 [13]

GANimation enables automatic animation of facial expressions in an image. GANimation helps to edit the facial expressions of any image and can give us the output in the form of continuous expressions/ interpolating the expressions in a smooth transition/ animation (GIF) on the same image without the need for any training i.e. in an unsupervised manner. The desired facial expressions are generated by employing the use of Action Units (AU). StarGAN [2] is one such example of editing an image or the expressions, but the main disadvantage was that it could only generate a few expressions that were present/ defined in the dataset. This limitation of StarGAN was solved using the GANimation approach. It makes use of a system called the Facial Action Coding System (FACS) [5] for generating various facial expressions with the help of different facial Action Units (AUs). In this, 30 different AUs were described some of which are AU1 (Inner Brow Raiser), AU20 (Lip Stretcher), AU2 (Outer Brow Raiser), AU26 (Jaw Drop), etc. This model makes use of the EmotioNet dataset, from which 2,00,000 images were used.

2.1.6 'InterFaceGAN', Yujun Shen, Ceyuan Yang, Xiaoou Tang, Fellow, IEEE, and Bolei Zhou, Member, IEEE, 2020 [15]

InterfaceGAN developed in the year 2019 - 2020 helps to understand that despite GANS having appealing synthesis quality, it remains much less explored about what knowledge GANs learn in the latent representation and how we can reuse such knowledge to control the generation process. For eg Given a latent code, how does GAN determine the attributes of

the output face, e.g., an elder man or a young woman? How are different attributes organized in the latent space? etc. It helps to provide a way to understand how a GAN works, and how a GAN model understands what it has learned in the latent representation (latent space) and combines them to form different random codes to generate a photo-realistic image. This also helps us to study disentangled representations learned by GAN and provides a pipeline to discover the properties of various facial representations, further it also helps to edit the images based on their pose, smile, age, gender, eyeglasses, etc.

2.1.7 'GANs N' Roses', Min Jin Chong and David Forsyth, 2021 [3]

This paper discusses the rendering of an input face image to it's equivalent anime version. Content is defined as what changes when an image is geometrically manipulated while Style is that which remains constant. The model implements Control, Consistency and Coverage. A stark variation from other related models like CouncilGAN [10] and AniGAN [8] is that the GNR framework takes a single image and performs several geometric augmentations like scaling, rotating and cropping to generate multiple images as training examples instead of using different images for training. The Framework includes a combination of Encoder and decoder. The encoder separates the input augmented images into their corresponding set of content codes c(x) and style codes s(x). This extracted set of content codes is passed to the decoder along with style codes which are sampled from a Normal distribution of anime style codes sz. This applies a variety of styles to the facial image without altering it's core content. The Descriptive Discriminator plays an important role in diversifying the generated images as it learns the Standard Deviation across minibatches.

2.1.8 'JoJoGAN', Min Jin Chong and D.A. Forsyth, 2021 [4]

JoJoGAN uses the Style transfer method which helps you transfer a face image into another style of your wish. It takes a face (content) image and then applies the style and texture of the reference image over the content image so as to get a blend of both images in the output image. JoJoGAN aims to solve problems like failure to capture distinct style details, diversity, or lack of image quality. It works by first approximating a paired training dataset and then finetuning a StyleGAN to perform one-shot face stylization. The fine-tuning process looks at a lot of generated realistic faces that look like the reference image. It then tries to convert these realistic image into the reference image as close as it can. After the network gets used to generating reference images based on the realistic image, if any content image is passed, it will convert it to the style of the reference images while not affecting the facial structure of the content image. JoJoGAN's style mapper produces good-looking outputs, faithfully has features from the style reference and also preserves the person's identity. It uses FID which is used to evaluate the quality and diversity of images generated by the GAN.

2.1.9 'StyleCLIP', Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, Dani Lischinski, 2021 [12]

StyleCLIP combines the generative power of StyleGAN with CLIP's image-text embedding to give very intuitive text-based image manipulation. CLIP [11] is trained on full sentences and the intuition behind this is that the model can learn a lot more things from sentences and find patterns between images and texts. It consists of two encoders, one for text and one for images. These encoders are trained to map images and corresponding text into a common latent space where it checks for the similarity between the text and the image. Using StyleCLIP we can manipulate an image by using the guidance of only text to get photorealistic images. We get disentangled edits, for example, by typing "Mohawk Hairstyle" the face remains intact but the hair of the input face image is changed to a mohawk hairstyle. The authors investigated three techniques that combine CLIP with StyleGAN (StyleGAN2) which are Text-guided Latent Optimization, Latent Mapper and Global Directions.

2.1.10 Comparison:

Table 2.1: Comparison between the various models

GAN models	Function	Dataset	GPU used by Author	Additional Losses mentioned
StyleCLIP	Text-based Image Manipulation	CelebA-HQ	NVIDIA GTX 1080Ti	Identity (ID), CLIP Loss
JojoGAN	Style Transfer	FFHQ, CelebA-HQ	Nvidia A40	Identity, Perceptual Loss
GANS N' Roses	Image to Anime	selfie2anime	NVIDIA Quadra RTX 4000	Style and cycle consistency, Diversity discriminator with Adversarial Loss
GANimation	Image and GIF editing	EmotioNet and RaFD	GTX 1080 Ti GPU.	Image-adversarial, Attention, Conditional, Identity and Full Loss
InterFaceGAN	Image Editing	CelebA	GTX 1080 Ti GPU.	Multitask Loss

2.2 Limitations

1. Lack of high quality face image datasets with textual descriptions
2. Lack of models for Face Generation but plenty for Face Image Manipulation
3. Mode collapse
4. Long Training time required
5. Quality of Facial images generated unsatisfactory

2.3 Problem Statement

To synthesize facial images from corresponding textual descriptions using Generative Adversarial networks (GANs).

2.4 Scope

1. GANs are unsupervised deep learning models and that can generate output that closely resembles real-life data(images, text, audio, video, etc).
2. Some of the applications of GANs are:
   (a) Face Aging/Deaging used to simulate future or past appearance of a particular person
   (b) Super Resolution used to upscale the quality of images
   (c) Generating fake images (Deep Fakes) for advertisement, modelling etc.
3. While extensive research has been conducted in the field of Image-to-Text Generation, the same level of research has not been conducted in the reverse, i.e., Text-to-Image Generation.
4. Text-To-Face generator can be used for generating sketches of criminals quickly by entering the features of their faces.

Chapter 3

Methodology

3.1 Previous system- AttnGAN [21]

Attentional Generative Adversarial Network (AttnGAN) enables fine-grained text-to-image generation through attention-driven, multi-stage refinement. The AttnGAN synthesizes fine-grained details at different sub-regions of the image based on natural language descriptions by paying attention to the relevant words. Furthermore, a deep attentional multimodal similarity model is proposed for training the generator using fine-grained image-text matching losses.

The Attentional Generative Adversarial Network (AttnGAN) has two main components as follows:

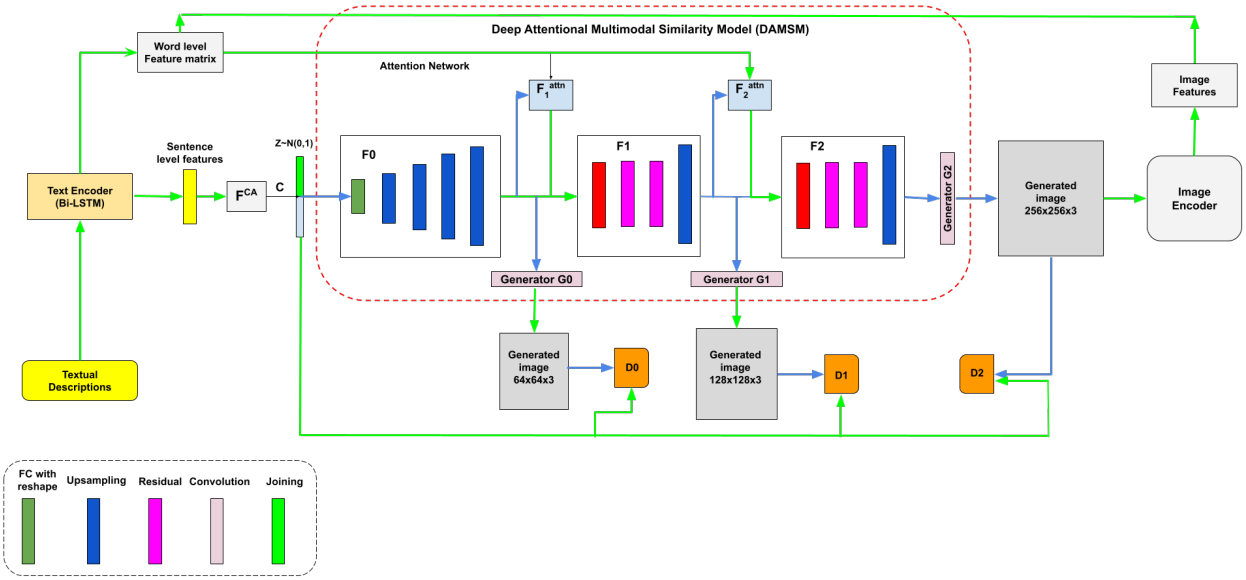

Figure 3.1: Block Diagram of AttnGAN

3.1.1 Attention Generative Network [24]

By giving attention to words that are more relevant to specific subregions, the attention model helps the generative network draw different subregions. The input text description is given to the text encoder which is a Bi-directional LSTM [17] which converts it to word level and sentence level features [4]. The sentence level features are conditionally augmented using noise. They are then combined with a noise vector sampled from a normal distribution. This is then input into the Attentional Generative Network. The attention model has two inputs, the word features and the image features from the previous hidden layer of the model. It has a total of 'm' generators, the more the generators more the computational load. It is seen that three stages save resources as well as give better outputs. These generators take the hidden states as input and then pass it through a series of convolutional and upsampling layers which generate images ranging from small-to-large in 'm' stages.

The output features given to the first generator produce a low-resolution image. The word level features are converted into an attention matrix of words and their relevance or importance. This matrix is used as an input to the subsequent layers after the first set of Convolutional layers. Upsampling operations are performed and the resultant feature map is passed to the second generator in order to generate another image with better resolution. This continues until we get a high resolution image with a focus on important features.

3.1.2 Deep Attentional Multimodal Similarity Model

The final image is given to an Image encoder which converts it to features and compares the Image-text similarity with the word feature matrix. It calculates the DAMSM (Deep At-

tentional Multimodal Similarity Model) loss [4]. The DAMSM loss is designed to learn the attention model in a semi-supervised manner, in which the only supervision is the matching between entire images and whole sentences (a sequence of words).

The DAMSM learns two neural networks that map sub-regions of the image and words of the sentence to a common semantic space, thus measuring the image-text similarity at the word level to compute a fine-grained loss for image generation.

3.2 Proposed System - Deep Fusion GAN (DFGAN) [18]

Deep Fusion GAN (DFGAN) is a text-to-image generative model which uses a simplified backbone architecture than Attngan with a single Generator and Discriminator network along with a pre-trained Bi-Directional LSTM as its text encoder.

