American International University-Bangladesh (AIUB)

Face Generation and Discrimination Using Generative Adversarial Network (GAN)

Generation and Discrimination of the Human Face Using

Generative Adversarial Network and Convolutional Neural Network

Chisty, Md Muzammel Hossain (18-36108-1)

Shohag, Koushikur Islam (18-36587-1)

Jahan, Nushrat (18-36586-1)

Khan, Asef Hossain (18-36263-1)

A Thesis submitted for the degree of Bachelor of Science (BSc)

in Computer Science and Engineering (CSE) at

American International University Bangladesh in June, 2021

Faculty of Science and Technology (FST)

Abstract

As Generative Adversarial Networks is one of the most creative deep learning model, it has been extensively studied in recent years. Mainly, its most significant impact has been in computer vision and natural language processing. In Generative Adversarial Network, there are two neural networks which acts like player of a game where one agent's gain is another agent's loss. This act results in more accurate output. It kept a great impact on many challenging problems specially in the area of computer vision like image-to-image generation, facial attribute manipulation and similar domains. Though it has achieved success in challenging problem of computer vision it still faces many challenges in real world problems like generating high quality images, diversity in images, stable training and many more. Though it has many limitations, these can be outgrown. In this paper we used generative adversarial network to generate faces from dataset using two neural networks generator and discriminator. The aim of this paper is to prove using generative adversarial network it is possible to create random faces which neither exists in the real world as well as in the dataset too. Here we shared our considerable success in generating fake images and some idea for future work.

Declaration

This thesis is our unique work, and we certify it. No component of this has been submitted elsewhere, either in part or in its entirety, for the award of another degree. The use of any material in this thesis has been appropriately acknowledged. We affirm that this thesis contains no content that divulges any organization's or connected parties' secrets. The American International University-Bangladesh (AIUB) will be renamed American International University-Bangladesh (AIUB).

. . . .
Nushrat
Nushrat Jahan
18-36586-1
Department of Computer Science
American International University-Bangladesh

. . . .
Shohag
Koushikur Islam Shohag
18-36587-1
Department of Computer Science
American International University-Bangladesh

. . . .
Asef
Asef Hossain Khan
18-36263-1
Department of Computer Science
American International University-Bangladesh

. . . .
Chisty
Md Muzammel Hossain Chisty
18-36108-1
Department of Computer Science
American International University-Bangladesh

Approval

The thesis titled **"Face Generation and Discrimination UsingGenerative Adversarial Network (GAN)"** has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfilment of the requirements for the degree of Master of Science in Computer Science on **June 16, 2021** and has been accepted as satisfactory.

. . . .

Dr. Debajyoti Karmakar
Assistant Professor
Md. Asiful Islam
Lecturer
Department of Computer Science
American International University-Bangladesh

. . . .

Dr. Dip Nandi
Associate Professor, Director
Department of Computer Science
American International University-Bangladesh

. . . .

Dr. Md. Mahbub Chowdhury Mishu
Assistant Professor and Head-in-charge (UG)
Department of Computer Science
American International University-Bangladesh

. . . .

Professor Dr. Tafazzal Hossain
Dean
Faculty of Science & Information Technology
American International University-Bangladesh

. . . .

Dr. Carmen Z. Lamagna
Vice Chancellor
American International University-Bangladesh

Acknowledgments

This thesis is the result of our cooperative efforts, as well as the initiative and consistent motivation of our supervisors. But first, we'd like to express our gratitude to Allah, who has given us the opportunity to work on this thesis over the past semester. Dr. Debajyoti Karmakar and Md. Asiful Islam of American International University-Bangladesh [AIUB] deserve special gratitude for their unwavering support, inspiration, and constructive criticism.This thesis would not be feasible without their great supervision and unwavering support. We are extremely appreciative for the opportunity to collaborate with them. We also want to express our gratitude to Dr. Carmen Z. Lamagna, our Vice Chancellor, Prof. Dr. Tafazzal Hossain, our Dean, Prof. Dr. Tafazzal Hossain, our Associate Dean, Mashiour Rahman, our Director, Dr. Dip Nandi, and our Head of Department, Dr. Mahbub Chowdhury Mishu, for their constant encouragement and support.
Finally, and most importantly, we thank our honorable parents for teaching us in both the arts and sciences, as well as for their unwavering support and encouragement in pursuing our passions, even when they crossed boundaries.

Keywords

Machine learning, GAN, DC GAN, Residual Blocks, Face Generation

Contents

List of Figures

List of Abbreviations and Symbols

Here are all the abbreviations that is used in this document.

Abbreviations	
GAN	Generative Adversarial Networks
DCGAN	Deep Convolutional Generative Adversarial Networks
ICGAN	Invertible Conditional Generative Adversarial Networks
CycleGAN	Cycle-consistent Generative Adversarial Networks
StackGAN	Stacked Generative Adversarial Networks
TAC-GAN	Text Conditioned Auxiliary Classifier Generative Adversarial Networks
NN	Neural Network
CNN	Convolution Neural Network

Chapter 1

Introduction

1.1 Generative Adversarial Networks

Generative Adversarial Networks (GANs) are a class of generative models which are very powerful. It was initially introduced for working with image related works such as image generation, image in-painting, image super-resolution and text to image conversion [Goodfellow et al., 2014] [Denton et al., 2015] [He et al., 2016] . Our GAN model consists of Generator and Discriminator. The Generator generates fake data from noise generated based on a specific seed vector. While The Discriminator provides probabilistic feedback that how good the generated data resembles a real distribution. The purpose of the Generator is to create fictitious data samples in order to deceive the Discriminator. On the other hand, the Discriminator seeks to discriminate between actual and fraudulent samples [Radford et al., 2015]. Both the Generator and the Discriminator are Neural Networks, and throughout the training phase, they compete with each other.

Most of the typical neural networking systems were easily fooled by adding small amount of noise to the original set of data. But in terms of GAN the Generator tries to outsmart the Discriminator and The Discriminator tries to provide probabilistic result of comparison between fake and real data at the same time. As this generator and discriminator both are neural networks and works in a competitive environment of each other during training phase so they get better and better in their particular task after repetition. GAN is basically a subset of Neural Network which follows the unsupervised learning approach where the discriminator develop their training data set with real image samples and the generator trains on the discriminator. GAN uses Convolutional Neural Network (CNN) modelling for filtering images and detecting patterns. GAN is used in Photo Blending, Face Aging, Text-to-Image Translations, Generating Cartoon Characters etc. Again, GAN can be used to creates videos given a photo, 3D object generation and style translations. GAN was heavily used in our thesis to generate random human faces which looks like real human faces.

Chapter 2

Literature review

GAN is a form of neural network architecture for generative modeling. It is a powerful way to learn the types of data distribution when used unsupervised, and it has come a long way over the years. GANs are inspired by the zero-sum of game theory. A non-cooperative zero-sum game is defined as a game in which two parties strongly oppose each other. Here, a gain on one side inevitably leads to a loss on the other side, and the gain on both sides is zero [Gong et al., 2017].

