FACE SYNTHESIS USING GENERATIVE ADVERSARIAL NETWORK

A DISSERTATION

Submitted in the partial fulfillment of the
requirements for the award of the degree of
MASTER OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

By
SHIVALI SHARMA
(16535037)

UNDER THE GUIDANCE OF
DR. BALASUBRAMANIAN RAMAN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE
ROORKEE- 247667, INDIA
MAY, 2018

AUTHOR'S DECLARATION

I declare that the work presented in this dissertation with title **"Face synthesis using Generative Adversarial Network"** towards fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science & Engineering** submitted in the **Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period of **May 2017 to May 2018** under the supervision of **Dr. Balasubramanian Raman**, Associate Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, India. The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date: .....
Place: ROORKEE

SHIVALI SHARMA
(16535037)
M.TECH (CSE)

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my Knowledge and belief.

Date: .....
Place: .....

Sign: .....
DR. BALASUBRAMANIAN RAMAN
(Associate Professor)
Indian Institute of Technology Roorkee

ACKNOWLEDGEMENTS

Dedicated to my family and friends, for standing by me through thick and thin, without whom i would not have gotten this far. I would like to express my sincere gratitude to my advisor **Dr. Balasubramanian Raman** for the continuous support of my study and research, for his patience, motivation, enthusiasm and immense knowledge. His guidance helped me in all time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my study. I would like to express my sincere appreciation and gratitude towards **Mr. Himanshu Buckchash** for his encouragement, consistent support and invaluable suggestions at the time I needed the most.

I am also grateful to the Department of Computer Science and Engineering, IIT Roorkee for providing valuable resources to aid my research.

SHIVALI SHARMA

ABSTRACT

Generative Adversarial Networks that fall under the class of generative models, aim at taking training examples from the training set and learning the probability distribution that generates those samples. The network is adversarial in the sense that the discriminator tries to maximize the probability of identifying the real data whereas the generator tries to fool the discriminator by producing synthetic data as close as possible to the ground truth value. In recent years, many powerful models using neural network architectures have been introduced that try to learn the discriminative features of the text representations. Also, GANs have been extremely successful in generating realistic images belonging to various categories. Influenced by the success of GANs, researchers thought of applying the GAN model in the human face synthesis task. There exist several attempts for the face synthesis task that try to generate real human faces from the form of input given to them . Unlike the already existing attempts to create human faces, our model tries to apply the concept of text-to-image synthesis [1] GAN in the generation of human faces from the text description stating the attributes o of their respective faces provided as the input. The training of Generator is assisted by the adversary Discriminator (Matching-aware Discriminator model(CNN)) that differentiates the results given by the Generator and the ground truth values. The Generator model would thus learn to generate the human faces that are similar to the ground truth values and thus try to cheat the adversary. The aim is to produce strong results and see the behavior of GAN model in the human face generation task.

Contents

List of Tables

List of Figures

Chapter 1 INTRODUCTION

The generative models that aim at generating observable values of data given some latent parameters find out the joint probability distribution over the data values and labels. There exist several models for generating the data values by learning the data distribution. Variational Autoencoders [4] work as probabilistic graphical models where lower bound on log likelihood of data has to be maximized. The generated samples tend to be a little blurry hence some other methods were needed. There also exist autoregressive models like PixelRNN [5] where the Recurrent Neural Network provide a shared, compact way of parameterizing a series of conditional distributions. These models also prove to be inefficient during sampling and could not provide low dimensional codes for some images. Generative Adversarial Networks (GANs), falling under the class of generative models, try to learn an estimate representation of training samples drawn from some data distribution [6]. GANs have a wide range of applications and prove to produce better samples than other methods that provide compelling reasons into investing time and resources for their detailed study.

$$\min_{G}\max_{D} V(G,D) = E_{x\sim p_{data}(x)}[log(D(x)] + E_{z\sim p_z(z)}[log(1-D(G(z))] \tag{1.1}$$

Generative Adversarial Networks (GANs), falling under the class of generative models, try to learn an estimate representation of training samples drawn from some data distribution.The basic idea behind GANs is an adversary game going on between two neural networks namely ***discriminator*** and the ***generator***. The generator creates data samples coming from same data distribution as the training data. The discriminator on the other hand performs the examination of these samples to decide whether they are real or fake. The generator and discriminator are represented as functions that are differentiable both respect to the input as well as their parameters. The generator and discriminator are represented as functions that are differentiable both respect to the input as well as their parameters. The **discriminator** is a function taking $\boldsymbol{x}$ as input and having $\theta^{(D)}$ as its parameter. Similarly, the **generator** is a function G taking z as its input and having $\theta^{(G)}$ as its parameter. The cost functions of both the players in the game are represented in terms of parameters of both. The aim of discriminator is to control its parameter $\theta^{(D)}$ and the generator aims to control the parameter $\theta^{(G)}$. Let the value function be V(G,D). Then the training of G and D is done simultaneously with the aim of adjusting parameters of G

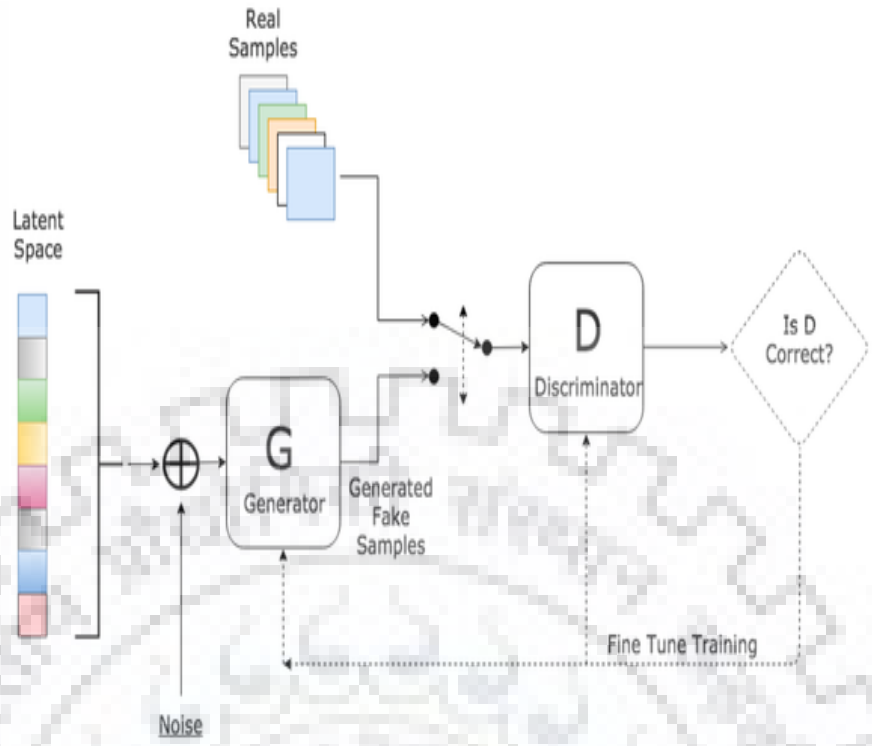

Figure 1.1: *GAN basic working*[1]

to minimize the value log(1-D(G(z)) and the parameters of D to minimize log(D(x)). The basic working equation of the GAN is represented by equation (1.1). The structure of generator and discriminator network could be easily understood with the help of Figure 1.1[1].