The Bi-Directional LSTM text encoder extracts semantic vectors from the textual description given. Two inputs are taken in by the Generator, the encoded sentence by the text encoder and a Gaussian distribution sampled noise vector to ensure that the generated images are diverse. The architecture then consists of UPBlocks which consists of three elements: The upsampling layer that 'upsamples' the image features, the residual block that takes care of vanishing gradients, and DFBlocks to fuse the text and image features during the image generation process. The image features are then transformed into images by a convolutional layer. A series of DownBlocks is used in the Discriminator to convert images into image features. Next, the sentence vector is duplicated and concatenated with the image features. The images are evaluated for their visual realism and semantic consistency using the adversarial loss. It helps distinguish between generated images and real images from the dataset while encouraging the generator to synthesize images with a high quality and text-image semantic consistency. Hinge loss [23] is used here as a mechanism to enhance the training process. Here, the final synthesized image is passed along with the real sample image to get the serial loss.

DFGAN uses a Matching-Aware zero-centered Gradient Penalty, a regularization technique which pushes the real points towards the minimum of the loss curve. Smoother convergence is achieved by smoothing this surface around and for the real data points. In addition, a Deep Fusion block is used, which fuses text and image seamlessly, resulting in a more focused background compared to other models.

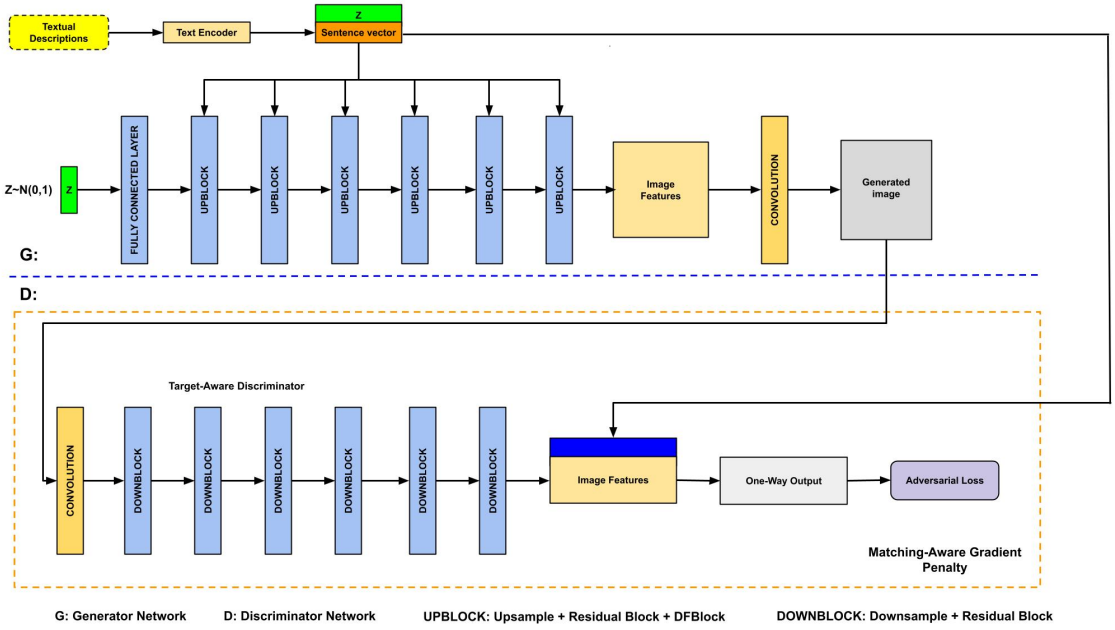

Figure 3.2: DFGAN Architecture

DFGAN consists of the following blocks in its working :

3.2.1 UpBlock

Upblock is a building block used in the generator network for image synthesis. It consists of a sequence of operations that increases the spatial resolution of the input feature maps while amalgamating the text-image context/information. It further consists of the following elements:

1. UpSampling Layer: The Upsampling layer is a simple layer with no weights that will double the dimensions of input. It upsamples the image features and passes it to the following layers.

2. Residual Block: It is a type of neural network layer that helps in improving the quality of the generated images. The residual block addresses the problem of vanishing gradients by introducing a shortcut connection between the input and output of the block, allowing the input to bypass some of the layers and be added directly to the output. This helps to preserve information and prevent the gradients from vanishing and degrading as it passes through the network.

3. DFBLOCK: Deep Fusion Block is used to enhance the text-image fusion process. A single DFBlock consists of multiple Affine layers stacked with ReLU layers between them which conditions the image features according the text. Strengthening the fusion

process promotes the diversity of visual features and enlarges the representation space to represent different visual features according to different text descriptions.

3.2.2 Convolution

It converts the image features into images. It is a transposed convolution layer which takes in the text-conditioned features to generate an image. A transposed convolution, also known as a deconvolution, is a type of operation used in Convolutional Neural Networks (CNNs) to increase the spatial resolution of the input feature map. Unlike a regular convolution that reduces the spatial dimension of the feature map, a transposed convolution increases it by "padding" the input with zeros and performing a convolution operation by using a learnable kernel (also known as a filter).

3.2.3 DownBlock

1. Downsampling Layer : This layer does the opposite of the upsampling layer. It reduces the dimensionality of the input feature map and captures more global features of the image. This results in the reduction of parameters and thus the computational cost of the network. The most common type of downsampling layer is the max pooling layer.

2. Residual Layer : This layer is the same as that used in the Generator. It counters the problem of vanishing gradients by introducing a shortcut connection between the input and output of the block.

3.2.4 Target-Aware Discriminator block

The Vanilla GAN Discriminator block is modified to be Target Aware i.e. to generate high quality images which are semantically consistent with the textual description entered by the user. This also increases the discriminator's ability to effectively differentiate between the generated and the real images.
In order to achieve this, the discriminator incorporates the following

3.2.4.1 Matching-Aware Gradient Penalty (MA-GP)

MAGP is employed so that the generated image resembles the textual description entered by the user i.e. to verify the text-image semantic consistency. It refers to the degree to which the textual description of an image matches the visual content of the image itself. MAGP achieves this by computing the distance between the generated image latent vectors and the real data latent vector. This ensures that the generated images are consistent with the text in terms of feature representation, which enhances the quality of the generated samples.

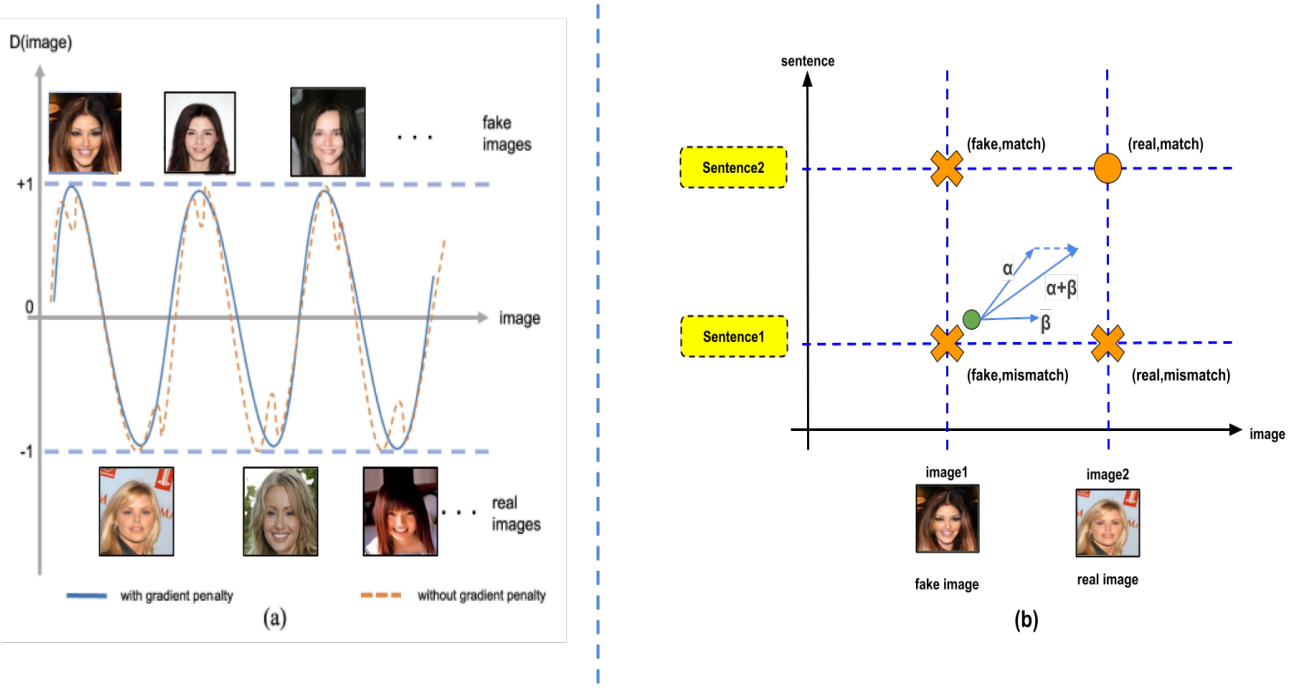

Figure 3.3: Matching-Aware Gradient Penalty (MA-GP)

3.2.4.2 One-Way Output

Research has demonstrated that utilizing the Two-Way Output method in text-to-image GANs can diminish the effectiveness of MA-GP and impede the generator's convergence. Specifically, the conditional loss creates a gradient (referred to as α) that directs towards the real and matching input data following backpropagation, while the unconditional loss generates a gradient (referred to as β) that solely directs towards the real images. However, the final gradient produced by combining α and β does not direct towards the intended real and matching data points. This diversion from the intended direction adversely affects the generator's capability to produce images that are both realistic and semantically consistent with the input text, leading to slower convergence.

To tackle this problem, a new approach known as One-Way Output is proposed for text-to-image synthesis. In this approach, the Discriminator concatenates the sentence vector and the image feature and generates only one adversarial loss through two convolution layers. This enables the single gradient to directly point towards the real and matching data points, leading to a more optimized and accelerated convergence of the generator. Overall, the One-Way Output approach proves to be a more effective and efficient method for text-to-image synthesis compared to the Two-Way Output approach employed in previous GAN models.

Chapter 4

Implementation

4.1 Dataset

The CelebA dataset is a large-scale face attributes dataset that was created by researchers at the Chinese University of Hong Kong, Z. Liu, P. Luo, X. Wang, and X. Tang and was released in 2015. It contains over 200,000 celebrity images, each labeled with 40 different attributes such as gender, age, facial expression, and presence of accessories like glasses or hats. The dataset is designed to be used for a variety of tasks, including face recognition, facial attribute analysis, and face synthesis.

The images in the CelebA dataset were collected from the internet and include a diverse range of celebrities from different ethnicities, ages, and professions. The dataset also includes images with a range of different lighting conditions, facial expressions, and poses, making it useful for training models that can handle a variety of real-world scenarios.

In addition to the labeled attributes, the CelebA dataset also includes landmark annotations for each image, which provide the locations of key facial features such as the eyes, nose, and mouth. This can be used to train models for tasks such as facial landmark detection or facial expression recognition.

The CelebA dataset has become a popular benchmark for face-related tasks in computer vision, and has been used in a wide range of research papers and competitions. The dataset

is freely available for download on the internet, and there are many pre-trained models and code examples available online for those looking to use the dataset for their own projects.

4.2 Software and Frameworks

The following software and frameworks were used to enable efficient development of the deep learning model and the web application.

1. **Python 3.7**: Python is a popular programming language that is widely used for machine learning and data analysis. Python 3.7 is an older version of Python that was released in 2018.

2. **Pytorch 1.6**: Pytorch is an open source machine learning library that is widely used for developing deep learning models. Pytorch 1.6 is a specific version of Pytorch that was released in 2020. It offers a flexible and dynamic computational graph that enables developers to easily define and modify their models during runtime.

