Generative AI Assignment - 2

Zaraar Malik
i212705@nu.edu.pk
Artificial Intelligence Department
National University of Computer and Emerging Sciences

***Abstract*—This project explores various Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) applied to different image generation tasks. The first phase involved separately implementing a Simple GAN and a Variational Autoencoder (VAE) to generate signature images collected from the class. The GAN learns to create realistic signatures, while the VAE captures the underlying distribution to generate new signature samples. In the second phase, a GAN integrated with a Siamese Network was trained on the Cats and Dogs images from the CIFAR-10 dataset. This Siamese GAN predicts a similarity score between real and fake images during the generator's training process, enhancing the model's capability to distinguish between real and synthetic data.**

Next, a Conditional GAN was implemented to generate real face images from corresponding face sketches. A latent space representation of the sketch, compressed to a size of 128, was fed into the generator to produce realistic facial images. Finally, a CycleGAN was used to perform image-to-image translation between face sketches and real face images. This two-way translation enabled the generation of realistic face images from sketches and vice versa. Each model demonstrates different aspects of GANs' capacity to learn complex data distributions, offering valuable insights into the potential of generative models in real-world applications such as signature synthesis, face generation, and sketch-to-image translation.

I. Question Number 1

A. *Introduction*

This was the First Question which asked each student to Train the Following 2 Models:

1) **VAE** : Variational Auto Encoder
2) **GAN** : Generative Adversarial Network

The purpose was to Reconstruct the Signature Images from a Random Noise Vector Collected from the Normal Distribution after Training the Models.

B. *Dataset Collection*

In this question, the dataset selected was basically collected in the previous assignment. It was the Signatures Dataset which consisted of the Signatures of Each Student in the Course of Generative AI and was personally collected by the respected Course Instructor. However, I use the Signatures which were manually annotated in order to keep the results of the Model up to the mark. Using DIP preprocessing Techniques resulted in less accurate Results of the processed images so it was finally concluded to use manually annotated images so that the model training can be streamlined.

C. *Data Preparation and Data Loaders*

After finalizing the dataset, as the size of the dataset was quite large, I made a custom Dataset class which will be used to load our Signature Dataset from the Directory into a new Format which can be used to easily Train the Models without the problem of running out of memory due to RAM crashes. Moving on, I made 3 Custom Datasets which are listed below:

1) **Training Dataset**
2) **Validation Dataset**
3) **Testing Dataset**

Furthermore, Each dataset was then passed onto the ”Data loader” which prepared the data into batches so that we can then start the training of our models. The preprocessing Techniques which i used on the Dataset are listed below:

1) **Rotate Right (15 Degrees)**
2) **Rotate Left (15 Degrees)**
3) **Enhancing Brightness and Contrast**
4) **Random Crop with size 64x128**
5) **Image Normalization**

D. *VAE Model Architecture*

Now the next step was to design the Variational Auto Encoder which was the First part to do in the provided question. Following is the Encoder of my VAE:

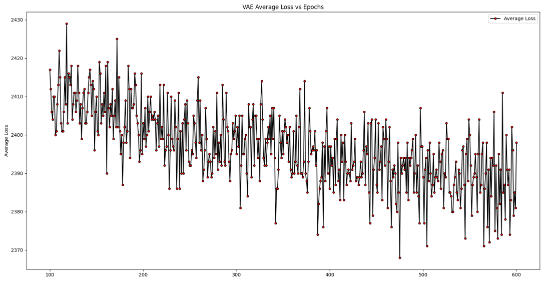
1) **Conv2d (results in 32,64,32 size)**
2) **Activation (ReLU)**
3) **Conv2d (results in 16,32,64 size)**
4) **Activation (ReLU)**
5) **Conv2d (results in 8,16,128 size)**
6) **Activation (ReLU)**
7) **Conv2d (results in 4,8,128 size latent dim**
8) **Activation (ReLU)**

Now, the second part of the VAE was the Decoder which is mentioned as following:

1) **ConvTranspose2d (results in 8,16,128 size)**
2) **Activation (ReLU)**
3) **ConvTranspose2d (results in 16,32,64 size)**
4) **Activation (ReLU)**
5) **ConvTranspose2d (results in 32,64,32 size)**
6) **Activation (ReLU)**
7) **ConvTranspose2d (results in 64, 128, 3 size)**
8) **Activation (Sigmoid)**

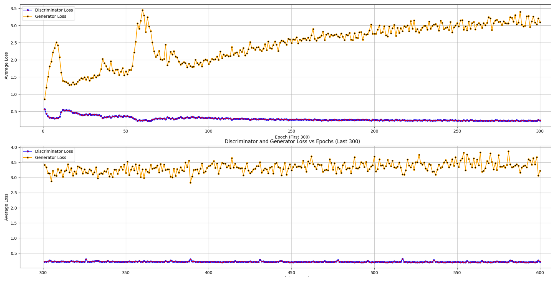
E. VAE Model Loss During Training

This is the Loss of the VAE during the Training Process :