The GAN models can be supervised or unsupervised. The supervised learning algorithms try to associate some input with the corresponding output given a training set of examples. Every example is a pair that has an input object and an associated output known as the supervisory signal. Based on the analysis of this training data, some inferred function is obtained which is either a regression function in case of continuous output or a classifier in case of discrete output. Some examples of supervised learning GAN models are Conditional GAN [7], Text GAN [8], DCGAN [9] etc. GAN models can also fall under the category of unsupervised learning. The unsupervised learning algorithms draw samples from some distribution and learn some information from that. They extract features from the given distribution and have no supervision signal to guide the learning process. The aim here is to have a representation that reflects the statistical structure of the input patterns. Examples of GAN with unsupervised learning include Video GAN [10] and Multi GAN [11].

[1] http://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html

Since the cost functions are non-convex, with continuous parameters and the parameter-space highly dimensional, finding the Nash equilibrium [6] for the purpose of GAN training is a very difficult-to-do task. There exist several papers focusing on improving the training of GANs [12] [9] [13]. The various approaches proposed for the training process of GAN have been explained by Salimans *et al* [14].

The automatic face generation task is an extremely useful task that finds its applications in various fields. Law enforcement field can get various benefits from a system that can synthesize images of the criminals automatically from the text description of their visual features given as input to them. In case of any such crime investigation, the witness plays a vital role in giving the description of the criminal to the cops who then make a sketch of the criminal with the help of any artist. But this task can be automated with the help of computers by merely giving the textual description of criminal's visual attributes as input to the trained model that can then give as output the image of the criminal. If an efficient model can be built, a very huge advancement can be done in this process of law. Also the entertainment field can get benefits from such model.

To generate the images of human faces, many works have been done till now. In their work [15] have developed a model that can transform a human face to its sketch and a sketch to its corresponding human face. The work by [16] uses a model that generates human faces from the given landmarks on the respective human face. They try to generate the faces of humans from the information preserved in the landmarks like the gender, pose and the visual attributes describing the face structure. Our speculation is that similar to the model used in [1] for generative adversarial text to image synthesis for the birds and flowers dataset, face generation can also be done if the visual attributes describing the face structure are given as input in a text file and the model generates the corresponding human face as output. Discriminator can be made efficient to make the model capable of producing clear human faces. Inspired by the work in [1], our proposed model has a Matching-aware Discriminator that takes three pairs namely (real text, right target face), (real text, fake target face) and (wrong text, right target face). The model would thus aim at better training of the generator to produce translations that are more similar to the ground truth values.

Several approaches exist in the literature to attain the visual attributes of the human face from its corresponding image. But the reverse problem is relatively unexplored. Many works exist that attempt to generate photo-realistic images from the text description given as the input

to them. In their work [17] have used a CNN based generative model that is Variational Auto Encoder for the generation of images from the text description. The work done by [1] tries to generate photo-realistic images of flowers and birds from the textual description of their features given as text file to them. Their model uses a matching aware discriminator that aims at minimizing the model error. The work by [3] used a model for generating realistic images from text descriptions by using a model that has two layers stacked one upon the other. One stage generates low resolution images that take into consideration the basic details of the image and the second stage generate more realistic and high resolution images by taking into account all the vivid object details and intricate detailing of the image. Inspired by the work of the text-to-image synthesis GAN done by Reed *et al* [1], we propose a method of image generation from the textual visual description using GAN model. Our model tries to produce realistic images by using the matching-aware discriminator model that pairs the correct text description with its correct human face and eliminates the chances of fake source vector combined with the real target image. The model aims to increase the range of input pairs and thus improving the training accuracy of the Generator.

The deep learning face generation model suggested by us has an ability to generate the target face given the visual description as input. This problem of generating faces from the textual description can be roughly divided in two parts namely language representation and synthesis of image. A challenging task is that the generation of images from text comprises an extremely multi-modal distribution which means that there could be multiple possible faces corresponding to a given text description of the visual attributes. So, from all the plausible configuration of the image pixels, the most accurate one has to be chosen that best describe the person's visual attributes. To tackle vanishing and exploding gradient problem Leaky-ReLU layers have been introduced in both the generator and the discriminator.

Chapter 2 LITERATURE SURVEY

2.1 *Generative Adversarial Networks(GANs)*

In their paper [6] has introduced the basic GAN network that estimate the generative models via an adversarial training process between the two models namely Discriminator and the Generator. In their paper [8] has worked on text generation using the GAN model. In their work [10] have modeled the scene dynamics for video generation and the video recognition task for predicting the plausible futures of static images. In their work [1] introduced a GAN network for the synthesis of images conditioned on text embeddings coming from the true data distribution in case of real image and from the interpolations of the real data in case of fake image. To improve the quality of images generated corresponding to the input, [3] and [2] also introduced their models that take into consideration the vivid object details and generate high resolution images. In their work [11] have introduced multi-agent message passing GAN where both the generators have the objective of improving their performance with respect to each other and with respect to the common discriminator model. DCGAN for unsupervised learning [9] has some changes as compared to the basic DCGAN model. Firstly, the pooling layers are replaced by the strided convolution layers in the generator that allows it to learn its own summary. Secondly, fully connected layers are removed. This ensures that the feature maps calculated for every classification category are easily passed back to the previous convolution layers for interpretation of category information unlike the fully connected layer in standard DCGAN that acts as a black-box hiding this information. Thirdly, batch normalization is used that gives several advantages like higher learning rates and a stable model.

2.2 *Matching Aware Discriminator*

In their work [1] presented a discriminator model that outperforms the standard discriminator model. The proposed model takes as input three pairs namely (real text , real target image), (real text, fake target image) and (wrong text, real target image) and try to distinguish the real pair of image from the fake ones. The model outperforms the standard GAN setting for image generation by taking into account all the errors that could occur while pairing the text caption with its corresponding image.

2.3 *Text-to-image synthesis using GAN*

In their work Reed *et al* [1] have used a generative adversarial model to generate photo-realistic images of flowers and birds from the textual description of their features given as input. So, they basically do the character to pixel values translation using the deep learning model. The work done by [3] use a stacked model of GAN for generating photo realistic images that involve vivid object details and all intricate information about the image. They use the concept of refining the image sketch where the first stage of the model tries to incorporate the basic image details followed by the second stage that takes into consideration the vivid object details for generating a high resolution counterpart of the low resolution output from the first stage. The work done by [2] prove to be the state-of-art in the image generation process as it includes the class information along with the textual description to take into account the structural coherence and to generate diverse range of outputs.

2.4 *Face synthesis*

Many works exist to generate human faces using GAN model. No work till now has attempted to generate human faces from the text description of the facial features given to the GAN model having a matching aware discriminator as done by our model. In their work [16] presented a model to generate images of humans form their respective landmark key-points. In their work [18] have generated faces from the respective sketch and visual attributes of the person. They have used a combination of CVAE and GAN model to generate the images. In their work [19] have presented an evolutionary computing model to generate the sketch of a person using the text descriptions of their visual attributes. The description is taken using a GUI which is then transformed to the sketch of the desired person and iterated until the user is satisfied with the sketch. The work done by [20] introduces a model that extracts the features of the face organs like nose, eyes, forehead, hair, eyebrows, ears and other facial boundary features and using these extracted features it learns to generate the face drawings.