3. **Streamlit**: Streamlit is an open source web framework that is used for building interactive data science applications. It allows users to create web applications quickly and easily, without needing to know HTML, CSS or JavaScript. It provides a simple and intuitive API that enables developers to create applications quickly and easily.

4. **CUDA**: CUDA is a parallel computing platform and programming model developed by NVIDIA for GPUs. It enables developers to use the power of GPUs to accelerate computation for deep learning and other data-intensive applications. It provides a set of tools and libraries that enable developers to write high-performance code for a wide range of applications.

5. **Visual Studio Code**: Visual Studio Code is a popular source code editor that is used by many developers. It has a range of features that make it useful for developing deep learning models, including support for debugging, syntax highlighting, code completion, and Git integration. It is highly customizable and supports a wide range of programming languages and frameworks.

6. **Weights and Biases (wandb)**: Weights and Biases (wandb) is a machine learning platform that provides tools for experiment tracking, visualization, and collaboration. It can be used with various machine learning frameworks, including Pytorch, which was used in this project. Wandb allows users to log and track various aspects of machine learning experiments, such as hyperparameters, model architecture, data, and metrics. It also provides interactive visualizations to help users analyze and compare their experiments. Additionally, wandb offers features for collaboration, such as sharing experiments with team members and creating reports.

4.3 System Requirements

The following are the system requirements for running the software:

1. **Operating System**: 64-bit Microsoft Windows 10.
2. **CPU Architecture**: x86_64 architecture with support for a Windows Hypervisor. A 2nd generation Intel Core processor or newer, or an AMD CPU is recommended.
3. **RAM**: At least 32 GB of RAM is required for running the software and training models.
4. **Graphics Processing Units (GPUs)**: The following NVIDIA GPUs were used for training the GANs:
   (a) Nvidia RTX 2060 Super
   (b) Nvidia Geforce GTX 1650
   (c) Nvidia Geforce GTX 1050
5. **Disk Space**: A minimum of 100 GB of available disk space is required for installing the necessary software and storing the datasets.
6. **CUDA Toolkit**: The NVIDIA GPU Computing Toolkit (Cuda toolkit) version 11.6 or later is required for running the deep learning models on the GPUs.

4.4 Libraries

1. **os**: A module that provides a way of using operating system dependent functionality like reading or writing to the file system.
2. **random**: A module that provides functions to generate pseudo-random numbers for various distributions.
3. **functools**: A module that provides higher-order functions and operations on callable objects.
4. **pandas**: A library that provides fast and flexible data structures for data manipulation and analysis.
5. **numpy**: A library that provides support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
6. **matplotlib**: A library that provides a comprehensive set of tools for creating static, animated, and interactive visualizations in Python.

7. **collections**: A module that provides specialized container datatypes beyond the built-in containers like lists, tuples, and dictionaries.

8. **dataclasses**: A module that provides a decorator and functions for creating classes that are primarily used to store data.

9. **PIL**: A library that adds support for opening, manipulating, and saving many different image file formats.

10. **tqdm**: A library that provides a progress bar for any iterable in Python.

11. **torch**: A library for building and training neural networks, with support for automatic differentiation and a dynamic computational graph.

12. **torchvision**: A package that provides access to popular datasets, model architectures, and image transformations for computer vision tasks using PyTorch.

4.5 Training Details

The primary programming language used for the construction of our model is Python (version = 3.6). To ensure a consistent and reproducible development environment, a virtual environment with required libraries including PyTorch was setup on the physical machines using Anaconda. In order to harness the capabilities of the GPU, Torchvision and CUDA toolkit are installed in the environment. Different deep learning libraries and packages are installed and managed using the 'conda' or 'pip' command on the Anaconda terminal.

DFGAN is trained using a combination of physical GPUs and cloud-based computing resources from Google's Colab. Colab provides access to high-performance GPUs which enabled a fast, efficient training process that was limited only by the free usage period available. This enabled us to run multiple versions concurrently albeit for a short period of time. A variety of different hyperparameters such as learning rates of Generator and Discriminator, batch sizes were experimented with, simultaneously, in order to optimize the performance of the model. This was done under the interactive supervision of the Wandb API which helped in narrowing down the combinations for the most optimum results.
Following are the list of learning rates of the Generator and Discriminator as well as the batch sizes used for experimentation

Table 4.1: Training Parameters on Tesla T4

Learning Rate (G/D)	Batch Size	Epochs Trained	Time Taken per Epoch (mins)
0.0001 / 0.0001	16	60	~42
0.0001 / 0.0001	32	70	~37
0.0001 / 0.0004	16	100	~38
0.0002 / 0.0002	16	45	~40
0.0002 / 0.0002	32	80	~36
0.0003 / 0.0003	16	50	~38

4.6 Working

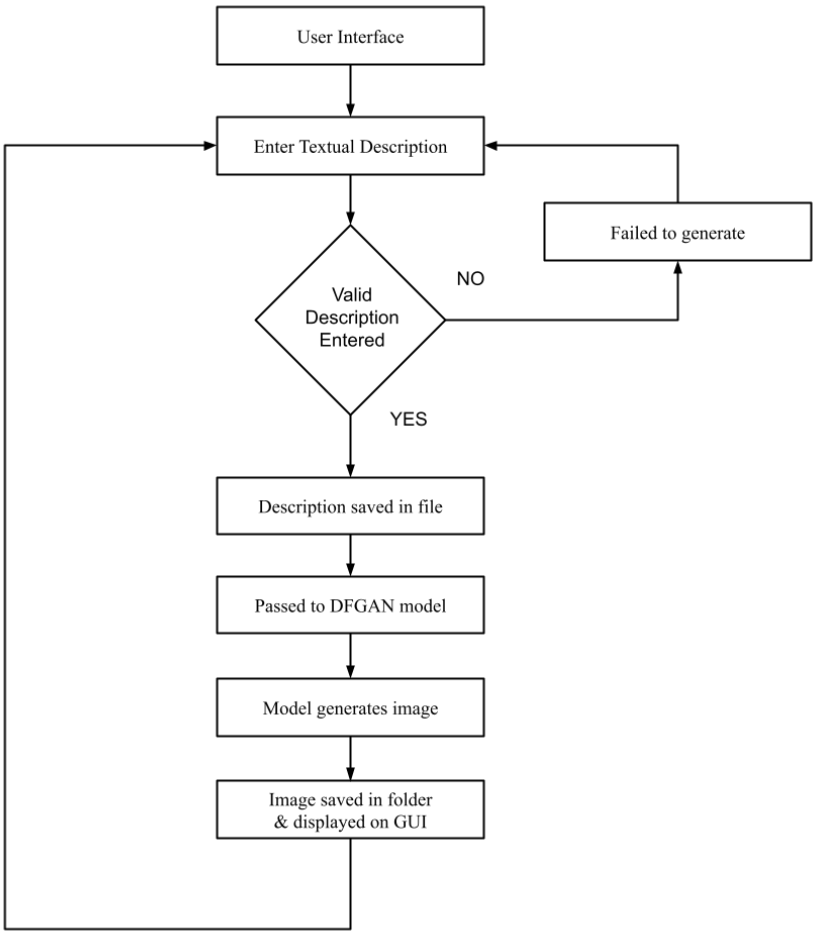

Figure 4.1: GUI Flowchart

The user interacts with the graphical user interface (GUI) made using streamlit to enter a textual description (captions) of the face they want to generate. The captions(sentences) should be limited to around 5 sentences as it is the optimum length to generate best outputs. The description is verified to see if it is valid. If not, then the image is not generated and the user is prompted to enter the description again. These captions are sent to a server using a JSON response.

The JSON response is received on the server where they are saved in a text file and passed to the DFGAN model. The model then generates the image locally based on the description provided by the user. Lastly, the generated image is converted to base64 and sent back to the server to be displayed on the user interface. The user can re-enter a new description if they want to generate another image.

Chapter 5

Results & Discussion

5.1 Results

The implemented results for the models described before are discussed below

5.1.1 AttnGAN

The AttnGAN model which is pretrained on the Caltech-UCSD Birds 200 (CUB-200) dataset is first tested for the generation of Bird images from text. The Model is then trained using the Multi Modal CelebA-HQ dataset for face images.

5.1.1.1 Birds

Pretrained model for birds

A crow which looks like a hummingbird

Bird with black feather,blue crown and long beak

A small yellow bird with red crown and short beak

Figure 5.1: Birds generated from textual descriptions

5.1.1.2 Face

Intermediate training epochs for Faces

Epoch 26

Epoch 28

Epoch 89

Figure 5.2: Intermediate Testing Images

5.1.2 Wandb

The Weights and Biases site for the tracking of losses of the Generator and Discriminator for Batch sizes 16 (lr=0.0002) and 32 (lr=0.0001)