2.1 Overview

In order to build a game world, many games nowadays rely on some form of algorithm-based procedural content generation mechanism. Some of these techniques are commonly used to construct terrains and trees, for example, but there are no restrictions on what may be built, which includes planets and monsters. Another intriguing application of procedural content generation techniques is the creation of dungeons, especially for the rogue-like game genre.

Despite the fact that these methods are somewhat different in terms of implementation, they all have one thing in common: they're all based on a rule system, and the overall goal is to duplicate patterns in a seemingly random manner in order to emulate our view of nature as discretely chaotic. Although some of these implementations do involve some unpredictability, this is usually kept to a minimal. In procedurally generated game environments, pseudo-randomness is frequently done by depending on seeds, which are a collection of static numbers used in a manner similar to how Perlin noise is formed. The seed is supposed to have a constant value across multiple playthroughs in a game world created by it. It is technically possible to forecast what will be generated if the seed and the rules that govern content production are known. The problem with material generated utilizing Perlin Noise techniques, despite their potency, is that designers have very little control over the generated content. Developers have the ability to adjust the rules for content creation, however traditional editing tools are preferred for customizing the game world to one's heart's content.

2.2 GAN-based image synthesis

Photo-realistic picture synthesis and manipulation are now possible thanks to generative adversarial networks (GANs). However, because large-scale generators (e.g., StyleGAN2) have a significant computational cost, it takes seconds to view the effects of a single edit on edge devices, making interactive user experience impossible.
Image synthesis, the fabrication of visuals using computer algorithms, is the most well-known example of content generation in popular culture. Over the last few decades, the capacity of machines to generate images has progressed well beyond predictions. As a result of these advancements, researchers have begun to examine the methodologies used to create these images, allowing for a more precise categorization system, which will be used to explain the many approaches to picture synthesis using machine learning. There are four generation strategies, according to " Image Synthesis Using Machine Learning Techniques." [Guérin et al., 2017]. The Figure 2.1 represents the classification of GAN based image synthesis.

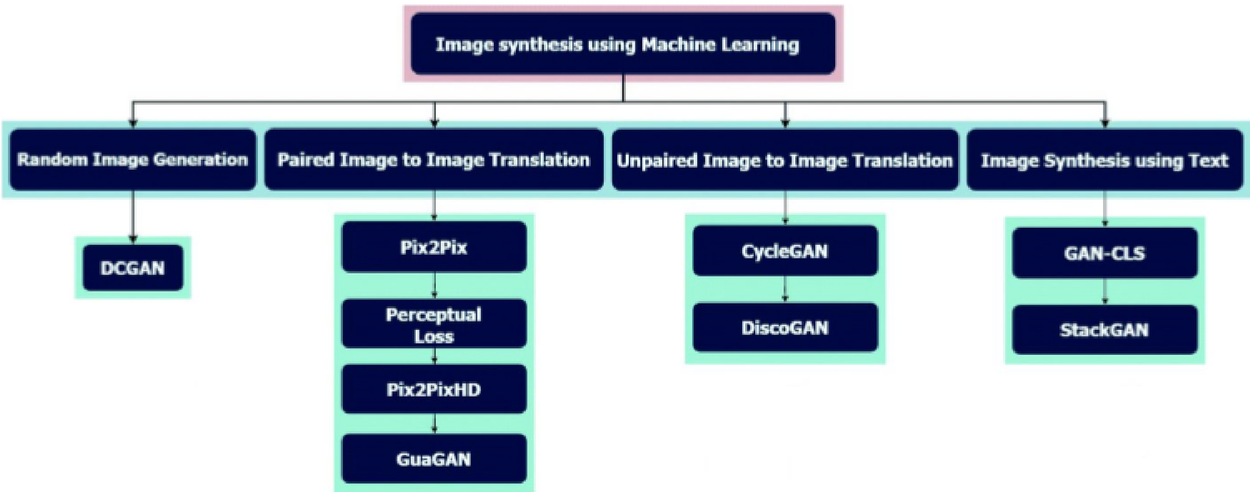


Figure 2.1: Classification of image synthesis,collected from Image Synthesis Using Machine Learning Techniques, 2019

2.2.1 Random Image Generation

The creation of a random image of a specific class. A random image generator that has been trained on a series of photographs of real faces will create realistic images of new faces that the generator has never seen before. The technique's main drawback is that it necessitates a huge training set. GANs were utilized to generate fresh realistic examples for the MNIST handwritten digit dataset, the CIFAR-10 small object picture dataset, and the Toronto Face Database in the original paper by Ian

Goodfellow et al. in their paper “Generative Adversarial Networks” [Goodfellow et al., 2014].The Figure 2.2 represents some examples of GAN based random image generation.

Figure 2.2: GAN examples for creating new plausible examples for picture collections,collected from Generative Adversarial Network, 2014

A model named DCGAN is used to demonstrate how to train stable GANs at scale in order to create instances of bedrooms in “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks” [Radford et al., 2015].The Figure 2.3 shows bedrooms which are generated by GAN.

Figure 2.3: GAN examples for creating new plausible examples for bedrooms,collected from Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015.

The DCGAN’s ability to perform vector arithmetic in the latent space between two inputs is a noteworthy achievement.The Figure 2.4 shows vector arithmatic examples which are generated by GAN.

It is demonstrated in “Progressive Growing of GANs for Improved Quality, Stability, and Variation” [Karras et al., 2017] that realistic images of human faces may be generated. The model was trained

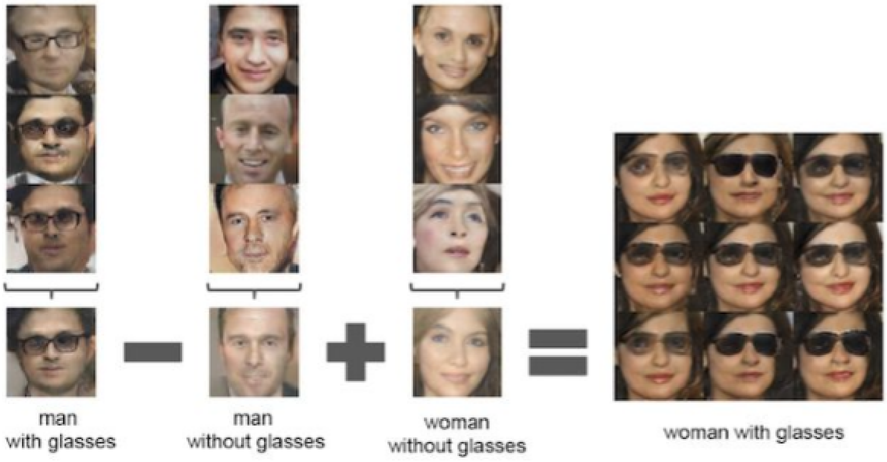


Figure 2.4: Vector Arithmetic examples for GAN-Generated Faces, collected from Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015.

using celebrity physical appearances, which means that the created faces contain features from existing well-known individuals, making them feel weirdly familiar to a degree. The Figure 2.5 shows photorealistic face examples which are generated by GAN.

Figure 2.5: Photorealistic Faces using GAN, collected from Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2017.