Chapter 3 PROBLEM FORMULATION AND PRELIMINARIES

3.1 *Problem Definition*

To generate plausible images from the detailed text descriptions, a model [1] with two features namely learning text feature representation capturing vital visual details and secondly using these features to create compelling images that might be mistaken by human for real was proposed by Reed *et al.* Both generator and discriminator are conditioned on side information i.e the text description. The model is similar to conditional GAN except that it is conditioned on text rather than class labels. It is known to be the first end to end differentiable model for directly converting the text into the pixels. The generator samples from noise prior z and uses the text encoder for encoding the text query t. The output $\phi(t)$ is passed on through fully connected layer, then leaky ReLu and then concatenated to noise vector. After this, the concatenated vector is fed forward to the deconvolution Generator and a synthetic image $\bar{x}$ is generated. Discriminator aka **Matching Aware Discriminator** tries to differentiate three types of embeddings of pairs namely *(real image, right text)*, *(real image, wrong text)* and *(fake image, right text)* [1]. The model gives good result but the images generated are not of very high resolution and lack some details and vivid object parts.

Face synthesis aims at generating realistic faces of human from the given textual description of the facial attributes of the respective person. The approaches available till now have never tried a Generative Adversarial Network for synthesizing the face images. With a properly designed model for this work, a great help could be provided to the law enforcement sector as the criminal faces which till now are generated manually with the help of sketch artist could then be prepared automatically from the machine by only providing the face details to the system in the form of a text file. Because of the huge success of the GAN models in the text-to-image synthesis task, the face synthesis task can also be combined with the adversarial framework in order to ensure that the faces produced by the model are as close as possible to the ground truth value. The face synthesis task using GAN network can be divided into three parts- training of the generator by error signals propagated from the discriminator which ensures that the generator produces face images that are as close to the ground truth value as possible, training of the discriminator that aims at increasing the probability of differentiating the fake images from the real ones by back propagating error signals and the final generation of faces by separating the generator model

from the discriminator after doing the proper training.

3.2 *Preliminaries*

3.2.1 *Generative Adversarial Networks(GANs)*

GANs fall under the category of generative models that use the adversarial training process to produce realistic outputs. The two networks namely Generator and the Discriminator are involved in a min-max game with each one of them trying to be better than the other one. The Generator G goes through the optimization process to re-generate the true data distribution p_{data}. The generator creates data samples coming from same data distribution as the training data. The discriminator on the other hand performs the examination of these samples to decide whether they are real or fake. The generator and discriminator are represented as functions that are differentiable both respect to the input as well as their parameters. The **discriminator** is a function taking $\boldsymbol{x}$ as input and having $\theta^{(D)}$ as its parameter. Similarly, the **generator** is a function G taking z as its input and having $\theta^{(G)}$ as its parameter. The cost functions of both the players in the game are represented in terms of parameters of both. The aim of discriminator is to control its parameter $\theta^{(D)}$ and the generator aims to control the parameter $\theta^{(G)}$. Let the value function be V(G,D). Then the training of G and D is done simultaneously with the aim of adjusting parameters of G to minimize the value log(1-D(G(z)) and the parameters of D to minimize log(D(x)). It does so by generating images which are a bit difficult for the Discriminator to differentiate from the real ones. Meanwhile, The Discriminator D is also processed and optimized to be able to distinguish the real images from the synthetic ones. Both the neural networks compete against one another to make their performance better. The objective function of the GAN is described by the equation (3.1).

$$\min_{G}\max_{D} V(G,D) = E_{x\sim p_{data}(x)}[log(D(x)] + E_{z\sim p_z(z)}[log(1-D(G(z))] \tag{3.1}$$

In this equation, x represents the real image which has been obtained from the real data distribution p_{data}, z represents the noise that has been sampled from some Gaussian or random distribution p_z.

3.2.2 Symmetric and deep joint embedding for text

The text input given to the GAN model is converted into the vector representation which is then fed forward in the network. The approach used by Reed *et al.* [21] has been used for the synthesis of vector from the given text description. The model which is the skip-thought model falls under the category of encoder-decoder models. The encoder and decoder used are basically GRU(variant of LSTM) models which will be described in detail in the next section.

3.2.3 LSTM(Long Short Term Memory)

The LSTM model are special type of RNN(Reccurant Neural Network) that overcome the memory problem with RNN i.e they can learn long range dependencies over the text. They have a chain of repeating modules that have 4 layers of neural network layers each.

In the Figure3.1[1], the line at the top of the diagram represents the cell state which is the key to the network. Information can be added or removed from the cells with the help of special units called gates. Gates can pass the new information or ignore the information depending upon the value of the sigmoid network which varies in the range of [0,1]. The first layer is the forget gate layer that makes the decision of what information has to be discarded. It is a sigmoid layer that outputs values between 0 and 1. 0 signifies completely throw away the information and 1 signifies completely store the information. There might be situation s where the model wants to forget and discard the previous information and save the new one for achieving the results. In that case, forget gate comes into play. It is represented by the equation 3.2.

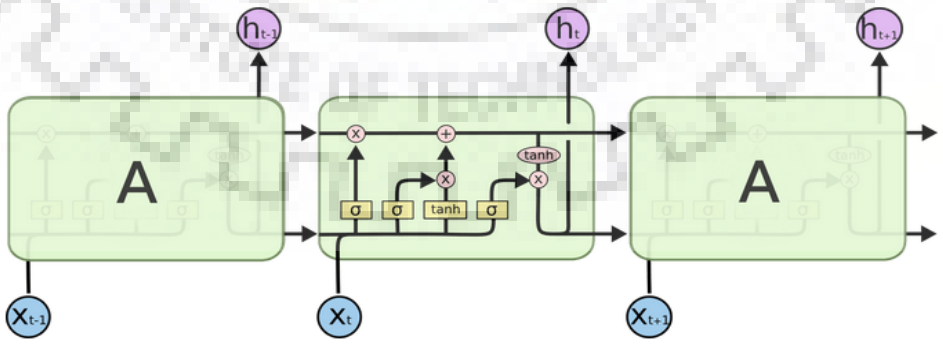

Figure 3.1: *Four interactive layers in the repeating module of LSTM*[1]

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{3.2}$$

[1] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

The information can also be required to be stored for further use in the network. In that case, input gate layer and tanh layer comes into player. Firstly, the input layer which is again a sigmoid layer decides what information to be updated. Then the tanh layer produces vector of the available candidate values to be used in the addition to the state. After this, the two outputs i.e one from the input gate and the other from the tanh layer are combined with one another. This combination gives the model capability to add new data to the layers and forget the previous one(using forget gate). The equation 3.3 represents the working of input gate and the equation 3.4 represents the working of the tanh gate. Combining these two gates is then carried out the update of the old cell state C_{t-1} into new cell state C_t. The update equation

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{3.3}$$

$$\tilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C) \tag{3.4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{3.5}$$

An important decision is to finalize what information to be sent as output which is done using the output gate. This gate uses a combination of sigmoid gate and a tanh gate. Here, sigmoid gate decides what portion of the cell state to be sent in the output. Meanwhile, the tanh gate takes the cell state and outputs values in the range[-1,1] which are then multiplied with the sigmoid layer output to finally decide the part of cell state to be sent in the output of the model. The equation 3.7 represents the final working at the output end of a LSTM module.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{3.6}$$

$$h_t = o_t \odot tanh(C_t) \tag{3.7}$$

3.2.4 *Gated Recurrent Unit(GRU)*

The LSTM networks suffer from the problem of vanishing gradient i.e in the back propagation, when the gradient of error with respect to the weights is fed backwards, the value of these gradients tend to become very small for the starting layers and hence the training process of neurons in these layers become very slow. This is not a good indication as these front layers serve as the building blocks of entire network and help learning the basic features and patterns

of input data. This problem of vanishing gradient can result in slow training process as well as low prediction accuracy. GRU based RNN tries to capture the long term dependencies in the sequential data and also overcome the vanishing gradient problem that is present in the LSTM networks, using two gates namely reset gate and update gate.