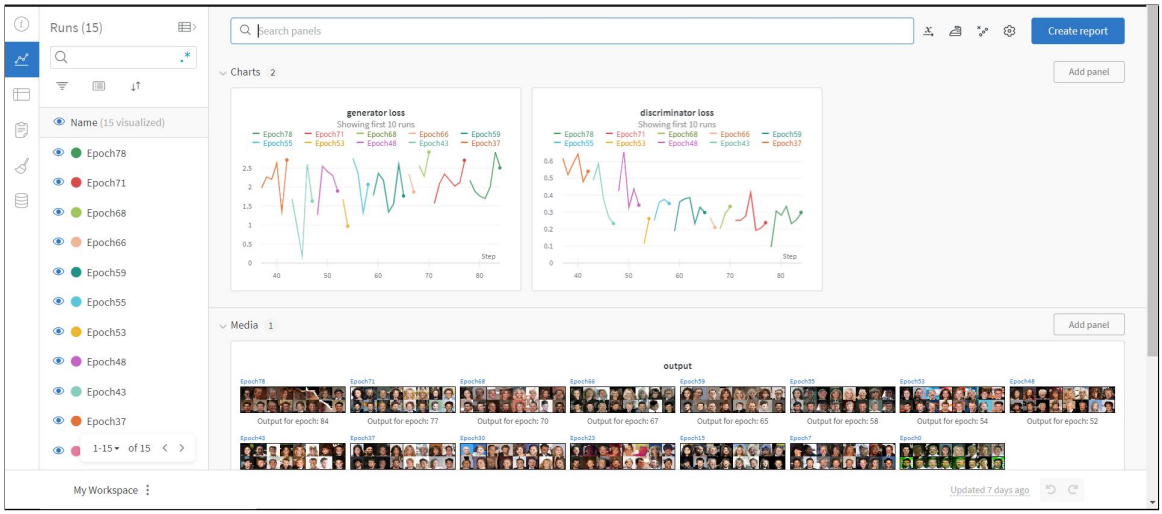

Figure 5.3: Wandb Interface

5.1.2.1 Batch 16 and Lr = 0.0002

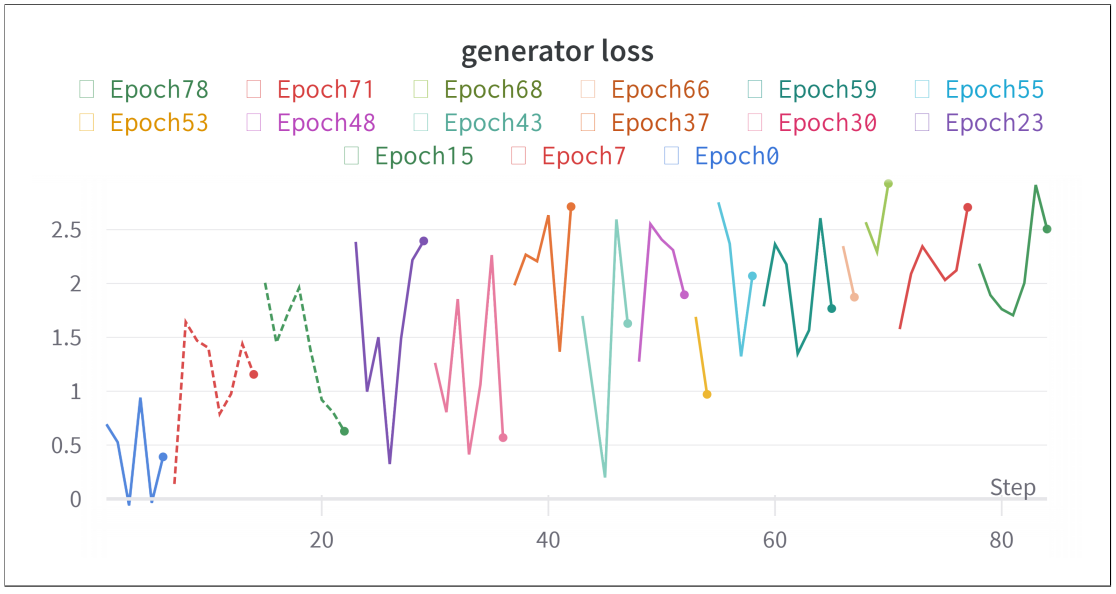

Figure 5.4: Generator Training graph

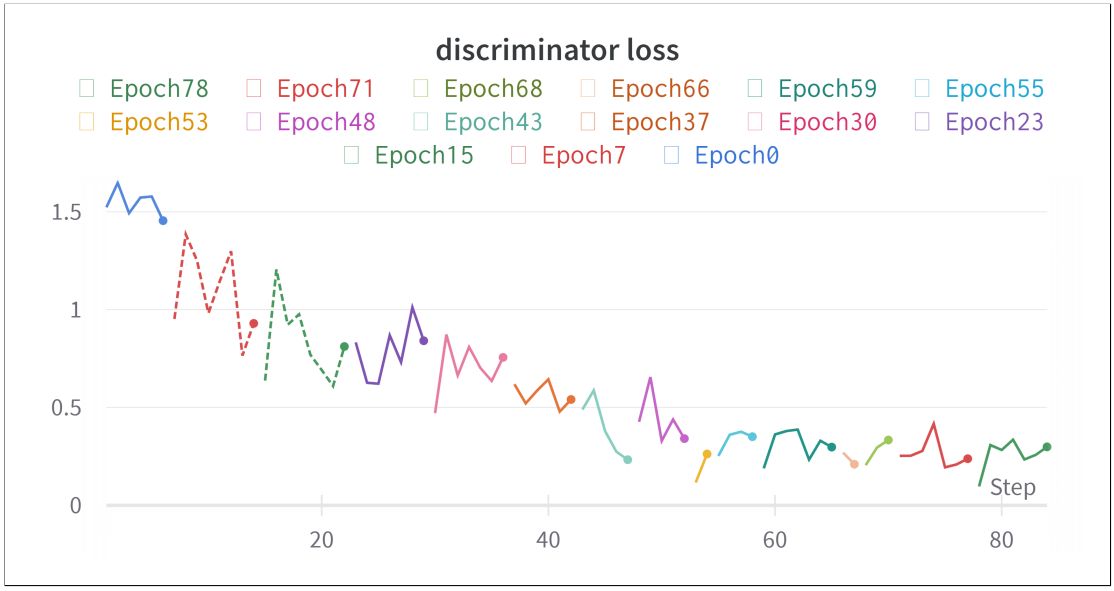

Figure 5.5: Discriminator Training graph

5.1.2.2 Batch 32 and Lr = 0.0001

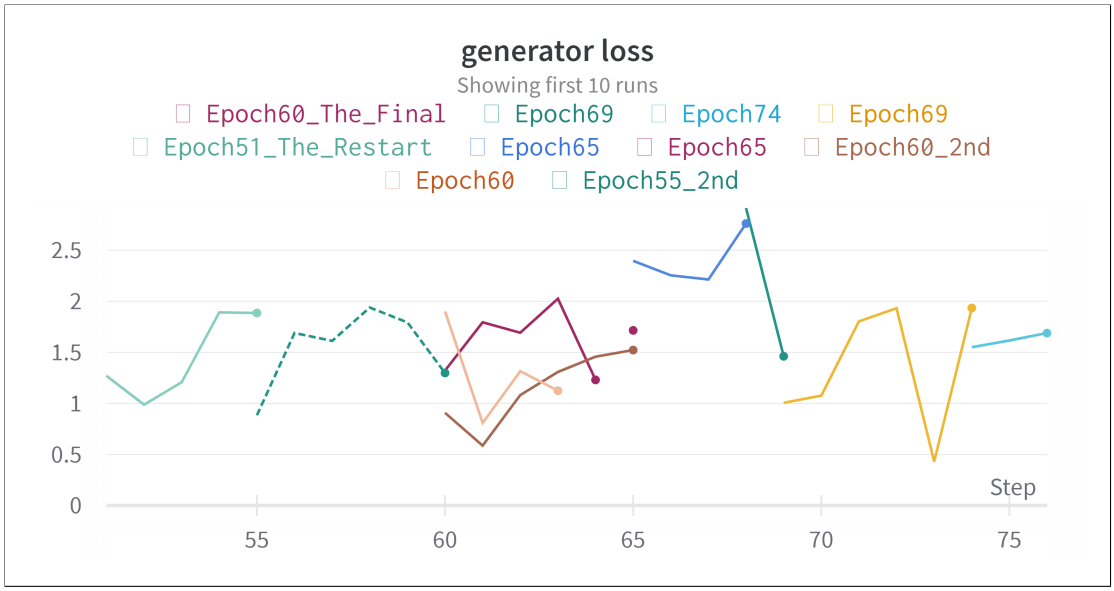

Figure 5.6: Generator Training graph

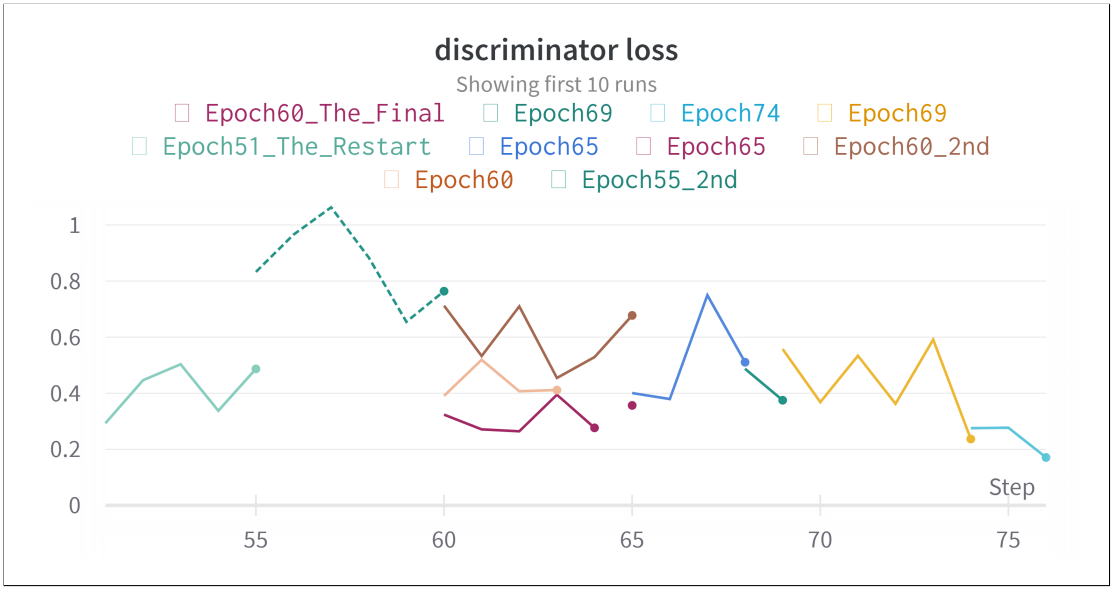

Figure 5.7: Discriminator Training graph

5.1.3 DFGAN

The DFGAN model trained with two different batch sizes and learning rates tested with two different textual descriptions

5.1.3.1 Young Female

Description : The attractive young female is attractive and has pale skin. She is very attractive and has heavy makeup. Her hair is black and wavy. She has a slightly open mouth and a pointy nose. The attractive female has a 5o' clock shadow and has wavy hair. She is wearing a lipstick.

1. **Batch-size 16**

Figure 5.8:
Lr = 0.0002 and Epoch 80

Figure 5.9:
Lr = 0.0003 and Epoch 45

2. **Batch-size 32**

Figure 5.10:
Lr = 0.0001 and Epoch 60

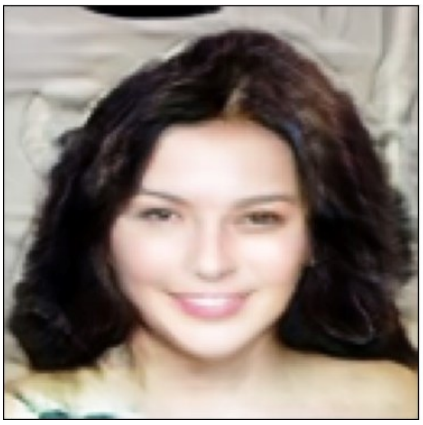

Figure 5.11:
Lr = 0.0002 and Epoch 39

5.1.3.2 Young Male

Description : This young man has high cheekbones. He has straight hair which is brown in colour. He has arched eyebrows and a slightly open mouth. He has a receding hairline. He is young with arched eyebrows.