It is also demonstrated in “Towards the Automatic Anime Characters Creation using Generative Adversarial Networks” [Jin et al., 2017] that GANs may be used to produce the faces of anime characters (i.e. Japanese comic book characters).Figure 2.6 shows example of GAN-Generated Anime Character Faces.

Figure 2.6: Anime Faces generated using GAN, collected from Towards the Automatic Anime Character Creation with Generative Adversarial Networks, 2017.

2.2.2 Image to Image Translation in Pairs

When paired photos from both sets are available, this method is used to synthesize an image from one category or set using an image from another category or set. GANs can also be utilized for Image-to-Image Translation, as shown in “Image-to-Image Translation using Conditional Adversarial Networks” [Isola et al., 2017]. GANs are used to produce frontal-view pictures of human faces from photographs taken at an angle in “Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis” [Huang et al., 2017]. Figure 2.7 shows an example of GAN-based Face Frontal View.

Figure 2.7: Face frontal view generated using GAN, collected from Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis, 2017.

A GAN model is utilized in “Unsupervised Cross-Domain Image Generation” [Taigman et al., 2016] to convert images from one domain to another, such as street numbers to MNIST handwritten digits and celebrity photographs to emojis or little cartoon faces. Figure 2.8 shows an example of celebrity photographs and emojis using GAN.

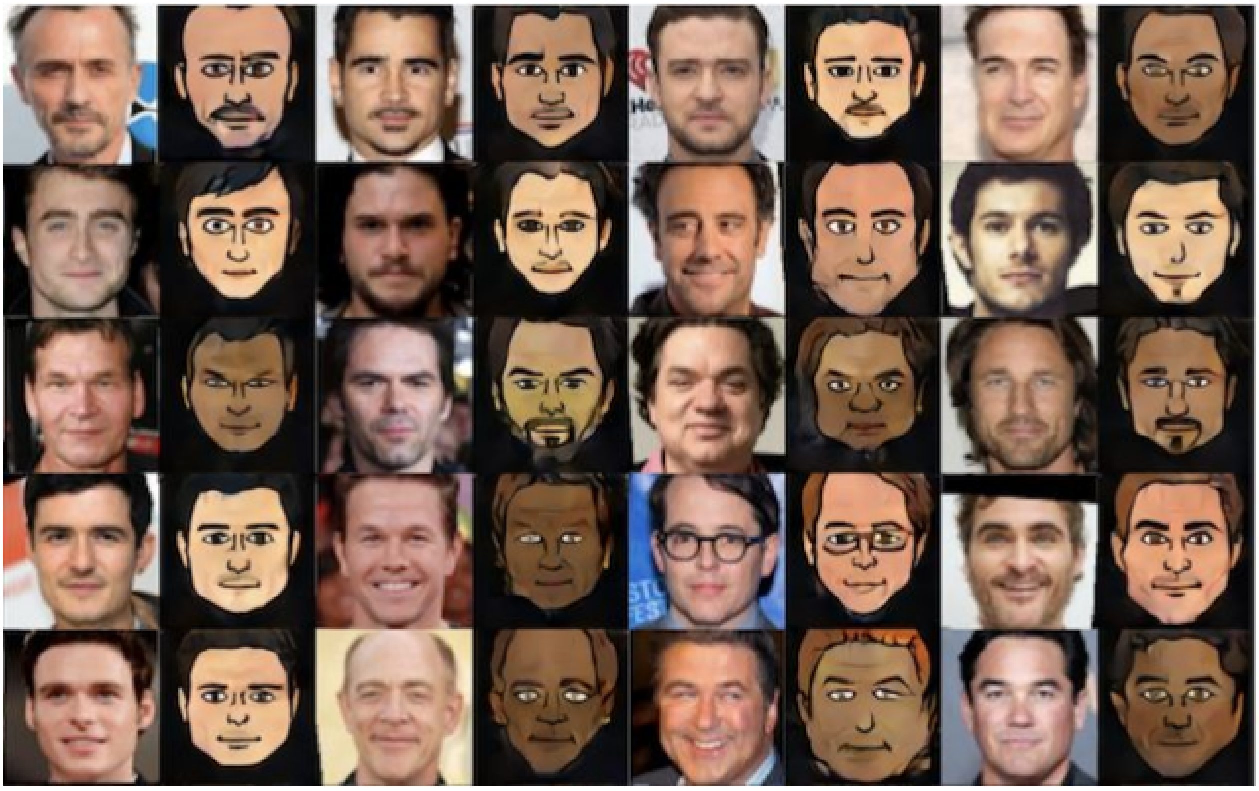


Figure 2.8: Celebrity Photographs and Emojis using GAN, collected from Unsupervised Cross-Domain Image Generation, 2016.

A model called IcGAN is used to reconstruct images of faces with specified properties, such as changes in hair color, style, facial expression, and even gender, in "Invertible Conditional GANs For Image Editing" [Perarnau et al., 2016]. Figure 2.9 shows an example of face photo editing using GAN.

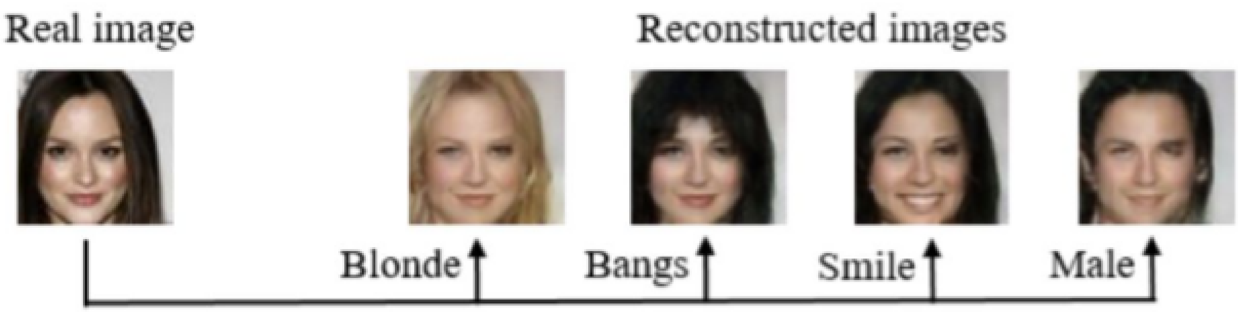


Figure 2.9: Face Photo Editing using GAN, collected from Invertible Conditional GANs For Image Editing, 2016

The production of faces with specific attributes such as hair color, facial expression, and glasses is explored in "Coupled Generative Adversarial Networks" [Liu and Tuzel, 2016]. Images with a variety of hue and depth are also created.Figure 2.10 shows an example of Generated Faces With and Without Blond Hair using GAN.

Figure 2.10: Generated Faces With and Without Blond Hair using GAN, collected from Coupled Generative Adversarial Networks, 2016.

A face photo editor is provided in "Neural Photo Editing using Introspective Adversarial Networks" [Brock et al., 2016] using a hybrid of variational auto-encoders and GANs. The editor enables for quick and realistic changes to human faces, such as hair color, hairstyles, facial expressions, positions, and the addition of facial hair.Figure 2.11 shows an example of Generated Faces With and Without Blond Hair using GAN.