In the Figure 3.2[2], line at the top is representing the cell state. The update gate is the combination of input and forget gates of the LSTM network.The update gate ensures how much of the previous memory has to be kept while the reset gate combines the new input with the previous memory. The internal structure of GRU is simpler than that of LSTM and hence their training is easier and faster as less number of modifications are needed to be made in their internal states. The following equations summarize the working of GRU:

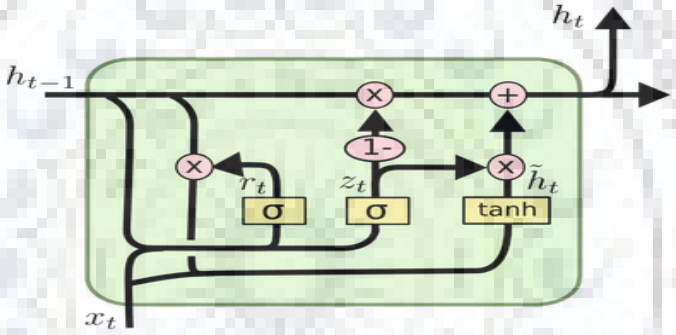

Figure 3.2: *Repeating module of GRU*[2]

$$z_t = \sigma(W_z.[h_{t-1}, x_t]) \tag{3.8}$$

$$r_t = \sigma(W_r.[h_{t-1}, x_t]) \tag{3.9}$$

$$\tilde{h_t} = tanh(W.[r_t \odot h_{t-1}, x_t]) \tag{3.10}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h_t} \tag{3.11}$$

The Equation 3.8 represents the working of update gate which is combination of input and forget gates of LSTM. Here the update gate multiplies the input x_t and the previous hidden state h_{t-1} with their respective weights. The Equation 3.9 represents the working of the reset gate which determines how much of the previous information to be preserved and thus how much to be discarded from passing to the next higher levels of the network.

[2] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

3.2.5 *Convolution Neural Network(CNN)*

Convolution Neural Networks are special type of networks that can process data with a known grid like topology. They are feed forward neural network with certain variation i.e addition of multiple convolution and pooling layers They are the first of their kind as a working deep neural networks trained with back propagation. They have the following three stages:

- Convolution Stage: In this stage,a set of linear activations are performed by performing set of parallel convolutions.
- Detector Stage: A non-linear activation function evaluates each of the linear activation produced in the first stage.
- Pooling Stage: This layer replaces the output of the network with the nearby outputs summary statistics. This helps make the representation invariant to small translations in the input data.

The Figure 3.3[3]represents the feature extraction layers which are in a repeating mode. The three level of layers in CNN are: input layers that accept the 3 dimensional data, feature extraction layers that comprise of 3 parts namely: Convolution layer, Detector layer and Pooling layer and classification layer that produce the class scores by passing higher order features through one or more fully connected layers. The feature extraction layer works by successively finding higher order features for the input.

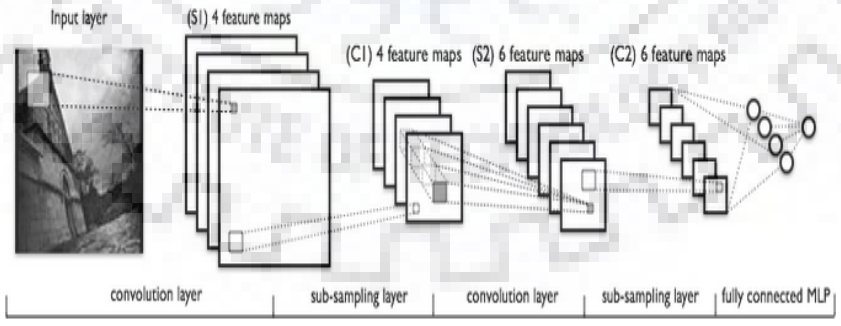

Figure 3.3: *The basic architecture of CNN*[3]

[3]https://deeplearning4j.org/convolutionalnetwork

3.2.6 *Matching Aware Discriminator*

The matching aware discriminator is basically a CNN that tries to maximize the probability of differentiating the fake translations from the real ones. It takes as input three pairs namely: (real input, real output), (real input, fake output), (wrong input, real output). By training itself to differentiate these pairs, the discriminator tries to increase the probability of rejecting the fake pairs and accepting the real ones.

3.2.7 *Minibatch discrimination*

GANs sometimes collapse to a value set of parameters where a single mode is imminent and the gradient of the discriminator points in the same direction for many points. The solution to this is to make the discriminator look at batch of data points before taking any decision. Here, for every input x_i, a vector of features f(x) is calculated by an intermediate discriminator layer. This f(x) is multiplied by a tensor that results in a matrix M_i. The L_1 distance is then calculated between the rows of the matrix for all samples and a negative exponent is applied to obtain c_b. This way the output o(x_i)for this particular minibatch layer of sample x_i is found by adding up all the $c_b(x_i,x_j)$ values with all other samples x_j as given in the equation(3.12). The next layer gets the concatenated value of the input to this layer and o(x_i). Minibatch features are calculated for both the samples from the data and the generator. This information from minibatch is used as side information by the discriminator to make decision.

$$c_b(x_i,x_j) = exp(-||M_{i,b},M_{j,b}||_{L_1}) \tag{3.12}$$

Chapter 4 PROPOSED APPROACH

We have used the model proposed by Reed *et al* [1] for the text-to-image synthesis task in order to generate face images given the textual attributes describing the visual features of the respective person. The face synthesis model using Generative Adversarial Network(GAN) consists of two basic components: Generator and the Matching aware discriminator. After the training is done, the generator module is separated from the discriminator and the text file describing the visual attributes of the person are given as input to it and it produces the target human face. The training phase requires the text file describing the visual attributes of the person and the corresponding ground truth values of the face images. A combination of texture and color attributes are given as input in the text file which is then converted to its skip-thought vector. The diagrammatic representation of the model is shown in fig 4.1. The text file that describes the visual attributes of the person are given as input to the generator that produces its corresponding text vector using the encoder model of skip-thought vector. The generator then combines this text vector and noise and then pass this concatenated vector to the deconvolution network of generator which produces synthetic image from the visual features. This synthetic face image is then fed to the discriminator which also receives the fake image(mismatching image with the text) and the wrong text input(mismatching text file). This matching aware discriminator receives the three pairs i.e (real text , real target image), (real text, wrong target image) and (fake text, real target image) and thus try to minimize the cross entropy across all these pairs. The aim of the