3. **Batch-size 16**

Figure 5.12:
Lr = 0.0002 and Epoch 80

Figure 5.13:
Lr = 0.0003 and Epoch 45

4. **Batch-size 32**

Figure 5.14:
Lr = 0.0001 and Epoch 60

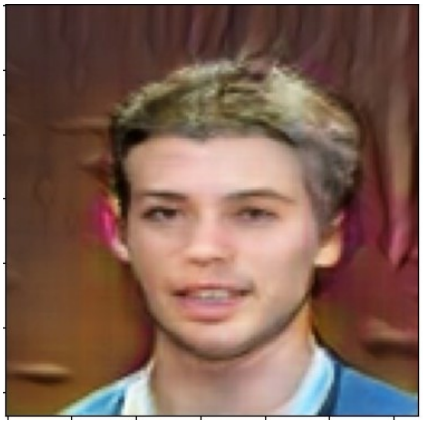

Figure 5.15:
Lr = 0.0002 and Epoch 39

5.1.3.3 GUI interface

The user interface to input the text for the model to generate the image and display it

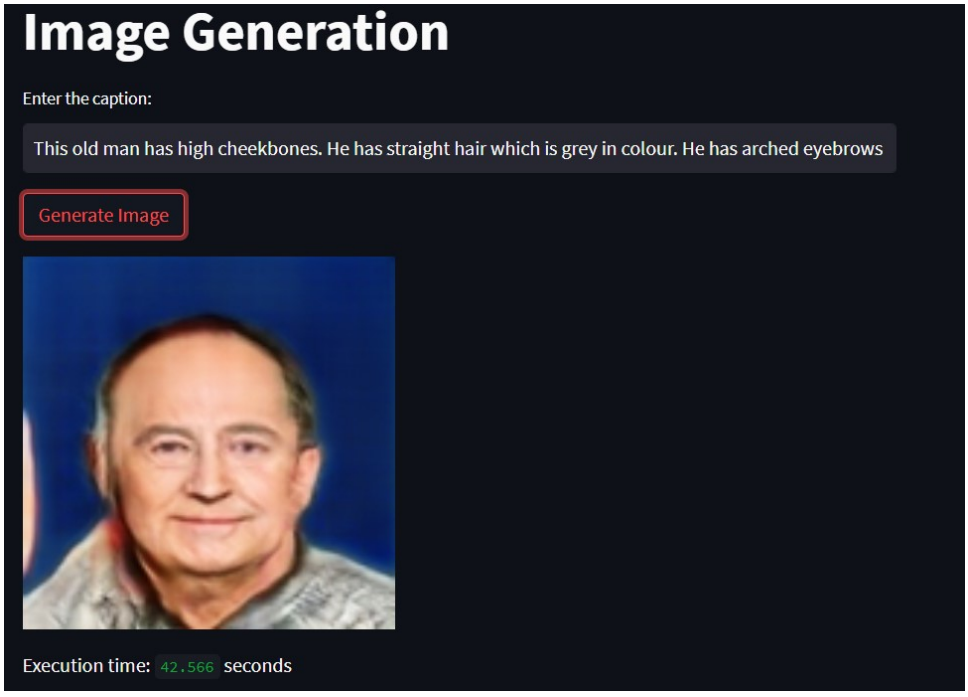

Figure 5.16: Batch 16 and lr=0.0002

5.1.3.4 Mode Collapse

The problem of mode collapse encountered during training as well as the degraded images generated using that trained model

Figure 5.17: Batch16, Lr= 0.0001 (G) and 0.0004 (D)

Figure 5.18: Batch 16, Lr: 0.0001 and Epoch 24

5.1.3.5 Out of Vocabulary

The model's performance is tested using descriptions containing out of vocabulary keywords/features

1. **Description :** This old man has gray balding hair and wears a blue necktie. He has wrinkles on his face and is old. He has small ears, a pointy nose and is frowning. He is wearing pink spectacles and has a chiseled jawline.

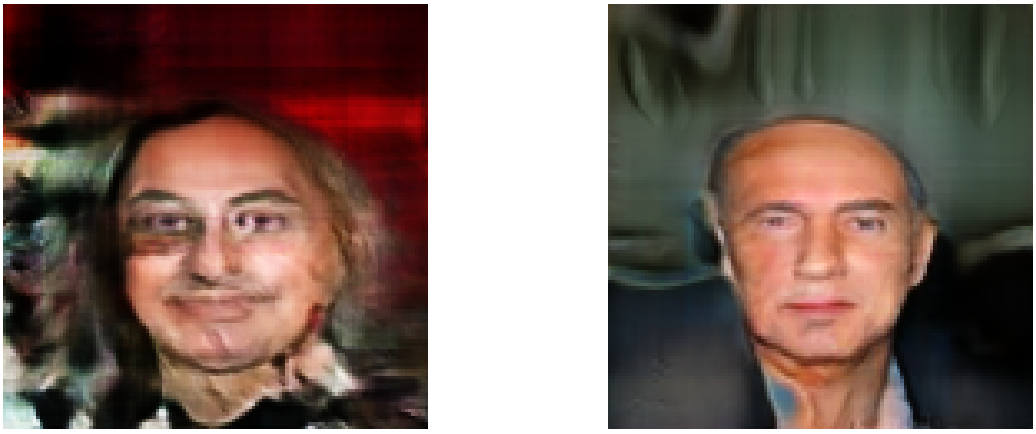

Figure 5.19: Batch 16 and Batch 32

2. **Description :** This old woman has gray balding hair and wears a blue necktie. She has wrinkles on her face and is old She has big ears, a pointy nose and is frowning. She is wearing pink spectacles, has big lips and a sexy jawline.

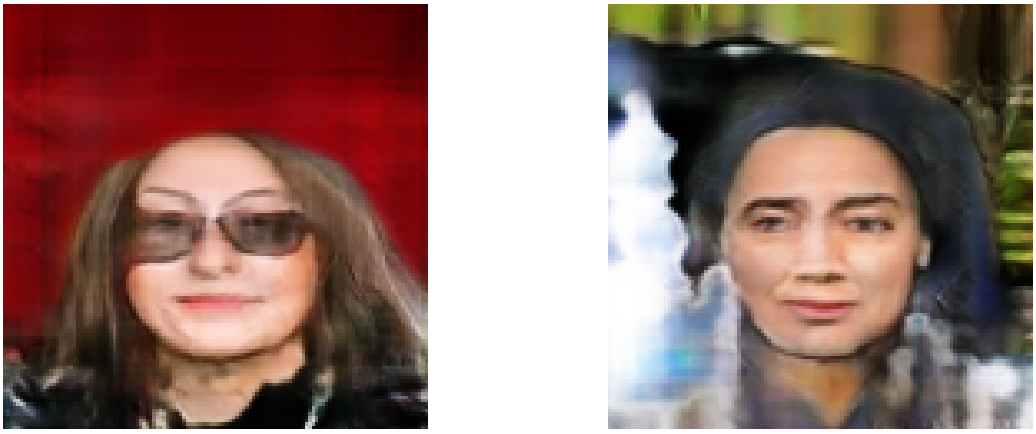

Figure 5.20: Batch 16 and Batch 32

5.2 Comparision/Analysis

DFGAN's performance is evaluated based on its ability to generate high-quality images while training on different hardware setups, such as Google Colab and physical GPUs, using various learning rates. A comparative analysis of the model suggests that the model had a relatively successful training process with promising results. During the training process, it is observed that the model faced the frequent issue of Mode Collapse [19], which required careful hyperparameter tuning to circumvent.

To optimize the model's performance, various hyperparameters, including batch sizes and learning rates were experimented with. It is observed that the quality of the generated images remained consistent for batch sizes of 16 and 32, with both configurations providing equally good results. The most promising combinations were found to be Batch-size 16 with

Lr= 0.0002 and Batch-size 32 with a variable learning rate. The Batch 32 version learns better low level facial features but collapses frequently at around 50 epochs. It however doesn't properly learn high level features like "Spectacles","Hat" even at epoch 60. The Batch 16 version persists for a longer time before collapsing epoch 80 onwards. It learns high level features like "Eyeglasses" very efficiently albeit at the cost of the quality of other low level facial characteristics. It is observed that the model generates better output for female faces than males. This could possibly be due to the fact that male faces have more complex facial features to learn like mustaches and beards which the model struggles to capture efficiently. Another contributor to this could be the inherent bias of the CelebA dataset towards females (Female:Male=58:42)

On average, the training time for an epoch ranged between 25-40 minutes, depending on the learning rate and batch size. Overall during training, the model demonstrated a high degree of fidelity and accurately captured the attributes specified in the captions. It is, however, unable to capture extremely high-level attributes such as "Hat","Beard", "Earrings" which require a higher number of epochs without collapse, following which the model's performance starts degrading.

While testing, it is observed that the model's performance improved with the repetition of similar captions, indicating that it learnt better with increased exposure to specific attributes. It is also observed that the model infers & learns features not explicitly mentioned in the dataset like old, female, etc. Out of vocabulary descriptions are used to test the robustness of the model. The Batch-size 32 version is found to be superior to the Batch-size 16 version in this regard. The output, however, degrades if it is exposed to a large amount of out of vocabulary captions.

Future work could focus on improving the model's performance in learning difficult attributes such as "Beard" and "Hat" efficiently while simultaneously reducing the computational time required for training. Overall, the generated images demonstrate a high degree of fidelity and provide a strong foundation for future work in this field.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Text-to-face generation using Deep Fusion GAN (DFGAN) is a promising approach for synthesizing high-quality realistic facial images from textual descriptions. This technology has the potential to revolutionize a wide range of applications, from creating lifelike avatars in video games and virtual reality environments to generating images of missing persons in forensic investigations. DFGAN is a novel GAN architecture that addresses the limitations of previous text-to-image synthesis approaches by incorporating a deep fusion approach to combine text and image modalities.

The hyperparameter tuning experiments conducted on DFGAN revealed the significant impact of batch size and learning rate on the model's performance. Although the model faced difficulty in learning high-level attributes such as "Hat", "Beard", "Earrings", it demonstrated robustness and generalization ability by inferring some facial features not explicitly mentioned in the dataset. There is still room for improvement in terms of reducing training time and improving the model's ability to learn difficult attributes.

Nonetheless, the success of text-to-face generation using DFGAN has significant implications for industries such as entertainment, e-commerce, and virtual assistants. As the technology advances, it has the potential to make virtual agents and digital interfaces more

human-like and engaging.

However, we also recognize the ethical considerations related to the use of text-to-face generation technology, such as the risk of misuse for creating deepfakes and infringing on individuals' privacy rights. It is crucial to acknowledge these concerns and take steps to address them in the development and use of text-to-face generation technology. One potential solution is to incorporate security measures, such as watermarking or other identifying markers, to help distinguish authentic images from manipulated ones.

6.2 Future Scope

There are several avenues for future research and development in the field of Text-to-Face generation using GANs.

1. Multimodal generation is an exciting area of future research for Text-to-Face generation using GANs. By incorporating multiple modalities such as text and visual inputs, more realistic and diverse image synthesis can be achieved. This could lead to the development of personalized and interactive applications such as virtual assistants, video conferencing, and gaming. To achieve this, existing architectures for Multimodal generation need to be modified to be more efficient. Researchers could consider incorporating datasets such as Flickr-Faces-HQ (FFHQ) and VoxCeleb2 into their models.

2. Generating synthetic images can help address issues of bias and underrepresentation in datasets. Normalizing datasets by generating images of data-scarce populations like senior citizens or young children, DFGAN could help improve accuracy and reduce bias of AI models that use facial recognition, emotion detection or related computer vision applications.

3. A promising area of research is cross-lingual generation, where researchers could explore ways to generate images based on text descriptions in multiple languages. Currently, text-to-face GANs can only generate images based on textual inputs in the same language. Generalizing the model to work with multiple languages has significant implications for applications such as language translation and cross-cultural communication. Datasets such as Multi30k and MultiNLI can be used for this purpose.

4. Researchers could investigate the possibility of incorporating features that allow users to customize the generated facial image, such as changing its skin color or race. This has potential applications in fields such as fashion, where virtual try-on solutions could be enhanced by allowing customers to visualize themselves with different skin tones or facial features. The model could be trained on datasets such as the CelebA dataset, which includes annotations for facial attributes such as race and skin color.

5. Furthermore, collaborations with industry partners could help translate our research into practical solutions or applications. Partnering with gaming companies to develop custom avatar generation systems or collaborating with e-commerce companies to develop virtual try-on solutions are some ideal examples of this. Such potential partnerships could help advance the field of text-to-face generation and facilitate a smooth integration of it into various industries.

Reference

[1] Xi Chen et al. “Infogan: Interpretable representation learning by information maximizing generative adversarial nets”. In: *Advances in neural information processing systems* 29 (2016).

[2] Yunjey Choi et al. “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8789–8797.

[3] Min Jin Chong and David Forsyth. “GANs N’Roses: Stable, Controllable, Diverse Image to Image Translation (works for videos too!)” In: *arXiv preprint arXiv:2106.06561* (2021).