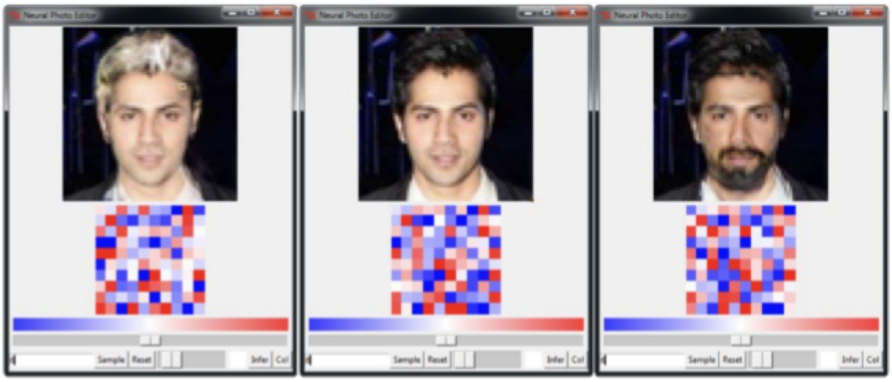


Figure 2.11: Generated Faces With and Without Blond Hair using GAN, collected from Coupled Generative Adversarial Networks, 2016.

2.2.3 Image to Image Translation without Pairing

When paired data from the input and target sets aren't available for training, this method is used. For training, images from both sets must be used, but each input image does not need to have a corresponding target image in the training data-set. The CycleGAN model delivers an excellent set of image-to-image translations in “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks” [Zhu et al., 2017]. Figure 2.12 shows an example of Translations of Four Image-to-Image using CycleGAN,.

Figure 2.12: Translations of Four Image-to-Image using CycleGAN, collected from Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, 2017

2.2.4 Image Synthesis from Text description

Used to synthesize images using a text description of the content of the image to be synthesized. When a thorough description of the image of a certain class is provided to image synthesis using text models, the model will produce an image of that particular class. A model for synthesizing images of birds, for example, can be developed utilizing a detailed description of the species. For training, images of a certain class must be provided coupled with a text description. It is proved in “StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks” [Zhang et al., 2017] that GANs can be used to generate realistic-looking images from textual descriptions of basic items such as birds and flowers. Figure 2.13 shows an example of Textual Descriptions and Photographs of Birds using GAN.

A fascinating example of text to image production of small items and scenes, including birds and flowers, may be found in “Generative Adversarial Text to Image Synthesis” [Reed et al., 2016b]. Another model is trained on the same dataset and produces comparable findings in “TAC-GAN – Text

Figure 2.13: Textual Descriptions and Photographs of Birds using GAN, collected from Stack-GAN:Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, 2016

Conditioned Auxiliary Classifier Generative Adversarial Network" [Dash et al., 2017]. The ability of GANs to generate images from text is expanded in "Learning What and Where to Draw" [Reed et al., 2016a] by employing bounding boxes and key points as recommendations as to where to draw a described object. Figure 2.14 shows an example of Photos of Object Generated From Text and Position Hints With a GAN

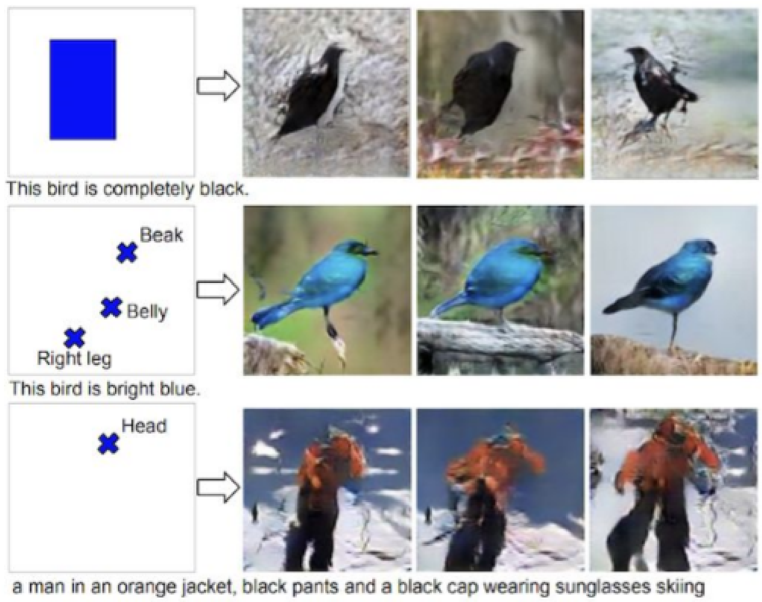


Figure 2.14: Photos of Object Generated From Text and Position Hints With a GAN collected from Learning What and Where to Draw, 2016

2.3 GAN-based music synthesis

Music generation is another fascinating achievement in the subject of machine learning, albeit it is not directly related to the scope of this research. MuseNet, a neural network trained to create the next note in a sequence given an input sequence, is a good illustration of this. This is based on a model called Sparse Transformer, which uses a concept known as ”sparse attention” to reassemble sequences. Sparse Transformers can use the latter to create not only music, but also graphics or any data set.

2.4 Terrain Generation by GAN

Terrain generation has also been done with GANs. Conditional GANs were used to build terrains using sketches in “Interactive example-based terrain authoring with conditional generative adversarial networks,”(Guérin É. et al., 2017) [Guérin et al., 2017] which is comparable to the goal of this study. Rivers, mountains, and other objects can be drawn and converted into a three-dimensional terrain. Another noteworthy application of GAN-generated height-mapping is the usage of data from NASA’s Visible Earth project. The latter example uses data from satellite pictures to depict the morphology of the Earth. After that, a DCGAN generates height maps and a Pix2Pix generates textures using these inputs.Figure 2.15 shows an example of Generating Terrains using a GAN.

Figure 2.15: Generating Terrains using a GAN, collected from Interactive example-based terrain authoring with conditional generative adversarial networks, 2017

Chapter 3

Methods

3.1 Introduction

From the literature review we can analyze the uses of GAN. As we have learnt a lot about GAN, now we are going to implement it to generate random human faces. So, our paper follows experimental methods. Though GAN has different applications, this paper focuses on face generation.
In order to generate faces we are going to use two Neural Networks, one for discriminator and another for generator. Here Generator will try to create random human faces using given random matrix or vector. On the other hand, Discriminator will try to detect whether generator's output is real or fake. That means discriminator will classify whether the generated human faces are real or fake. Using CNN discriminator will a give a single output for every single images. Figure 3.1 shows the flow of discriminator and figure 3.2 shows the flow of generator.

We initially chose DCGAN approach to generate faces. DCGAN uses transposed convolution technique to perform up-sampling of 2D images. However the model evolved to contain Residual Blocks. We want to work with human faces, so we used human faces as our training dataset.

3.2 Training Dataset

In order to train our datasets we needed a dataset to train. We used kaggle dataset [kaggle, 2021] which contains human faces with plain background. The dataset contained many images of a same human from slightly different angles and with slight expression changes. Other than the dataset gathered from kaggle we added some more face images. Most of the collected images had a plain background. The accuracy mostly depends on the training datasets. It is very useful to have a very large dataset. As for the dataset as a whole, the original dataset came from [kaggle, 2021] to which we added a few hundred more of our images, which was then augmented on a duplicate basis. The whole operation was performed in Google Colab, using google drive as the storage medium. The data was generated for 64x64 pixel images and the generation was done based on a specific seed size.