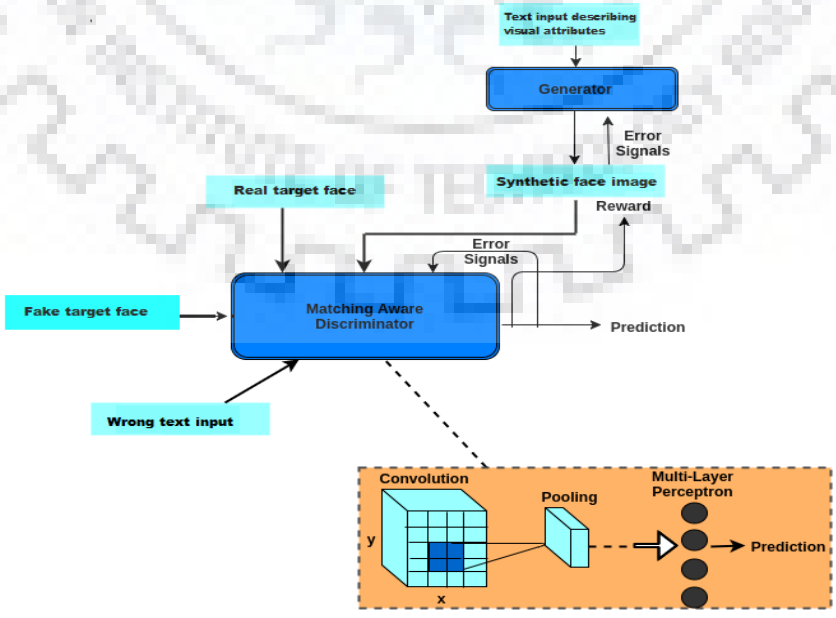

Figure 4.1: *Architecture of proposed adversarial face synthesis model*

discriminator is to increase its efficiency and thus distinguish real images from the fake ones with highest probability possible. The goal of the GAN model is to make the synthetic face image generated by the generator as close as possible to the ground truth value. The generator and the discriminator play an adversary game against each other where the generator aims at producing samples that get accepted by the discriminator and the discriminator trains itself and tries to make itself better with the error signals. Aim is to minimize the mean cross entropy across the prediction from the discriminator.

4.1 *Generator*

The generator is a deconvolution model in which noise sampling is first done through the noise prior and then the text description is encoded using the GRU and this GRU based text encoder then gives as output the skip-thought vector corresponding to the input text file. The following equations describe the encoding process of the text file to its corresponding vector:

$$z_t = \sigma(W_z.[h_{t-1}, x_t]) \tag{4.1}$$

$$r_t = \sigma(W_r.[h_{t-1}, x_t]) \tag{4.2}$$

$$\tilde{h_t} = tanh(W.[r_t \odot h_{t-1}, x_t]) \tag{4.3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h_t} \tag{4.4}$$

In the above equations, r_t represent the reset gate, z_t represent the update gate, $\tilde{h_t}$ represent the current cell state and h_t represent the updated cell state. The value of h_t in the N^{th} iteration represent the final vector corresponding to the input sentence. A smaller dimension compressed version of the text embedding vector is concatenated to the noise and then passed through the deconvolution network of generator. The generator does the work of expanding the visual features of the face to corresponding high resolution face image and the discriminator on the other hand extracts features from the face image using convolution process. So, we can say that the generator is a kind of convolution network that uses transposed convolutions to generate the synthetic images from the text input and noise samples. The discriminator on the other hand helps in finding out what portions of the synthetic image to alter in order to generate images that are close to the ground truth value. The discriminator has the task of differentiating the real images from the fake ones and it does so by updating itself from the gradients of the error that are back

propagated from the network.

4.2 *Matching Aware Discriminator*

The matching aware discriminator takes the hypothesis from the generator as input and tries to maximize the probability of differentiating the real translations from the fake ones. During training, the aim of generator is to maximize the chances of discriminator accepting the translations done by the generator and the aim of the discriminator is to reject as many fake translations as possible. With the help of the error signals from the discriminator, both the generator and the discriminator are trained to make themselves as efficient as possible. Thus there is a min-max game going between the two.

Chapter 5 EXPERIMENTATION AND RESULTS

5.1 *Architecture used*

We have performed the experiments for the text-to-image synthesis task using the model proposed by Reed *et al* [1] for the synthesis of face images from the text description of their visual attributes including both the texture and the color attributes. The diagrammatic representation of the model is shown in the figure 5.1. In the generator which is a deconvolution neural network, the noise

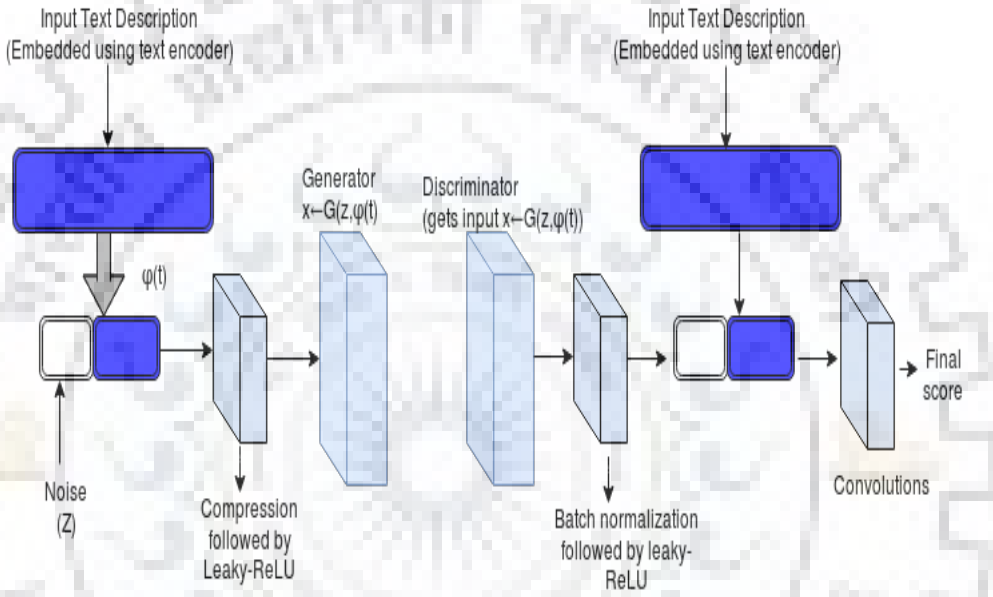

Figure 5.1: *Architecture of text-to-image synthesis model for face synthesis task*

and text combines to generate the images at the output end. The text is encoded using the text encoder(GRU) and the resultant text embedding is compressed followed by activation by Leaky-ReLU and then the noise sampled from the noise prior is concatenated to this text embedding. This concatenated vector is expanded to the size of the desired image and then several filters are applied on it to produce the desired image at the output end. This means, after the concatenated vector generation, normal deconvolution is applied on the text in the generator(CNN) that produces the synthetic image. Hence, the task of the generator is to generate photo-realistic face images by expanding the visual features. That is the reason why deconvolution network is used in the generator network.

The discriminator is a convolution neural network that performs the reverse steps followed in the generator. Firstly, the image is convolved to reduce its size and then concatenated with the reduced text embedding. This vector is compared with the vector corresponding to the real text

and real image pair and hence it is tried to reduce the mean cross entropy across the predictions made.

5.2 *Experiments*

In this section, the performance of proposed adversarial model for face synthesis is evaluated on a publically available Libor Spacek's facial images dataset. Specifically, we address the question as to how effective the Generative Adversarial Network using matching-aware discriminator [1] is for the task of face synthesis given the visual attributes in textual form as input to it.

5.3 *Experimental Setup*

5.3.1 Dataset

We experimented with the Libor Spacek's facial images dataset[1] from which we used 1000 images of 200 different persons. The text files describing the visual attributes of these 200 different persons having face images in different poses and expressions are generated by us. There are 5 captions in the text file generated per image in the dataset. So, a total of 1000 images of 200 persons(taken from publically available Libor Spacek's facial images dataset) and 1000 text files describing the visual features(including both texture and color features) of these persons where each text file includes 5 captions per face image have been used for the training process of the GAN based face synthesis model. 100 images from this Libor Spacek's facial images dataset are used by us for the testing purpose.