[4] Min Jin Chong and David Forsyth. “Jojogan: One shot face stylization”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 128–152.

[5] Paul Ekman and Wallace V Friesen. “Facial action coding system”. In: *Environmental Psychology & Nonverbal Behavior* (1978).

[6] Ian Goodfellow et al. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[7] 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*. 2019, pp. 4401–4410.

[8] Bing Li et al. “AniGAN: Style-Guided Generative Adversarial Networks for Unsupervised Anime Face Generation”. In: *IEEE Transactions on Multimedia* (2021).

[9] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014).

[10] Ori Nizan and Ayellet Tal. “Breaking the cycle-colleagues are all you need”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 7860–7869.

[11] Xuran Pan et al. “Contrastive Language-Image Pre-Training with Knowledge Graphs”. In: *arXiv preprint arXiv:2210.08901* (2022).

[12] Or Patashnik et al. “Styleclip: Text-driven manipulation of stylegan imagery”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2085–2094.

[13] Albert Pumarola et al. “Ganimation: Anatomically-aware facial animation from a single image”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 818–833.

[14] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *preprint arXiv: 1511.06434* (2015).

[15] Yujun Shen et al. “Interfacegan: Interpreting the disentangled face representation learned by gans”. In: *IEEE transactions on pattern analysis and machine intelligence* (2020).

[16] Upasna Singh et al. “Generative Adversarial Networks: A Survey”. In: (2021).

[17] Ralf C Staudemeyer and Eric Rothstein Morris. “Understanding LSTM–a tutorial into long short-term memory recurrent neural networks”. In: *arXiv preprint arXiv:1909.09586* (2019).

[18] Ming Tao et al. “Df-gan: A simple and effective baseline for text-to-image synthesis”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16515–16525.

[19] Hoang Thanh-Tung and Truyen Tran. “Catastrophic forgetting and mode collapse in gans”. In: *2020 international joint conference on neural networks (ijcnn)*. IEEE. 2020, pp. 1–10.

[20] Weihao Xia et al. “Tedigan: Text-guided diverse face image generation and manipulation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 2256–2265.

[21] Tao Xu et al. “Attngan: Fine-grained text to image generation with attentional generative adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1316–1324.

[22] Yutong-Zhou-cv. “Awesome-Text-to-Image: A Survey on Text-to-Image Generation/Synthesis.” In: (2021). URL: `https://github.com/Yutong-Zhou-cv/Awesome-Text-to-Image`.

[23] H Zhang, I Goodfellow, D Metaxas, et al. “Odena. Self-attention generative adversarial network”. In: *Proc. Int. Conf. Mach. Learn.* 2019, pp. 7354–7363.

[24] Han Zhang et al. “Self-attention generative adversarial networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 7354–7363.

Appendices

Lokmanya Tilak College of Engineering

Sector-4, Vikas Nagar, Koparkhairane, Navi Mumbai-400709

1st International Conference on Recent Trends in Multidisciplinary Research and Innovation (ICRMIR-23)

Participation Certificate

This is to certify that

Dr./Prof./ Mr./Ms. NITISH BHATTACHARJEE

has presented a paper titled GANs AND THEIR APPLICATIONS FOR FACE SYNTHESIS : A LITERATURE SURVEY

in the 1st International Conference on Recent Trends in Multidisciplinary Research and Innovation ICRMIR-23 organised by Lokmanya Tilak College of Engineering on March 17-18, 2023

Dr. Smita Ambarkar/ Dr. Rajeshree Shinde	Dr. Subhash K. Shinde	Dr. Vivek Sunnapwar
Conference Organizer	Conference Chairperson & Vice Prinicipal	Patron & Principal

Lokmanya Tilak College of Engineering

Sector-4, Vikas Nagar, Koparkhairane, Navi Mumbai-400709

1st International Conference on Recent Trends in Multidisciplinary Research and Innovation (ICRMIR-23)

Participation Certificate

This is to certify that

Dr./Prof./ Mr./Ms. VINEET DABHOLKAR

has presented a paper titled GANs AND THEIR APPLICATIONS FOR FACE SYNTHESIS : A LITERATURE SURVEY

in the 1st International Conference on Recent Trends in Multidisciplinary Research and Innovation ICRMIR-23 organised by Lokmanya Tilak College of Engineering on March 17-18, 2023

Dr. Smita Ambarkar/ Dr. Rajeshree Shinde	Dr. Subhash K. Shinde	Dr. Vivek Sunnapwar
Conference Organizer	Conference Chairperson & Vice Prinicipal	Patron & Principal

Lokmanya Tilak College of Engineering

Sector-4, Vikas Nagar, Koparkhairane, Navi Mumbai-400709

1st International Conference on Recent Trends in Multidisciplinary Research and Innovation (ICRMIR-23)

Participation Certificate

This is to certify that

Dr./Prof./ Mr./Ms. MIHIR DICHWALKAR

has presented a paper titled GANs AND THEIR APPLICATIONS FOR FACE SYNTHESIS : A LITERATURE SURVEY

in the 1st International Conference on Recent Trends in Multidisciplinary Research and Innovation ICRMIR-23 organised by Lokmanya Tilak College of Engineering on March 17-18, 2023

Dr. Smita Ambarkar/ Dr. Rajeshree Shinde	Dr. Subhash K. Shinde	Dr. Vivek Sunnapwar
Conference Organizer	Conference Chairperson & Vice Prinicipal	Patron & Principal

Lokmanya Tilak College of Engineering

Sector-4, Vikas Nagar, Koparkhairane, Navi Mumbai-400709

1st International Conference on Recent Trends in Multidisciplinary Research and Innovation (ICRMIR-23)

Participation Certificate

This is to certify that

Dr./Prof./ Mr./Ms. ROLAND DSOUZA

has presented a paper titled GANs AND THEIR APPLICATIONS FOR FACE SYNTHESIS : A LITERATURE SURVEY

in the 1st International Conference on Recent Trends in Multidisciplinary Research and Innovation ICRMIR-23 organised by Lokmanya Tilak College of Engineering on March 17-18, 2023

Dr. Smita Ambarkar/ Dr. Rajeshree Shinde	Dr. Subhash K. Shinde	Dr. Vivek Sunnapwar
Conference Organizer	Conference Chairperson & Vice Prinicipal	Patron & Principal

GANs and their applications for Face Synthesis: A Literature Survey

1st Nitish Bhattacharjee
Dept. of Electronics and Telecommunication Engineering
Xavier Institute of Engineering
Mumbai, India
nitish.rupak@gmail.com

2nd Vineet Dabholkar
Dept. of Electronics and Telecommunication Engineering
Xavier Institute of Engineering
Mumbai, India
vineetdabholkar2002@gmail.com

3rd Mihir Dichwalkar
Dept. of Electronics and Telecommunication Engineering
Xavier Institute of Engineering
Mumbai, India
mihirdichwalkar@gmail.com

4th Roland Dsouza
Dept. of Electronics and Telecommunication Engineering
Xavier Institute of Engineering
Mumbai, India
roland180102@gmail.com

***Abstract*—Generative Adversarial Networks also known as GANs were discovered by computer scientist Ian Goodfellow in 2014. GAN is an approach to generative modeling that employs competitive neural networks called Generator and Discriminator to generate a new set of data based on training data that is inseparable from it. The generator is analogous to a forger generating fake data and the discriminator is the detective whose main role is to differentiate input data as genuine or fake. Discriminative models in GANs play the role of a classifier. On the other hand, generative models generate realistic images from random samples such as random noise. Within less than a decade, the technology of GANs has witnessed rapid growth and has made a huge impact in the field of AI. With the continuous advancements in GAN technology, it is now possible to synthesize fake images or manipulate real images in such a manner that they are nearly indistinguishable to the human eye. In this paper, we explore GANs and their application in Face Synthesis by reviewing some state-of-the-art models that include GANimation, StyleCLIP, JOJOGAN, InterFaceGAN, GANs N Roses and AttnGAN.**

***Index Terms*—Artificial Intelligence, Deep Learning, Discriminator, Face Synthesis, Generative Adversarial Networks, Generative Learning, Latent Vector, Neural Networks.**

I. Generative Adversarial Networks (GANs) [14]

GANs are generative frameworks which can produce new data instances that, despite closely resembling the training data, are completely novel. Basically, GANs consist of two components: the generator and discriminator. The function of the Generator is to create data samples that resemble those from the problem domain while that of the Discriminator is to identify whether the input sample provided to it is produced by the generator (fake) or from the problem domain's dataset (real). [4] The term adversarial in this network arises from the fact both these networks are engaged in a competitive minimax game wherein each network tries to outperform the other. It is the objective of the Discriminator to maximize the possibility of accurately labeling the generated examples while the Generator tries to minimize the possibility of the Discriminator correctly classifying its generated examples as fake. The GAN is trained by first freezing the Generator parameters, performing a forward pass, and then back-propagating the error to update the Discriminator's parameters. The Discriminator is now frozen while the forward pass and backpropagation is performed to update the Generator parameters. The objective function is given as:

$$\min G \max DV(D,G) = Ex \sim \text{ Pdata } (x)[\log D(x)] + Ez \sim P(z)[\log(1 - D(G(z)))] \quad (1)$$

Since the inception of GANs a variety of researchers have made significant contributions to the development of improved GANs, including C-GAN, DC-GAN, BI-GAN,SR-GAN, INFO-GAN and CYCLE-GAN.

A. Conditional Generative Adversarial Networks (CGANs) [17],[12]

In CGAN images are generated by applying an additional conditional setting to a normal GAN. For the generation of images, 'Conditioning' is applied to Generator as well as the Discriminator over supplementary information, for example, as data or labels, it is able to generate the data point with the target label. The Conditional GAN differs from the conventional (Vanilla) GAN in that it can control the generation of data. By providing additional information Convergence will also be faster. Applications of CGANs are Image-to-Image translation, Video generation, Convolutional face generation, Text-to-image synthesis, etc. The Conditional GAN Minimax Loss Function is given as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x \mid y)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z \mid y)))] \quad (2)$$

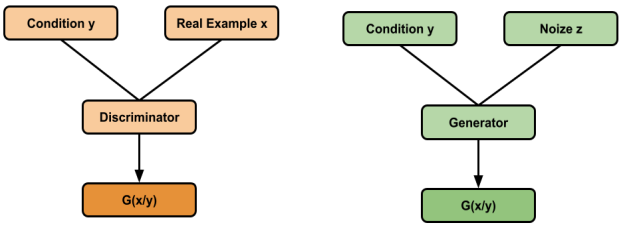

Fig. 1. Flow of Conditioanl GAN

B. Deep Convolutional Generative Adversarial Networks (DCGAN) [17]

Deep Convolutional GAN (DCGAN) uses convolutional layers in the Generator and transpose convolutional layers in the Discriminator. It is capable of high-resolution image generation from a limited set of images. The layers which DCGANs employ are not fully connected and use batch normalization[7] in both the generator and discriminator. In DCGAN all pooling functions are replaced with strided convolutions. A great deal of success has been achieved with DCGAN in generating new images. It helps to outpaint the image repeatedly up to a greater extent by creating larger, extended realistic images. DCGANs are introduced to tackle the problem of mode collapse.Mode collapse occurs when the generator gets over biased towards a some outputs and are not able to produce outputs of every variation from the dataset.