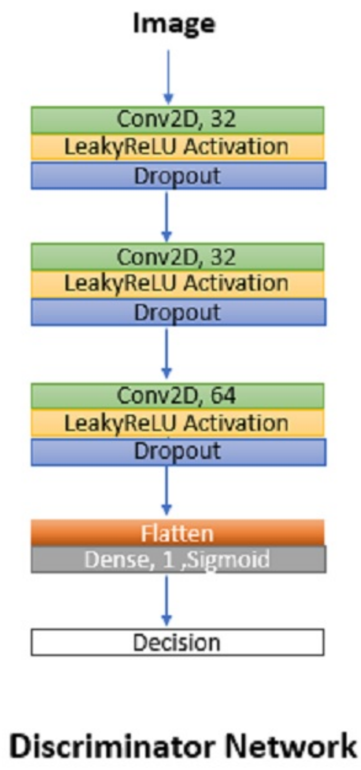


Figure 3.1: Flow of GAN-discriminator

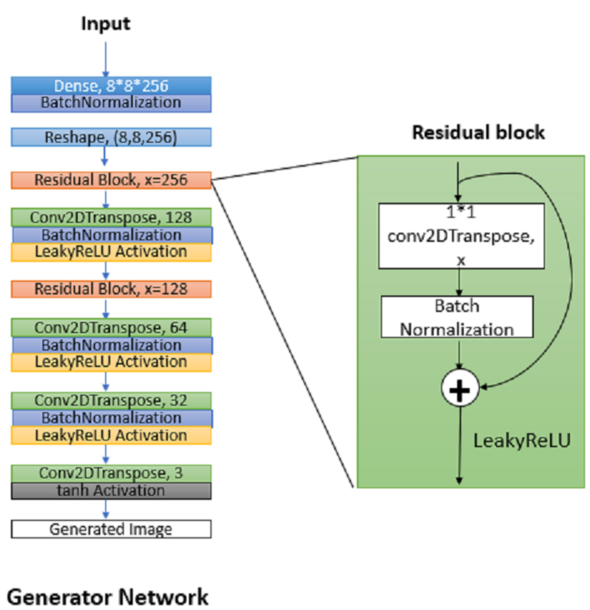


Figure 3.2: Flow of GAN-generator

3.2.1 Improving the Data set

In order to improve the dataset we augmented the dataset from 8000 to 32000 by concatenating the dataset with itself. This produced better results than the original 8000 dataset. Issues regarding over-fitting didn't arise due to the fact that our discriminator is supposed to recognize the dataset faces. In order to reduce processing cost we had to normalize the data-set. For uniform shuffling the buffer size was kept larger than the dataset size. The resolution of the images had to re-scaled into 64*64 pixels. The NumPy image array data had to be converted to float32.

3.3 Procedures

Firstly, we normalize the images. Then we have to create the discriminator, the generator network, and the full training loop. Then we will have to train the network to generate new faces. While implementing GAN, the training of generator and discriminator happens one after another. Like the discriminator is trained for a step then the generator is trained for a step and the whole process repeats again and again.

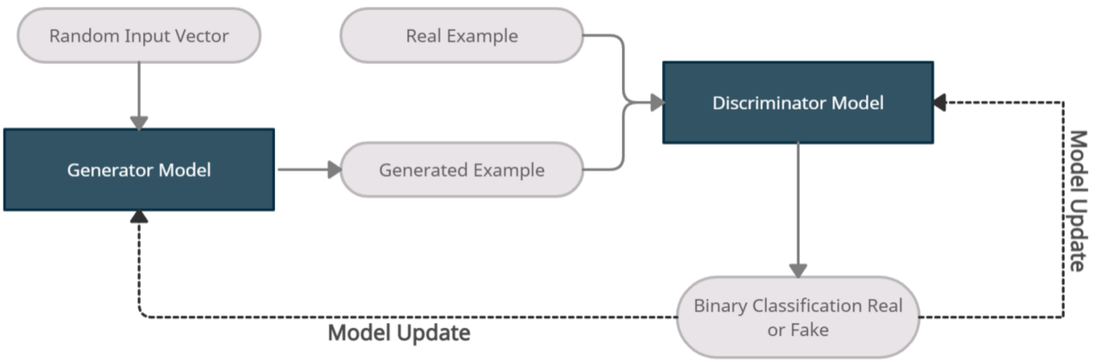
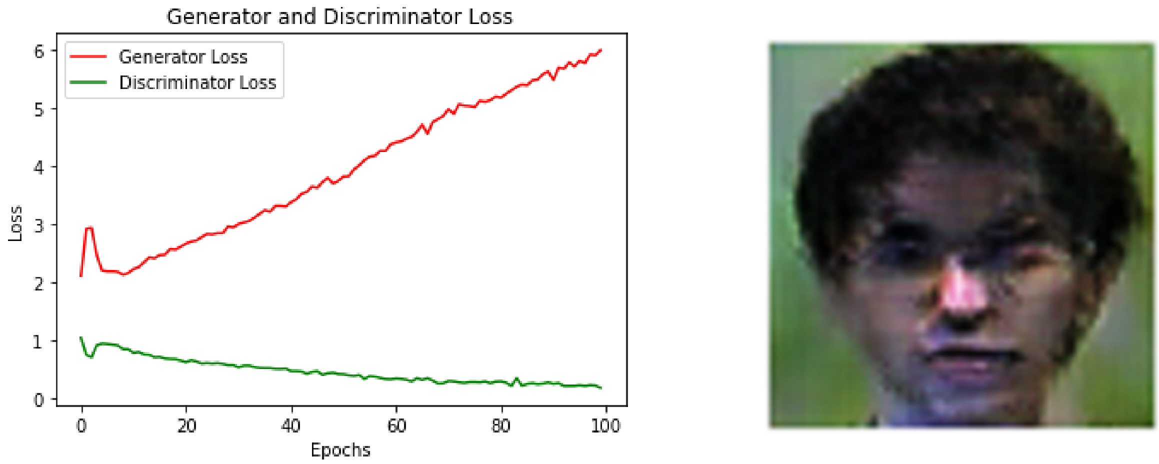


Figure 3.3: Flow of generating output

3.4 Evolution of Model

Initially the model was based on [Jeff Heaton, 2021]. Which is essentially a pure DCGAN model. And the model was designed to produce human faces but it was not that sophisticated it was only for learning purposes. In the figure 3.4 it can be seen that the Generator loss is on a high slope and dramatically increasing. Though the discriminator loss is decreasing gradually.