5.3.2 Hyper-parameters used

We experimented with the values of hyper-parameters for our Generative Adversarial Network model of face synthesis as summarized in the Table 5.1. These values are used for the evaluation of proposed model for face synthesis task.

5.4 *Evaluation measures*

For evaluation of the performance and quality of generated faces by the given adversarial model, we used the inception score [22] that finds the effectiveness of the model. Inception score is

[1] `http://cmp.felk.cvut.cz/~spacelib/faces/`

Flags	Value Used	Description
z_{dim}	100	Number of dimensions that the noise vector has.
t_{dim}	256	Number of Dimensions for the text embedding's representation in latent form
batch_size	64	Size of batch during mini-batch discrimination
image_size	64	Image size used in training
gf_dim	64	Count of convolution filters in the first layer of generator
df_dim	64	Count of convolution filters in the first layer of discriminator
caption_vector_length	2400	Length of the vector generated using skip-thought vector model
n_classes	200	Count of discrete classes used
learning_rate	0.0002	Rate used for learning process
β_1	0.5	Momentum for Adam optimizer

Table 5.1: *Hyper-parameters used by proposed model for the evaluation of GAN based face synthesis.*

based on the fact that high quality samples i.e the ones which are as close as possible to the ground truth values are expected to yield high prediction confidence i.e p(y|x) means given the input, the confidence of producing the correct output is high and highly varied.
The inception score is represented by the following formula:

$$I = exp(E_{x \sim p_g}(KL(p(y|x)||p(y))) \tag{5.1}$$

In the above equation, $x \sim p_g$ represents that the image x has been samples from the data distribution p_g. KL represents the Kullback-Leibler divergence between the two data distributions p(y|x) and p(y), p(y|x) represents the class probability distribution of the generated samples and p(y) represents the class probability distribution of the original samples. The Kullback-Leibler divergence basically calculates how one probability distribution is diverging from the other expected probability distribution. This is found by using the cross entropy across the two values. A KL divergence of 0 symbolize that the two probability distributions are similar to each other while a value of 1 symbolize great diversity in the behavior of the two distributions. The calculation of KL divergence is represented by the Equation(5.2) The main objectives of using inception score [23] are stated below with respect to the result expected from a generative model:

- The generated images should consist of clear images and objects whose vivid details are clear and they form sharp boundaries instead of being blur. So, p(y|x) needs to have low entropy to ensure that there is a single clear object in the generated image.

- The images generated should be diverse in the sense that they cover all the classes available and generate distinct and diverse images. This means a high entropy is expected for p(y).

$$D_{KL}(P||Q) = \sum_{i} P(i) log P(i)/Q(i) \tag{5.2}$$

5.5 *Compared models and methods*

For the evaluation of our model in the synthesis of face images from the given text description of the visual attributes of the respective person, we have used three models that are the state-of-art models for the text-to-image synthesis using GAN. The compared models does the text-to-image conversion task using bird and flowers dataset and not the facial images dataset. The comparison is done based on the quality(sharpness and diversity) of generated images by all the models which is determined using the inception score. These models are described as follows:

- **Generative Adversarial Text-to-image synthesis GAN**: To generate plausible images from the detailed text descriptions, a model [1] with two features namely learning text feature representation capturing vital visual details and secondly using these features to create compelling images that might be mistaken by human for real is used. Both generator and discriminator are conditioned on side information i.e the text description. The model is similar to conditional GAN except that it is conditioned on text rather than class labels. It is known to be the first end to end differentiable model for directly converting the text into the pixels. The model gives good result but the images generated are not of very high resolution and lack some details and vivid object parts. They have done their experiments on two datasets namely: CUB dataset of bird images and Oxford-102 dataset of flower images.

- **Stack GAN**: To generate photo-realistic images capturing the vivid object details, Stack GAN [3] came into play that has one layer stacked upon the other. Based on the text description given to it, the Stage-I draws the basic structure and basic colors of the image with raw shapes and then the output of generator of this stage along with the text description is fed to the Stage-II that generates the photo-realistic counterpart of the previously generated image with the vivid object details included. They performed their experiments on three datasets namely: CUB dataset of bird images, Oxford-102 dataset of flower images and MS-COCO dataset containing multiple backgrounds.

- **TACGAN**: Based on the idea of Auxiliary Classifier GAN [24], the Text Conditioned Auxiliary Classifier GAN [2] was introduced that conditions the generated image on text rather than class labels. Similar to [1], this method can also generate promising images that disentangle the style and content of the images. They have performed their experiments on the Oxford-102 dataset of flower images and prove to be the best model for the text-to-image synthesis with the highest inception score achieved.

5.6 *Results*

5.6.1 Qualitative results

By performing our experiments on the facial images dataset, we found out that for **epochs=150**, our model was giving the best results. Experiments with different epochs have been conducted and results obtained are compared. The results shown in the Figure 5.2 show the results for epochs equal to 150.

We will now show all the results obtained for the different values of epochs in the training process of Generative Adversarial Network used for face synthesis.

The figures 5.3, 5.4, 5.5, 5.6, 5.7 and 5.8 show the qualitative results obtained when the text input are given to the generator of the GAN model of face synthesis at different epoch values.

5.6.2 Quantitative results

As there is an adversary game going on between the generator and the discriminator of the used GAN model, both of them try to compete against each other to make their respective performance better than the other one. The Figure5.9 and Figure 5.10 represent the g_loss and the d_loss respectively of the used GAN model during validation. For evaluation of the performance and quality of generated faces by the given adversarial model, we used the inception score [22] that finds the effectiveness of the model. Inception score is based on the fact that high quality samples i.e the ones which are as close as possible to the ground truth values are expected to yield high prediction confidence i.e p(y|x) means given the input, the confidence of producing the correct output is high and highly varied. The Table 5.2 shows the inception score corresponding to the various comparison models used to measure the quality of generated images by our model doing face synthesis task. Though no other model till now has tried to generate faces from the text