C. Information Maximizing Generative Adversarial Networks (Info-GAN) [17]

One drawback of the plain GANs is that we have no control over the images that GANs produce, so in Info-GAN to control the types of images produced, we need to feed some specific information with the random noises to the generator and force it to utilize the information when creating fake images. The additional information we are feeding should be related to the types of features we want the images to have. An "information theory" extension of the Generative Adversarial Network, InfoGAN can learn disentangled representations unsupervised. This is done by attaching a regularization term Lambda, it is the regularization constant and I(c;G(z,c)) term is the mutual knowledge between the latent code and the generator output.

$$\min_G \max_D V_{\text{infoGAN}}(D,G) = V_{GAN}(D,G) - \lambda I(c;G(z,c)) \quad (3)$$

with

$$V_{GAN}(D,G) \equiv \mathbb{E}_{x\sim p_{\text{data}}} \log D(x) + \mathbb{E}_{z\sim p_z(z)} \log(1-D(G(z,c))) \quad (4)$$

D. SRGAN (Super-Resolution GAN) [17]

The function of the Super Resolution GAN architecture is to recover fine textures from the image when we upscale it so that its quality doesn't degrade. It combines a deep network with an adversarial network to produce higher-resolution images. The SRGAN uses a DC (Deep convolutional) network with residual blocks. A feature loss and an adversarial loss form a part of the objective function to be optimized. During training, high resolution images are downsampled to low resolution images. The GAN generator then increases the number of pixels by upsampling the low-resolution images to super-resolution images. A discriminator distinguishes the high-resolution images and backpropagates the loss to update both the discriminator as well as the generator. While the generator and discriminator are trained in accordance with the conventional GAN procedure, SRGAN takes help of another loss function to reach the final point which is the perceptual loss function.

$$l^{SR} = \underbrace{\underbrace{l_x^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{adversarial loss}}}_{\text{perceptual loss (for VGG based content losses)}} \quad (5)$$

E. StyleGAN [9]

There are three main components of StyleGan : progressive growing,noise mapping network,adaptive instance normalization.It applies the basic progressive GAN architecture which implies that the size of the generated image incrementally increases from very low resolution to high resolution. For upsampling we use Bi-linear Sampling,it uses two new randomness references to generate a synthetic image. In Bi-linear sampling, we use all nearby pixel values to calculate the pixel value using linear interpolation. In the mapping network, we have eight layers and we will get an output of a dimension vector the same as the input . The output of the mapping network is going to each of the convolution layers which are present in our network. In Style GAN we can control the image synthesis using scale-specific modifications. Before applying AdaIn, each activation map is accompanied by Gaussian noise. Before passing through the AdaIN (Adaptive Instance Normalization)[6] module, mapping network output (w) is transformed by an affine transformation (A). This transforms the encoded mapping into the generated image. Further, the generator model is modified so that it no longer takes an input point from the latent space. To introduce style variation at a given level of detail, noise is introduced. Through regularization, the blocks and layers can localize the style to the specific areas of the model and level of detail. Image generation and style establishment are possible with StyleGAN.

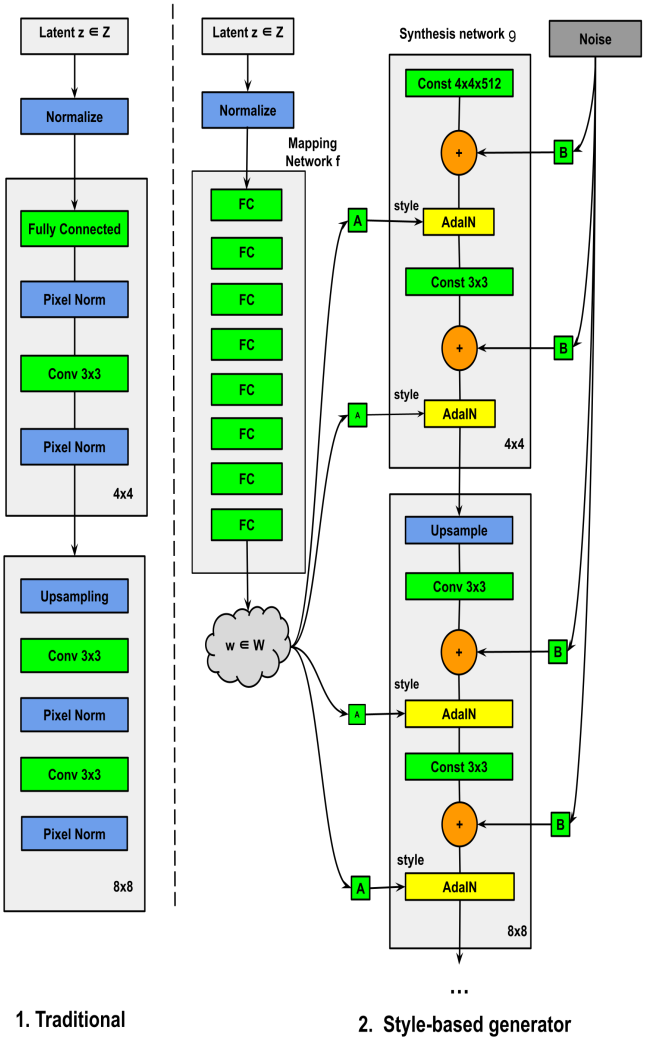

Fig. 2. Architecture of StyleGAN

II. Face Synthesis Models

A. GANIMATION [15] [20]

GANimation enables automatic animation of facial expressions in an image. This has found applications in the areas of movie production, photography, e-commerce, and many more. GANimation helps to edit the facial expressions of any image and can give us the output in the form of continuous expressions/ interpolating the expressions in a smooth transition/ animation (GIF) on the same image without the need for any training i.e. in an unsupervised manner. The desired expressions are generated using various Action Units (AUs). For any expression to be developed, the muscles of the face should be expanded or contracted, by using this phenomenon to generate a desired expression the corresponding action units are used. StarGAN[2] is one such example of editing an image or the expressions, but the main disadvantage was that it could only generate a few expressions that were present/ defined in the dataset. It failed to generate other varied expressions. This limitation of StarGAN was solved using the GANimation approach. It makes use of FACS, an acronym for Facial Action Coding System, to generate a variety of face expressions with the help of different facial AUs. In this, 30 different AUs were described for various facial muscles such as Inner Brow which uses AU1, Outer Brow which uses AU2, Jaw which uses AU26 etc. This model makes use of the EmotioNet dataset, from which 2,00,000 images were used.

1) Image adversarial loss: This is similar to the vanilla GAN loss, but instead of the conventional function which is JS divergence, WGAN-GP[5] is used which makes use of Earth Mover Distance.

$$\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[D_{\mathrm{I}}\left(G\left(\mathbf{I}_{\mathbf{y}_o}\mid\mathbf{y}_f\right)\right)\right]-\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[D_{\mathrm{I}}\left(\mathbf{I}_{\mathbf{y}_o}\right)\right]+\lambda_{\mathrm{gp}}\mathbb{E}_{\widetilde{I}\sim\mathbb{P}_{\widetilde{I}}}\left[\left(\left\|\nabla_{\widetilde{I}}D_{\mathrm{I}}(\widetilde{I})\right\|_2-1\right)^2\right] \quad (6)$$

2) Attention Loss: : A seam transition between the input image and the morphed image is ensured by this mechanism, which helps the model to give attention to the relevant portions/regions of the output image.

$$\lambda_{\mathrm{TV}}\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[\sum_{i,j}^{H,W}\left[\left(\mathbf{A}_{i+1,j}-\mathbf{A}_{i,j}\right)^2+\left(\mathbf{A}_{i,j+1}-\mathbf{A}_{i,j}\right)^2\right]\right]+\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[\|\mathbf{A}\|_2\right] \quad (7)$$

3) Conditional Loss: : This helps to verify/ check whether the target facial expression condition is satisfied or not.

$$\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[\|D_{\mathbf{y}}\left(G\left(\mathbf{I}_{\mathbf{y}_o}\mid\mathbf{y}_f\right)\right)\right]-\mathbf{y}_f\|_2^2\Big]+\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[\left\|D_{\mathbf{y}}\left(\mathbf{I}_{\mathbf{y}_o}\right)-\mathbf{y}_o\right\|_2^2\right] \quad (8)$$

4) Identity Loss: : It ensures that the original identity of the person is maintained in the new generated image.

$$\mathscr{L}_{\mathrm{idt}}\left(G,\mathbf{I}_{\mathbf{y}_o},\mathbf{y}_o,\mathbf{y}_f\right)=\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[\left\|G\left(G\left(\mathbf{I}_{\mathbf{y}_o}\mid\mathbf{y}_f\right)\mid\mathbf{y}_o\right)-\mathbf{I}_{\mathbf{y}_o}\right\|_1 \quad (9)$$

5) FullLoss: : For generating the final output/ desired image (Iyg), all previous partial loss functions are added together to build a combined loss function called the Full loss.)

$$\mathscr{L}=\mathscr{L}_{\mathrm{I}}\left(G,D_{\mathrm{I}},\mathbf{I}_{\mathbf{y}_r},\mathbf{y}_g\right)+\lambda_{\mathrm{y}}\mathscr{L}_{\mathbf{y}}\left(G,D_{\mathrm{y}},\mathbf{I}_{\mathbf{y}_r},\mathbf{y}_r,\mathbf{y}_g\right)+\lambda_{\mathrm{A}}\left(\mathscr{L}_{\mathrm{A}}\left(G,\mathbf{I}_{\mathbf{y}_g},\mathbf{y}_r\right)+\mathscr{L}_{\mathrm{A}}\left(G,\mathbf{I}_{\mathbf{y}_r},\mathbf{y}_g\right)\right)+\lambda_{\mathrm{idt}}\mathscr{L}_{\mathrm{idt}}\left(G,\mathbf{I}_{\mathbf{y}_r},\mathbf{y}_r,\mathbf{y}_g\right) \quad (10)$$

where λ_{idt}, λ_y and λ_A represent the hyper-parameters. They control the importance of every loss term.

B. JoJoGAN: One Shot Face Stylization [3]

JoJoGAN[6] uses the Style Transfer method which helps to transfer a facial image into a new art style learned from a reference image. Using a pre-trained StyleGAN2[9] and a GAN inversion procedure, JoJoGAN creates a paired dataset from a single style reference and fine-tunes it. Assume that, T is for GAN Inversion process, G is StyleGAN, s is the Style parameters in StyleGAN's S-space, θ for the parameters of the vanilla StyleGAN. JoJoGAN uses four steps:

1) GAN inversion: Inverting the reference style image 'y' yields a style code w = T(y), and from that the s parameters s(w).
2) Training set: Find a set of style codes S that are "close" to s. Pairs (si , y) for si ∈ S will be the paired training set.
3) Finetuning: StyleGAN is adjusted to get '$\hat{\theta}$' such that G(si ; $\hat{\theta}$) ≈ y.
4) Inference: For a given input 'u', the stylized face is G(s(T(u)); $\hat{\theta}$) (so G○ s ○T is the style mapper).

In short, StyleGAN2 takes the reference image and generates the most realistic image it can out of it. Then, it generates all the faces that look alike to the realistic reference image by choosing the closest neighbors in the latent space and pairs them up with the original reference to fine-tune another pure StyleGAN2 phase model. This fine-tuning process looks at a lot of generated realistic faces that look like the reference image and then tries to convert the realistic image into the reference image as close as it can. After the network gets used to generating reference images based on the realistic image, if a content image is passed through, it will convert it to the style similar to the reference images while not affecting the facial structure of the content image.