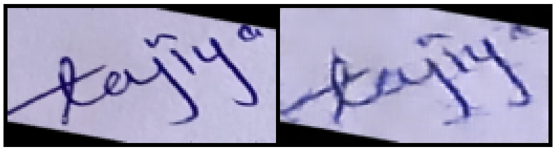
F. GAN Model Loss During Training

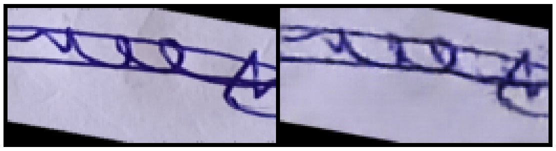
This is the Loss of the GAN during the Training Process :

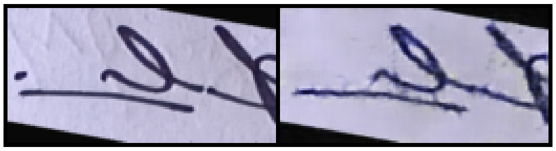


G. Results of VAE

Following are the Results of the VAE during the Training Process





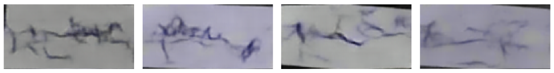


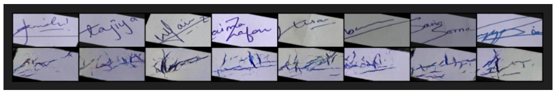
H. VAE Inferrence Output

Following are the Reults of the VAE during Inferrence using a Random Noise :

I. GAN Training Output

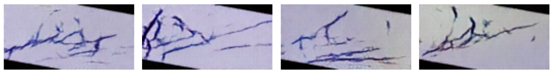
Following are the Results of the GAN during the Training Process





J. GAN Inferrence Output

Following are the Results of the GAN during the Inferrence :



II. Question Number 2

A. Introduction

In this question, the dataset selected was basically the CIFAR-10 Dataset. We were assigned to just keep the images of Cats and Dogs in the Dataset. Then we were supposed to train the following Custom Model:

1) **SN-GAN**: Siamese Network Based Generative Adverserail Network

B. Model Architecture

The Generator of the Model is as following:

1) **ConvTranspose2d (results in 256,4,4 size)**
2) **Activation (ReLU)**
3) **ConvTranspose2d (results in 128,8,8 size)**
4) **Activation (ReLU)**
5) **ConvTranspose2d (results in 64,16,128 size)**
6) **Activation (ReLU)**
7) **ConvTranspose2d (results in 3,32,32 size latent dim**
8) **Activation (Tanh)**

The Discriminator was basically a simple Siamese Network which i implemented after a through study from the internet. It takes input 2 different images and then predicts a probability score which is basically similarity score in this case between the 2 input images or specifically, the real image and the generated image. The Discriminator of the Model is as following:

1) **Conv2d (results in 16x16 size)**
2) **Activation (Leaky ReLU)**
3) **Conv2d (results in 8x8 size)**
4) **Activation (Leaky ReLU)**
5) **Fully Connected Layer**
6) **Activation (Leaky ReLU)**
7) **Fully Connected Layer**

C. Conclusion

The main issue with which I faced in this network was that I was completely unable to train the network. Despite alot of effort, the Generator loss never decreased and produced quite useless outputs which cannot be considered as something.

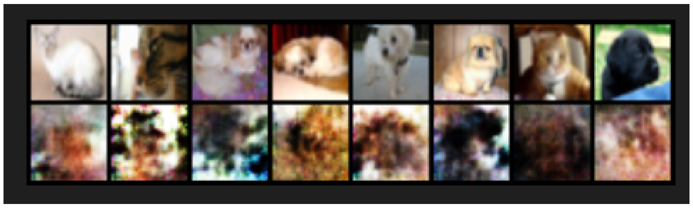


Fig. 1. This is a sample of the Training output

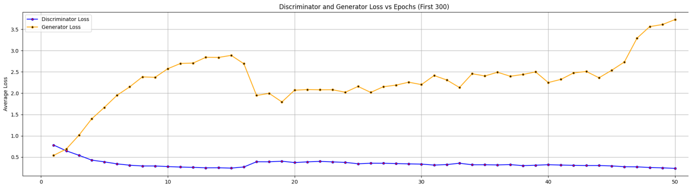


Fig. 2. This is the GAN Loss

Following are the Generated Outputs during the Training: Following is the Loss of the Model During Training Process:

Following are the Generated Outputs using a random vector:

Fig. 3. This is a sample of the Inferrence output

III. Question Number 3

A. *Introduction*

In this question, we were asked to implement a Conditional GAN architecture for which the Conditions were basically the Sketches of Human Faces. Based on the Human Sketch, we were supposed to make a Realistic image of the specified human sketch.

B. *Introduction**Model Architecture***

So, the model architecture in simlpe terms was such that the Sketch was encoded into a latent vector. Then this latent vector was fed into the generator. The Generator gave an output which was then compared with the original image and then in the same manner the model was trained as the model in the previous question.