Figure 5.2: *Results obtained for epochs=150*

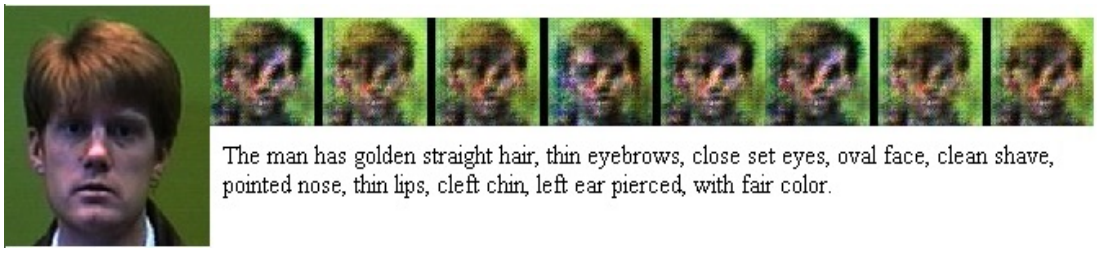

Figure 5.3: *Face sample obtained for epochs=50*

Figure 5.4: *Face sample obtained for epochs=80*

Figure 5.5: *Face sample obtained for epochs=140*

Figure 5.6: *Face sample obtained for epochs=160*

Figure 5.7: *Face sample obtained for epochs=200*

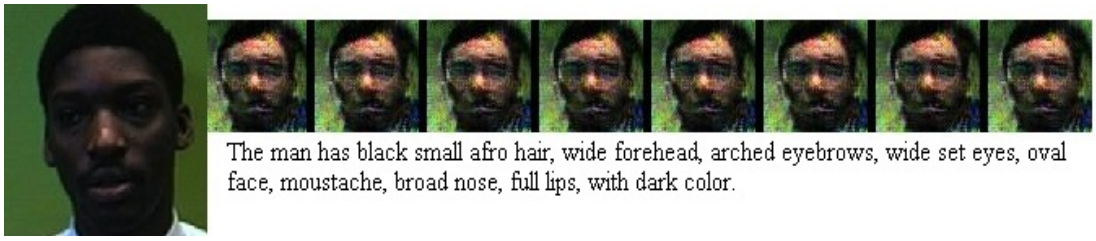

Figure 5.8: *Face sample obtained for epochs=250*

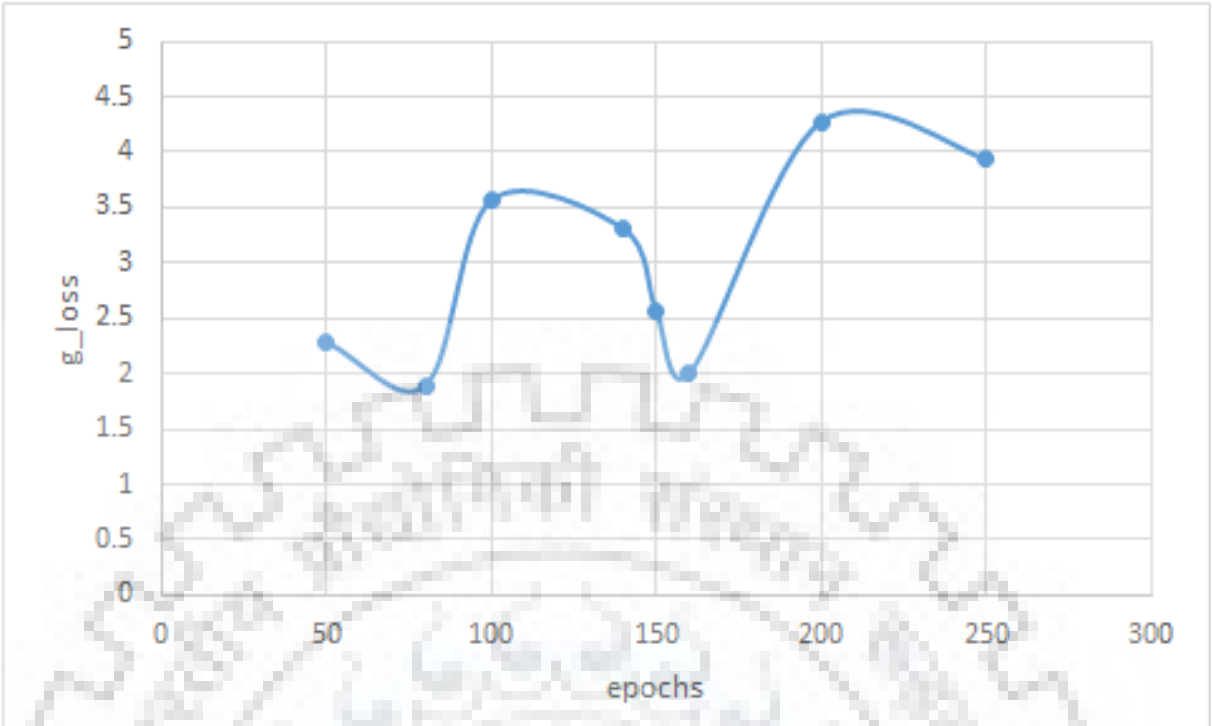

Figure 5.9: *Generator loss curve for the used model*

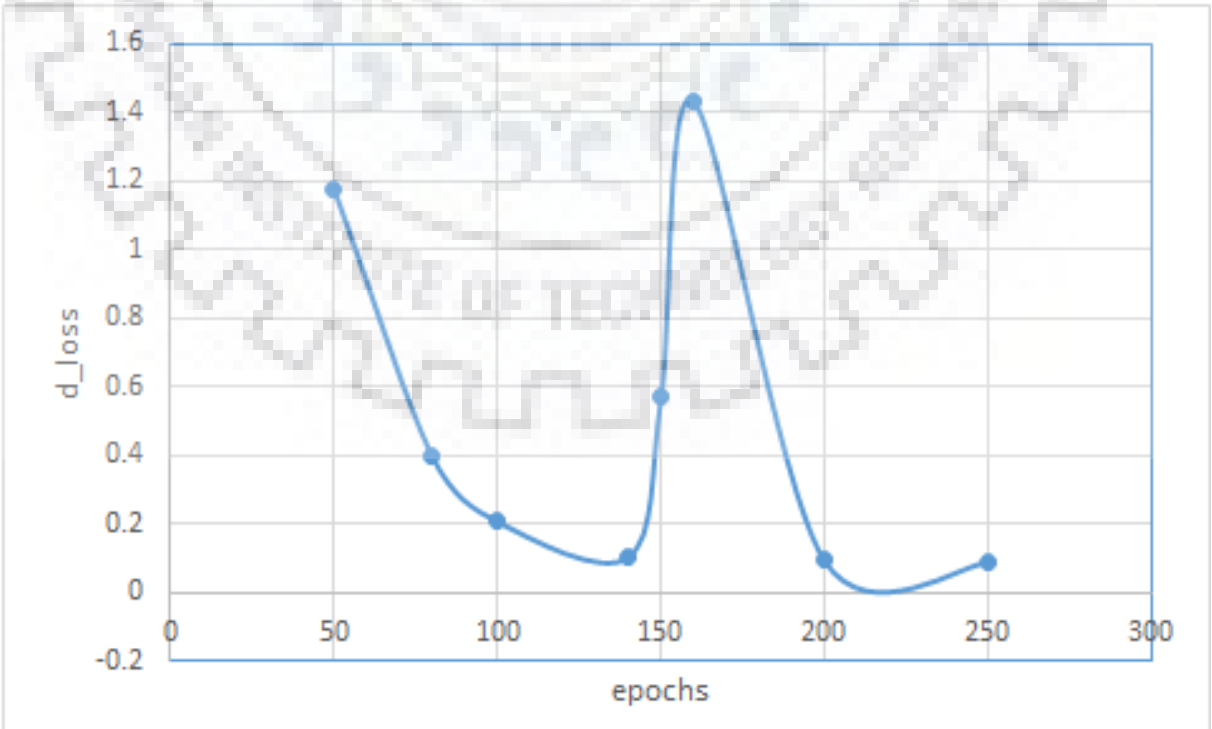

Figure 5.10: *Discriminator loss curve for the used model*

Dataset	Model used	Inception Score
Libor Spacek's facial images	**Text-to-image synthesis GAN**	**1.386± .012**
Oxford-102	Text-to-image synthesis GAN	2.66 ± .03
Oxford-102	Stack GAN	3.20 ± .01
Oxford-102	TACGAN	3.45 ± .05
CUB bird	Text-to-image synthesis GAN	2.88 ± .04
CUB bird	Stack GAN	3.70 ± .04
CUB bird	TACGAN	3.78 ± .05

Table 5.2: *Inception score comparison for various text-to-image generation models with the proposed face synthesis GAN model. [2] [3]*

attributes describing visual attributes as input to them using GAN model, we are comparing the inception score of our results with the inception score of other GAN models who perform the text-to-image synthesis task as well as those who are the state-of-art in the text-to-image conversion. The comparison helps us to find out how clear, sharp and diverse images our model is able to produced as compared to other models. The comparisons show that for the face synthesis task, the face images generated by our model are not as sharp and as clear as expected. Other comparison models are generating very clear and sharp images corresponding to the text caption given to them. The birds and flowers images generated by the Text-to-image GAN [1], flowers and bird images generated by the Stack GAN [3], and the flowers and bird images generated by TACGAN [2] are very sharp and clear compared to the outputs generated on our dataset using the model given by Reed *et al* [1].