JoJoGAN's style mapper produces good-looking outputs, faithfully has features from the style reference and also preserves the person's identity. It uses FID which evaluates the diversity and quality of the images generated by JoJoGAN. The authors finetuned JoJoGAN for 200 to 500 iterations on Nvidia A40 depending on the reference using the Adam[1] optimizer at a learning rate of 2×10^{-3} which takes about 30 to 60 seconds

C. StyleCLIP: Text-Driven Manipulation of StyleGAN imagery [14]

StyleCLIP combines the generative power of StyleGAN[9] with CLIP's[10] image-text embedding to give very intuitive manipulation of images using text. CLIP is trained on full sentences and the intuition behind this is that the model can learn a lot more things from sentences and find patterns between images and texts. It consists of two encoders, one for text and the other for images. These encoders are trained to map images and corresponding text into a common latent space to check for similarity between them. Using StyleCLIP we can manipulate an image by using the guidance of only text to get photorealistic images. We get disentangled edits, for example, by typing "Mohawk Hairstyle" the face remains intact but the hair of the original face image is changed to a mohawk hairstyle. The authors investigated three techniques that combine CLIP with StyleGAN (StyleGAN2) which are as follows:

i) Text-guided Latent Optimization: The latent code is optimized so that it produces an image with high similarity to the target but remains close to the original one. Even though it takes a few minutes of optimization to apply manipulations on the image, this approach, for the most part, is very flexible. The first step in optimization is common for all three approaches. The image is inverted i.e. latent code which corresponds to the input image is found for the optimization and then duplicated and inserted into the pre-trained StyleGAN to obtain an image.

ii) Latent Mapper: A latent residual mapper is trained for a particular textual description. Provided with a beginning point in the latent space i.e. the input image to be manipulated, it outputs a local step in latent space. The first step in this approach is that the image is inverted and the latent code is inserted into a network known as the mapper which is text-specific and the cosine similarity is computed between all the image-text pairs. This then returns an offset which is added to the input latent code and entered into a pre-trained StyleGAN.

iii) Global Directions: Maps a text prompt to an input global direction in StyleGAN's style space. Using this we can control the degree of manipulation as well as the disentanglement. Here our goal is to find the global direction in style space such that by traversing along this direction the target attribute is modified for an arbitrary image. In order to apply the Global Direction method to different (image, text prompt) pairs, a single-time preprocessing is required

TABLE I
DIFFERENCES BETWEEN THE THREE METHODS FOR COMBINING STYLEGAN AND CLIP

	Pre-Proc (hours)	**Average Train Time (hours)**	**Average Test Time**	**Image as Input**	**Latent Space**
Text-guided Latent Optimization	-	-	98 sec	Required	W+
Latent Mapper	-	10 to 12	0.075 sec	Required	W+
Global Directions	4	-	0.072 sec	Not Required	S

The training and testing timings are for the model trained on a NVIDIA GTX 1080Ti GPU.

D. InterfaceGAN: Interpreting the Disentangled Face Representation Learned by GANs [16]

With InterfaceGAN, we can understand what knowledge GANs comprehend from the latent representation, and how this knowledge can be reused to produce output or for the overall generation process and how the different attributes are organized in the representation. When a latent representation is given, the GAN model governs the attributes which are required to output a face and distinguishes between the characteristic attributes of an old woman and a younger woman as an example. It helps to provide a way to understand how a GAN works, and how a GAN model understands what

it has learned in the latent representation (latent space) and combines them to form different random codes to generate a photo-realistic image. This also helps us to study disentangled representations learned by GAN and provides a pipeline to discover the properties of various facial representations, further it also helps to edit the images based on their pose, smile, age, gender, eyeglasses, etc. InterfaceGAN employs two different GAN models, i) PGGAN (Progressive Growing GAN) [10] and ii) StyleGan [8]. 1. PGGAN: PGGAN first creates a boundary/Hyperplane using SVM Classifiers to distinguish between different attributes, and further this information is used by StyleGan for editing images. 2. StyleGAN: Helps to edit a generated image or a real image whose latent code is obtained using GAN inversion. In this, it helps to edit the image w.r.t pose, smile, gender, age, and eyeglasses. 3. GAN Inversion: A typical GAN generator takes latent codes as input which means it cannot work with real images directly, thus here GAN inversion technique is employed which reverses the process and maps the image space to latent space. This is employed in two different ways, i) Optimization-based approach and ii) encoder-based approach.

E. GANs N Roses [3]

Image-to-Image translation is a process in which a given image is modified on the basis of a reference image. This paper discusses the rendering of an input face image to it's equivalent anime version. Content is defined as what changes when an image is geometrically manipulated while Style is that which remains constant. The model implements Control, Consistency and Coverage. Control means tweaking the generated images, Consistency meaning that the style code remains constant across a variety of different faces rendered to anime and Coverage i.e. any variation of an anime image can be generated by mixing content code and style code. A stark variation from other related models like CouncilGAN[13] and AniGAN[11] is that the GNR framework takes a single image and performs several geometric augmentations like scaling, rotating and cropping to generate multiple images as training examples instead of using different images for training. The Framework includes a combination of Encoder and decoder. The encoder separates the input augmented images into their corresponding set of content codes c(x) and style codes s(x). This extracted set of content codes is passed to the decoder along with style codes which are sampled from a Normal distribution of anime style codes sz. This applies a variety of styles to the facial image without altering it's core content. The Descriptive Discriminator plays an important role in diversifying the generated images as it learns the Standard Deviation across minibatches.

The Loss function to be minimized consists of three parts

1) Lscon: It ensures Style Consistency i.e. all the style codes must be identical for the augmented input images.
2) Ladv: It is the adversarial loss of the Discriminator combined with the Minibatch Standard Deviation to ensure output diversity.
3) Ladv: It is the adversarial loss of the Discriminator combined with the Minibatch Standard Deviation to ensure output diversity.

An innovative application of the GNR framework is that is very efficient at video translation without explicit training on temporal sequences.

F. AttnGAN [19]

Attentional Generative Adversarial Network (AttnGAN) generates photorealistic images of faces (or birds) using a detailed textual description. It consists of an Attention Network which helps to capture and generate word-level information or features along with sentence-level info. The AttnGAN consists of two major components i.e. a) The Attentional generative network and b) The Deep Attentional Multimodal Similarity Model.

1) Attention Generative Network: By giving attention to words that are more relevant to specific subregions, the attention model helps the generative network draw different subregions.The given textual description is input to a text encoder. This encoder is a Bi-directional Long Short Term Memory (LSTM)[18] which converts it to word level and sentence level features [4]. The sentence-level features are conditionally augmented using noise. They are then combined with a noise vector sampled from a normal distribution. This is then input into the Attentional Generative Network. The attention model has two inputs, the word features in addition to the image features from the preceding hidden layer of the model. It has a total of 'm' generators, the more the generators, more the computational load. It is seen that the three stages save resources as well as give better outputs. These generators take the hidden states as input and then pass it through a series of convolutional and upsampling layers which generate images of small-to-large scales in m stages.

The output features provided to the initial generator produce an image of low resolution. The word level features are converted into an attention matrix of words and their relevance or importance. This matrix is used as an input to the subsequent layers after the first set of Convolutional layers. Upsampling operations are performed and the resultant feature map is given to the next generator in order to produce another image with better resolution. This continues until we get a high-resolution image with focus on important features.

2) Deep Attentional Multimodal Similarity Model (DAMSM) : The final image is then given to an Image encoder which will convert it to features and compare the Image-text similarity with the word feature matrix. It calculates the DAMSM loss [4]. This loss acts as a measure of the level of similarity between the generated image and the input sentence which is done in a semi-supervised manner. In order to compute a fine-grained and refined loss for image generation, DAMSM learns two neural nets that map subregions of the image to a common semantic space.

TABLE II
COMPARISON BETWEEN FACE SYNTHESIS GAN

GAN models	**Function**	**Dataset**	**Additional Losses**
StyleCLIP	Text-based Image Manipulation	CelebA-HQ	CLIP Loss
JojoGAN	Style Transfer	FFHQ, CelebA-HQ	Identity, Perceptual Loss
GANS N' Roses	Image to Anime	selfie2anime 4000	Style and cycle consistency, Adversarial Loss
GANimation	Image and GIF editing	EmotioNet, RaFD	Image-adversarial, Attention, Conditional, Identity, Full Loss
InterFaceGAN	Image Editing	CelebA	Multitask Loss
AttnGAN	Text to Bird/Objects	CUB/COCO	DAMSM, Conditional, Unconditional losses

III. CONCLUSION

The generation of faces with great precision and realistic display is one of the greatest achievements of Generative Adversarial Networks. This paper provides a brief overview of GANs and their use in face synthesis applications. A wide range of face synthesis domains has been explored, including Style Transfer, Text Driven Manipulation, Facial Expression Animation, and DeepFakes.

Using this powerful technique, one can manipulate or generate faces and even video content in an extremely realistic manner, making it nearly impossible to distinguish whether something is fake or not. As technology continues to evolve, GANs will continue to improve their performance in not only the field of Face Synthesis but also in other fields.

REFERENCES

[1] Sebastian Bock, Josef Goppold, and Martin Weiß. "An improvement of the convergence proof of the ADAM-Optimizer". In: *arXiv preprint arXiv:1804.10587* (2018).

[2] Yunjey Choi et al. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8789–8797.

[3] Min Jin Chong and David Forsyth. "Jojogan: One shot face stylization". In: *European Conference on Computer Vision*. Springer. 2022, pp. 128–152.

[4] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[5] Ishaan Gulrajani et al. "Improved training of wasserstein gans". In: *Advances in neural information processing systems* 30 (2017).

[6] Xun Huang and Serge Belongie. "Arbitrary style transfer in real-time with adaptive instance normalization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1501–1510.

[7] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[8] 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*. 2019, pp. 4401–4410.

[9] Tero Karras et al. "Analyzing and improving the image quality of stylegan". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8110–8119.

[10] Tero Karras et al. "Progressive growing of gans for improved quality, stability, and variation". In: *arXiv preprint arXiv:1710.10196* (2017).

[11] Bing Li et al. "AniGAN: Style-Guided Generative Adversarial Networks for Unsupervised Anime Face Generation". In: *IEEE Transactions on Multimedia* (2021).

[12] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784* (2014).

[13] Ori Nizan and Ayellet Tal. "Breaking the cycle-colleagues are all you need". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 7860–7869.

[14] Or Patashnik et al. "Styleclip: Text-driven manipulation of stylegan imagery". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2085–2094.

[15] Albert Pumarola et al. "Ganimation: Anatomically-aware facial animation from a single image". In: *Pro-*

ceedings of the European conference on computer vision (ECCV). 2018, pp. 818–833.

[16] Yujun Shen et al. “Interfacegan: Interpreting the disentangled face representation learned by gans”. In: *IEEE transactions on pattern analysis and machine intelligence* (2020).

[17] Upasna Singh et al. “Generative Adversarial Networks: A Survey”. In: (2021).

[18] Ralf C Staudemeyer and Eric Rothstein Morris. “Understanding LSTM–a tutorial into long short-term memory recurrent neural networks”. In: *arXiv preprint arXiv:1909.09586* (2019).

[19] Tao Xu et al. “Attngan: Fine-grained text to image generation with attentional generative adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1316–1324.

[20] Zheng Yuan et al. “Attributes aware face generation with generative adversarial networks”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 1657–1664.