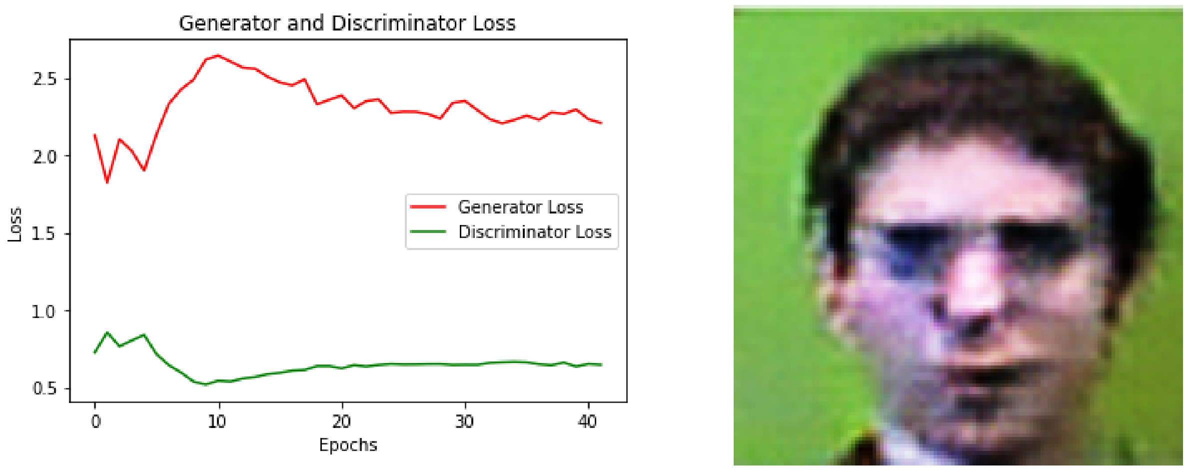


(a) Model Loss Graph

(b) Model Output

Figure 3.4: Initial DCGAN

In order to improve on this model we merged this initial DCGAN model with the model from keras GAN documentation [tensorflow, 2021]. After we merged the model the generator and discriminator loss was stabilized it was not increasing or decreasing as it was seen in the initial DCGAN model figure 3.5.

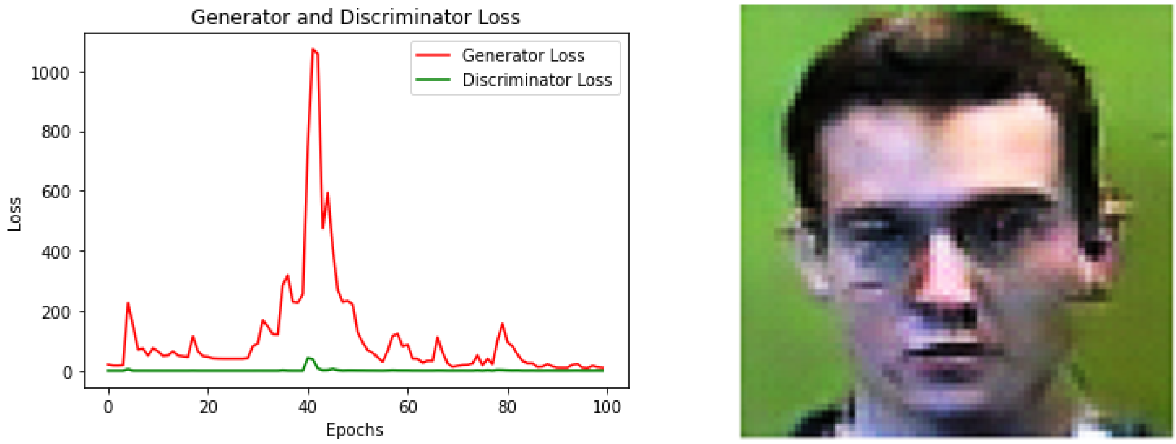


(a) Model Loss Graph

(b) Model Output

Figure 3.5: Initial DCGAN with Keras Implemented

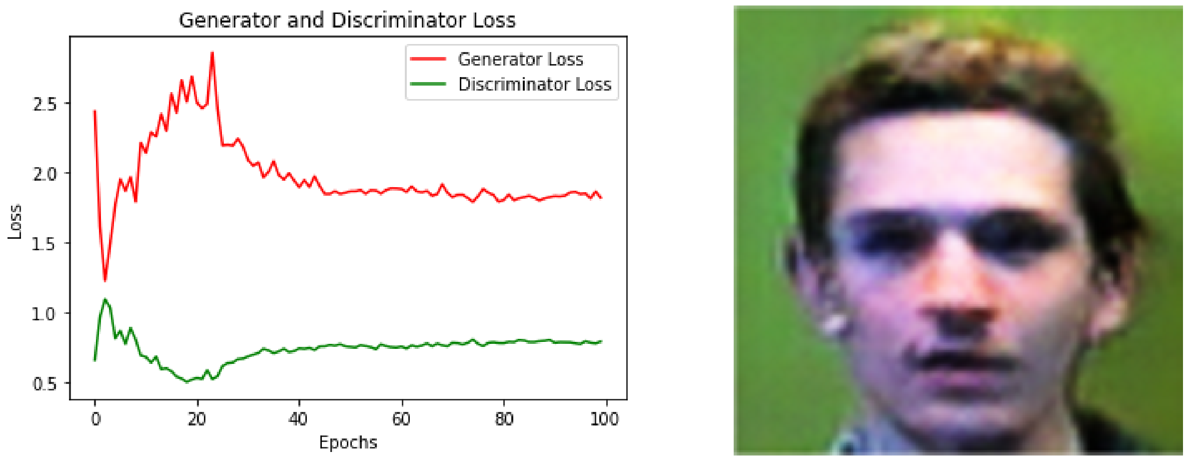
For further improvement Residual Blocks was introduced to the model. Before applying Residual Blocks the model was using the keras sequential model. However in order to add the Residual Blocks we felt the need to convert the models into a functional one. After which the residual blocks were added. In this point the Residual Blocks were working with ReLU activation layers. There was a high fluctuation noticed in the loss graph of Generator but the output was better than those previous models figure 3.6.



(a) Model Loss Graph (b) Model Output

Figure 3.6: Residual Block Implemented with ReLU Activation

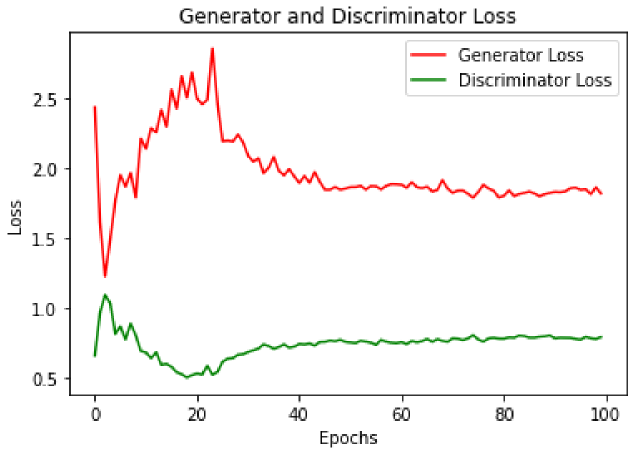
As it was seen a high fluctuation of loss rate in generator of Residual Blocks with ReLU activation we decided to change the activation function to LeakyReLU because LeakyReLU is more balanced and learns faster than ReLU. In the figure 3.7 it could be noticed that after some epoch both discriminator and generator loss was stabilized thus the output generated were way better than those of previous ones.