As we can see from all the qualitative results in the previous section, the GAN model of Reed *et al* used by us performs the best for face synthesis task when the number of epochs are set to 150. The Table5.2 depicts the comparison of the used mode for face synthesis with other models used as state-of-art for the text-to-image synthesis task. The two graphs 5.9 and 5.10 depict that there is no relation between the generator and discriminator loss. Both try to improve own performance by making an attempt of defeating the other one. No standard curve is found for the generator and the discriminator loss. The value of each of the discriminator and generator loss can be variable depending upon the variable setting and the direction to which the training process is heading(positive or negative). The discriminator loss is the combination of loss across 3 pairs namely: (real image, right text), (real image, wrong text) and (fake image, right text). The mean cross entropy is ensured to be reduced across above stated pairs. The loss across

discriminator and generator try to train the model with gradient values so that each layer can update its weight values and hence modify all the parameters with the best values in order to give the best results. Adam optimizer is also used to improve the learning rate. These gradients when back propagated help make the network performance better.

Chapter 6 CONCLUSION

In this report, we show that the Generative Adversarial Network with matching aware discriminator as used by [1] can be used for the task of face synthesis given the text file describing the visual attributes of the respective person. The experiment results on the Libor Spacek's facial images dataset are not very good but they definitely show some scope of improvement. With some more modifications in the model, its parameters and by increasing the number of training samples used, we can further improve the performance of the GAN network for this face synthesis task.

The focus of the current work is to extract the visual features from the input text file given to the model and then synthesize face images that are as close as possible to the ground truth value. The model used by us as proposed by Reed *et al* [1] works amazingly well for the generation of images of birds and flowers but for the task of face generation, not all the vivid details of the texture and color of the face could be captured. Law enforcement field can get various benefits from such a system that can synthesize images of the criminals automatically from the text description of their visual features given as input to them. In case of any such crime investigation, the witness plays a vital role in giving the description of the criminal to the cops who then make a sketch of the criminal with the help of any artist. But this task can be automated with the help of the model used by us by merely giving the textual description of criminal's visual attributes as input to the trained model that can then give as output the face image of the criminal. With some more modifications in this model and using a larger dataset for training, a very huge advancement can be done in this process of law. The figure 5.2 show the qualitative performance of our model for the face synthesis while the Table5.2 show the quantitative result that depicts that the model for this particular application is not outperforming the other comparison models. The main reason for less score compared to other models is the less sharpness, clarity and diversity of the generated images.

In the presented work, we have shown the ability of our model to generate face images by only providing the textual description of the visual attributes of the desired person. The model can be easily extended to the Stack-GAN model used by Zhang *et al* [3] and the TACGAN model used by Dash *et al* [2]. These possible extensions can help generate more clear and sharp face images with more vivid and intricate details taken into account.

REFERENCES

[1] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

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

[3] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," *arXiv preprint arXiv:1612.03242*, 2016.

[4] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[5] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *arXiv preprint arXiv:1601.06759*, 2016.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[7] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[8] Y. Zhang, Z. Gan, and L. Carin, "Generating text via adversarial training," in *NIPS workshop on Adversarial Training*, 2016.

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

[10] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances In Neural Information Processing Systems*, 2016, pp. 613–621.

[11] A. Ghosh, V. Kulharia, and V. Namboodiri, "Message passing multi-agent gans," *arXiv preprint arXiv:1612.01294*, 2016.

[12] E. L. Denton, S. Chintala, R. Fergus *et al.*, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in neural information processing systems*, 2015, pp. 1486–1494.

[13] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, "Generating images with recurrent adversarial networks," *arXiv preprint arXiv:1602.05110*, 2016.

[14] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[15] X. Tang and X. Wang, "Face sketch synthesis and recognition," in *Computer vision, 2003. proceedings. ninth ieee international conference on*. IEEE, 2003, pp. 687–694.

[16] X. Di, V. A. Sindagi, and V. M. Patel, "Gp-gan: Gender preserving gan for synthesizing faces from landmarks," *arXiv preprint arXiv:1710.00962*, 2017.

[17] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, 2015, pp. 3483–3491.

[18] X. Di and V. M. Patel, "Face synthesis from visual attributes via sketch using conditional vaes and gans," *arXiv preprint arXiv:1801.00077*, 2017.

[19] N. Bansode and P. Sinha, "Face sketch generation using evolutionary computing."

[20] H. Wang and K. Wang, "Facial feature extraction and image-based face drawing," in *Signal Processing, 2002 6th International Conference on*, vol. 1. IEEE, 2002, pp. 699–702.

[21] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 49–58.

[22] Z. Zhou, W. Zhang, and J. Wang, "Inception score, label smoothing, gradient vanishing and-log (d (x)) alternative," *arXiv preprint arXiv:1708.01729*, 2017.

[23] S. Barratt and R. Sharma, "A note on the inception score," *arXiv preprint arXiv:1801.01973*, 2018.

[24] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," *arXiv preprint arXiv:1610.09585*, 2016.

Shivali

ORIGINALITY REPORT

5%	1%	3%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Submitted to University College London Student Paper	1%
arxiv.org Internet Source	1%
Xu Tao, Zhou Yun. "Fall prediction based on biomechanics equilibrium using Kinect", International Journal of Distributed Sensor Networks, 2017 Publication	<1%
Pedram Ghamisi, Naoto Yokoya. "IMG2DSM: Height Simulation From Single Imagery Using Conditional Generative Adversarial Net", IEEE Geoscience and Remote Sensing Letters, 2018 Publication	<1%
Submitted to University of Edinburgh Student Paper	<1%
Submitted to UT, Dallas Student Paper	<1%
"Computer Vision – ECCV 2016", Springer	

Nature, 2016 Publication	<1%
"Computational Linguistics", Springer Nature, 2018 Publication	<1%
Kenji Enomoto, Ken Sakurada, Weimin Wang, Hiroshi Fukui, Masashi Matsuoka, Ryosuke Nakamura, Nobuo Kawaguchi. "Filmy Cloud Removal on Satellite Imagery with Multispectral Conditional Generative Adversarial Nets", 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 Publication	<1%
"Artificial Neural Networks and Machine Learning – ICANN 2017", Springer Nature, 2017 Publication	<1%
Submitted to National University of Singapore Student Paper	<1%
Submitted to University of Queensland Student Paper	<1%
S. Palazzo, C. Spampinato, I. Kavasidis, D. Giordano, M. Shah. "Generative Adversarial Networks Conditioned by Brain Signals", 2017 IEEE International Conference on Computer	<1%