(a) Model Loss Graph (b) Model Output

Figure 3.7: Residual Block Implemented with LeakyReLU Activation, Learning Rate Constant

In the Runtime of the current model it was noticed that at some point when the generator and discriminator loss would stabilized, their improvement became inconsistent. As such their learning rate of Adam optimizer was reduced based on the epoch count. Thus the output was improved as seen on the figure 3.8



(a) Model Loss Graph

(b) Model Output

Figure 3.8: Residual Block Implemented with LeakyReLU Activation, Adaptive Learning Rate

Chapter 4

Results or findings

4.1 Introduction

Generative Adversarial Network works on the principle of unsupervised learning so the network learns itself from the training dataset to generate images. The primary two networks for generating newer face are generator network and the discriminator network. First the discriminator network is trained with fake images data so it can better distinguish between the fakes and real images. To do so a training data set is needed with human faces in different variation. A training date set is created for the task and is processed for the training and generating the output from the networks. This data set helps the discriminator to better distinguish between real and fake faces, and the generator to better produce or generate fake faces to fool the discriminator. Thus, this competition between both the networks produces the expected output of new human face.

4.2 Data Set Collection

The main target of the thesis was to generate a new human face. We used Deep Convolutional Neural Network for generating those faces with the help of two main networks, the generator network and the discriminator network. As we have already discussed that the technology used for this face construction is Generative Adversarial Networks (GAN) follows an unsupervised learning so each network generator and discriminator consists of two data sets. A training data set needed to be developed for discriminator to discriminate between real and fake images.

Considering to create a newer face a dataset with human faces were collected from different sources. Approximately 8000 samples of humans face with different background and postures were taken for the dataset with people of different race, complexion, gender, age and shape. This dataset helped enrich our training of the unsupervised learning of the discriminator network to distinguish between real and fake faces and the generator to create better fake images. Following are a few samples from the dataset gathered from an online source named “Kaggle” [kaggle, 2021]. However a larger dataset would have helped a lot more. The Figure 4.1 represents some figure from dataset.

Figure 4.1: Sample Dataset Collected from Online

To enrich the dataset more variation of faces dataset was developed for the thesis. A combined dataset has been developed for the research with the dataset previously mentioned collected from Kaggle and a newer one. A collection of approximately 300 images were collected with different variation of faces to combine then with the data set collected from Kaggle for better output. Few of the sample images are shown from the newer dataset in the figure 4.2.

Figure 4.2: Newly Introduced Dataset

4.3 Data Processing

Collecting data for training the data set needs processing to work with. As this thesis work is related to generation of new human faces in an unsupervised learning approach images taken for the data set needs processing for the network. To process the data a few aspects, need to be considered:

- Image Size
- Image Resolution
- Color Shade
- Storing Data Type
- Resize Method
- Reshape Method

The collected images were stored locally in a machine to read. First, the images were read from the local storage to a NumPy file to generate square images and resizing them. Then the images were stored in an array for the training dataset. For all the local images an iteration was applied to take all the sample dataset images from local storage to the NumPy array resizing them in the 64px*64px square size. With all the resized images in the array it needs to be reshaped accordingly with the image channels taken 3 for the process. Reshaping them it needs to be converted to a suitable datatype. For this case float32 was used. Finally, the color normalization was done by diving the training data by 127.5 and subtracting 1 from it and renamed the files to store.

As the generator network produces the fake images from this training set of images the above processed data is served to the generated network for generating fake faces. A function is defined to save the fake produced images with a preview margin of 64px * 64px with added noise. For this generator Gaussian noise was used. The produces image with added noise is previewed in a plot. A sample of the noisy image is shown below in figure 4.3.

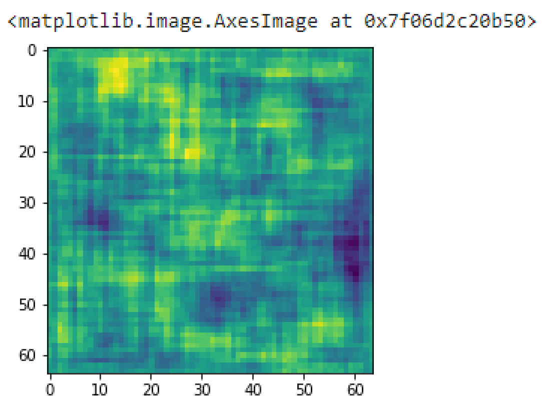


Figure 4.3: Image with added noises

4.4 Output and Summary

The goal of the thesis was to generate newer images of human faces that does not exist. To do that the above-mentioned data set is collected and processed to observe the visual output of the images. The initial dataset with approximately 8000 images were taken for training the dataset. Eventually the collected images were added to enrich the dataset later for better output. Some of the network generated output image samples are shown below in figure 4.4.

Figure 4.4: Network generated output image

Thus, the purpose of the thesis is achieved to generate newer human face with an optimized approach and new dataset.

Chapter 5

Discussion

5.1 Outcome

The goal of the thesis was to research on the Generative Adversarial Network (GAN) and generate new human faces that does not exist. Basically, GAN is a sub set of the Deep Convolutional Neural Network which follows an unsupervised learning approach. Two primary networks were implemented to achieve the output those are, generator and discriminator network. At first the discriminator was trained with real data then the generator generated fake images for the discriminator to distinguish between fake and real images. This is how the training data set of the discriminator improved. Now the discriminator is trained on a fake data and the generator is trained with the output of the discriminator. This is how both the networks gets trained to produce better fake images. The generator gets trained to produce better fake images and discriminator better distinguishes the real and fake images.

Now we could successfully achieve the goal to produce new faces using our dataset in a more optimized approach with a newer combined dataset. To begin with the training of the networks the dataset provided shown in the following Figure 5.1.

Figure 5.1: Example of input dataset

The output result that the network generated shown in Figure 5.2.

Figure 5.2: Example of output dataset

These output images are generated from the generator network of human faces that does not exist in the world based our training data set. Therefore, the results clearly show that the goal of the thesis is achieved with the designing, modelling and implementing for the thesis work.

5.2 Theoretical Comparison of the Result

With the theoretical concepts of GAN, the results were comparable with the variation of different parameters. To generate images the image square size, number of channels, epochs count, batch size and buffer size were already introduced. The number of epochs is a parameter that controls how many times the learning algorithm runs over the whole training dataset. Each sample in the training dataset has had the opportunity to update the internal model parameters once each epoch. Now increasing the epochs results in the discriminator to act steep down the curve of distinguishing between real and fake images. So, the accuracy increases to a certain threshold that the discriminator fails to identify fake images and the generates creates better fake images that dos not gets rejected by the discriminator.Here are few of the samples when the epoch count was 100, shown in Figure 5.3.

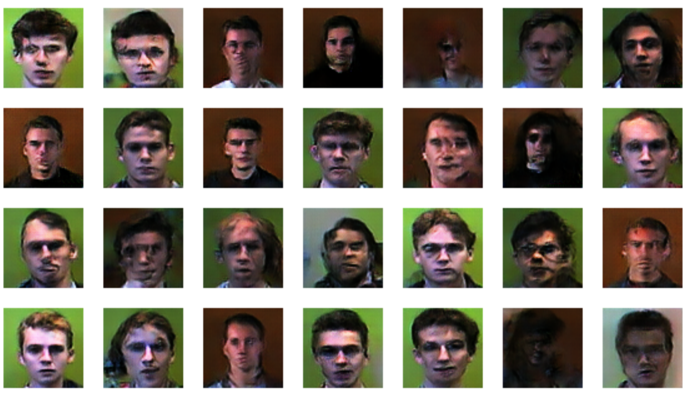


Figure 5.3: Samples of dataset when epoch is 100

When the epoch count was 100 the generated were still distorted to some extent and the loss functions of generator and discriminator were plateauing. As the epoch count were increased the improvement rate became more and more insignificant.

5.3 Limitations Doing Research

Generative Adversarial Network technology follows generative method for generating fake images. Thus, it deals with a large dataset of images to train the dataset. To prepare the dataset the real images samples need to be processed with defined parameters such as the batch size, buffer size, image channels, size etc. Again, to increase the score of success function and increase the accuracy the epochs were increased. Now to process the large set of images with the mentioned parameters and with high epoch a powerful graphics processing unit is required to prepare the dataset and train the networks to generate better images. A GPU with less processing power fails to process the images of the dataset faster and accurately. So, to process the dataset for training the networks a powerful graphics processing unit is required.

Again, to train the generator and discriminator networks the real image samples are required. We have used approximately 8500 real image samples for the training data set. It would have generated more accurate images if the dataset could be enriched with more samples. But there was a limited real image samples for the research which could be increased to increase the score of the success function.

As we have used Google Colab we could not store more than around 80k images in our training data set because Google Colab provides a limited amount to RAM. And There was also GPU Limitation which increases time per epoch and not enough epoch can be tested there.

5.4 Significance of the Research

With the help of Convolutional Neural Network (CNN), Generative Adversarial Network (GAN) is implemented. This newer technology is used in various applications where self-trained generative components are needed to produce. GAN is mostly used to generate data from scratch using its unsupervised learning approach to train the data set of the networks. Mostly it is used to generate the images. Here are few of the applications of GAN:

- Style Transfer
- Image to Image Translation
- Text-to-Image Translation

- Face Aging
- Cell Gene Imputation
- Data Augmentation

For this research work GAN technology is used to develop or generate new human faces that does not exist with the help of generative method. This research opens a new door to serve a new face for any intelligent bot or assistant. Any human like face but that does not exist in the world can be generated to serve to an artificial intelligent bot or assistant.
Again, it can be used in augmented reality to provide a human like face to any object. The applications can be further extended to holographic posters or banners as well. Therefore, it could be said that this research work to generate new human face has significant applications in different sectors where new face construction is needed. Beside this, the dataset can be changed to almost exclusively anything while maintaining the same model to produce the desired output.

Chapter 6

Conclusion

Generating new human faces with the help of a sub set of Convolutional Neural Network which is Generative Adversarial Network (GAN) might be one of the most applicable research to support augmented reality and other related fields. With this research, the successful face generation could be achieved, the limitations could be resolved to improve the output result and new information could be added for the future research work. The main findings from this research were face generation using two primary networks in an optimized approach.

Again, some variation in generated images could be observed when different parameters were varied such as epoch and sample image data. It was noticed that the generated faces were not much accurate when the epoch were less. Testing the network with 100 epoch it generated distorted faces with an increased failure function score. Taking 1000 epoch the network could generate more accurate faces and it was observed that after 700-750 epoch the network was unable to generate more accurate faces and the score of the success function remained almost constant.

One of the primary limitations of the research was to overcome the graphics processing unit issues. As this research was on the generative model to generate images of human faces so a large dataset of image needed to process and train the training dataset of the networks. For that a powerful GPU was necessary and we overcame it with the help of the online tool named Google Colab to implement and generate images. Again, the dataset initially had fewer images so the images were not much accurate and thus we introduced new dataset by combing new images with the previous dataset to generate much accurate images. Thus, we could overcome few limitations and challenges by adopting the mentioned measures.

Finally, the primary goal of the research was satisfied to generate fake images from the generator network by fooling the discriminator network with our dataset and the configurations in a more optimized approach. However, future improvements of the research could be achieved by introducing a variety of images dataset which might increase the accuracy of the output but keeping in mind that it should not make the discriminator training dataset more accurate or powerful initially which would result in not discriminating between real or fake images at the beginning. This could lead the generator to generate fake images but the discriminator unable to discriminate them. So, the end output might be inaccurate.

Bibliography

[Brock et al., 2016] Brock, A., Lim, T., Ritchie, J. M., and Weston, N. (2016). Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*.

[Dash et al., 2017] Dash, A., Gamboa, J. C. B., Ahmed, S., Liwicki, M., and Afzal, M. Z. (2017). Tac-gan-text conditioned auxiliary classifier generative adversarial network. *arXiv preprint arXiv:1703.06412*.

[Denton et al., 2015] Denton, E., Chintala, S., Szlam, A., and Fergus, R. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. *arXiv preprint arXiv:1506.05751*.

[Gong et al., 2017] Gong, M., Niu, X., Zhang, P., and Li, Z. (2017). Generative adversarial networks for change detection in multispectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2310–2314.

[Goodfellow et al., 2014] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.

[Guérin et al., 2017] Guérin, É., Digne, J., Galin, E., Peytavie, A., Wolf, C., Benes, B., and Martinez, B. (2017). Interactive example-based terrain authoring with conditional generative adversarial networks. *Acm Transactions on Graphics (TOG)*, 36(6):1–13.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Huang et al., 2017] Huang, R., Zhang, S., Li, T., and He, R. (2017). Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE international conference on computer vision*, pages 2439–2448.

[Isola et al., 2017] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

[Jeff Heaton, 2021] Jeff Heaton, j. (2021). t81$_5$58$_{d}$*eep*$_{l}$*earning*.

[Jin et al., 2017] Jin, Y., Zhang, J., Li, M., Tian, Y., Zhu, H., and Fang, Z. (2017). Towards the automatic anime characters creation with generative adversarial networks. *arXiv preprint arXiv:1708.05509*.

[kaggle, 2021] kaggle, k. (2021). faces$_{d}ata_{n}ew$.

[Karras et al., 2017] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

[Liu and Tuzel, 2016] Liu, M.-Y. and Tuzel, O. (2016). Coupled generative adversarial networks. *arXiv preprint arXiv:1606.07536*.

[Perarnau et al., 2016] Perarnau, G., Van De Weijer, J., Raducanu, B., and Álvarez, J. M. (2016). Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*.

[Radford et al., 2015] Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

[Reed et al., 2016a] Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B., and Lee, H. (2016a). Learning what and where to draw. *arXiv preprint arXiv:1610.02454*.

[Reed et al., 2016b] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016b). Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR.

[Taigman et al., 2016] Taigman, Y., Polyak, A., and Wolf, L. (2016). Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*.

[tensorflow, 2021] tensorflow, t. (2021). Deep convolutional generative adversarial network.

[Zhang et al., 2017] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915.

[Zhu et al., 2017] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Stay FOCUSED Drink COFFEE Move FORWORD