C. *Conclusion*

The trainig was quite dificult because first issue was GPU resources. The next was model optimization, checkpointing, etc etc. Following are the results

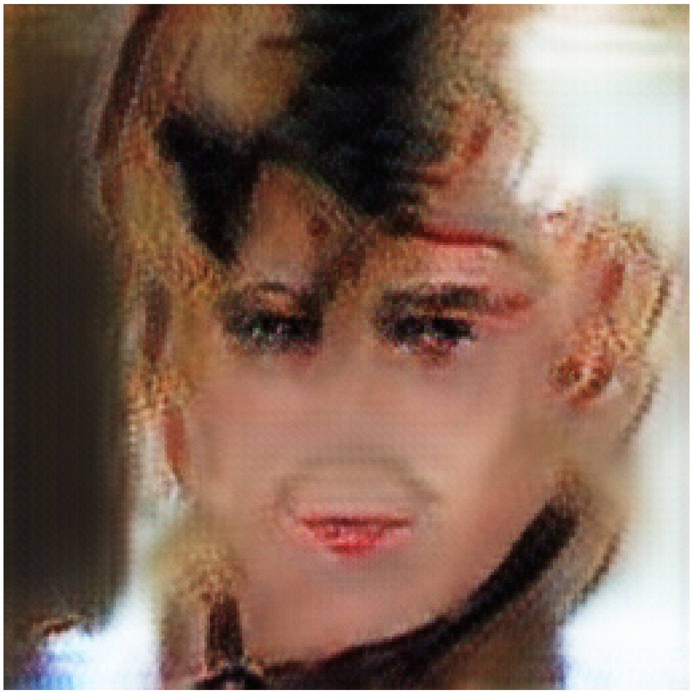


Fig. 4. This is a sample of the Training output

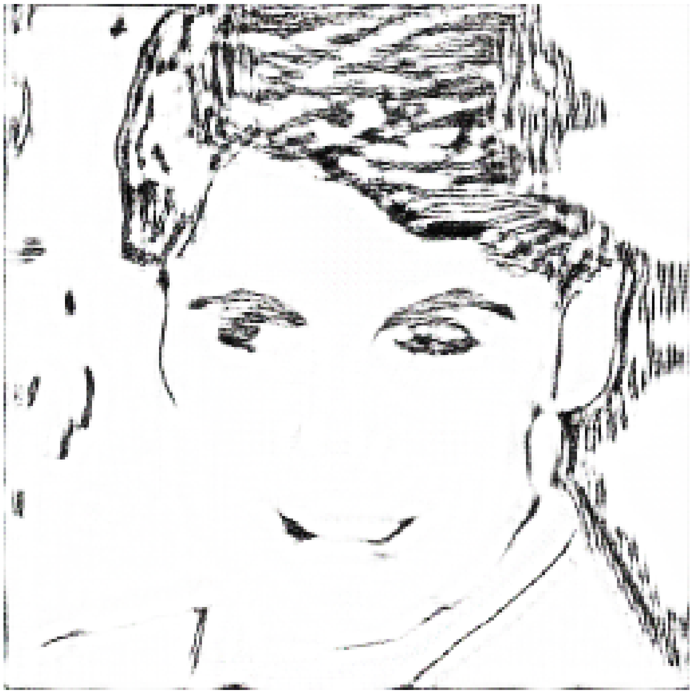


Fig. 5. This is a sample of the Conditional Input

IV. Question Number 4

A. *Introduction*

This is the Last Question of the Assignment and the most difficult question of the assignment. It took nearly 2 days just to train the model and roughly around 1 week to get some reuslt's from the Model.

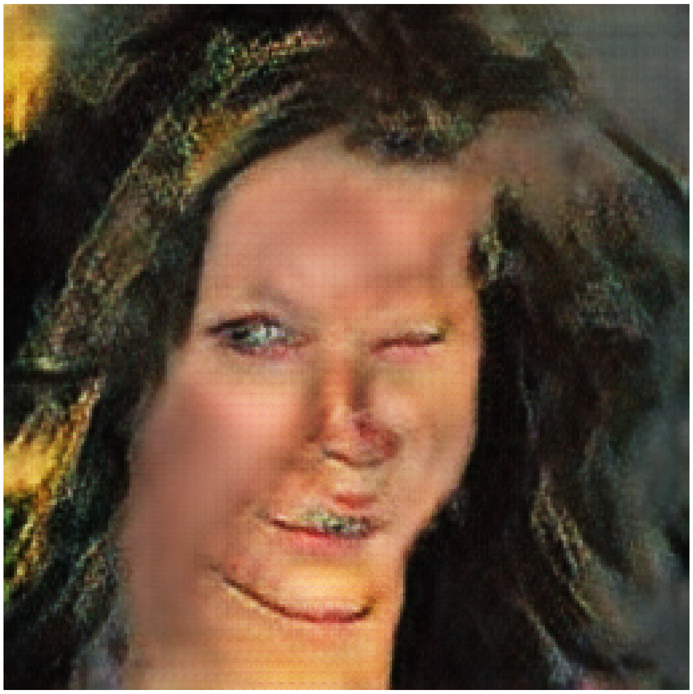


Fig. 6. This is a sample of the Testing output

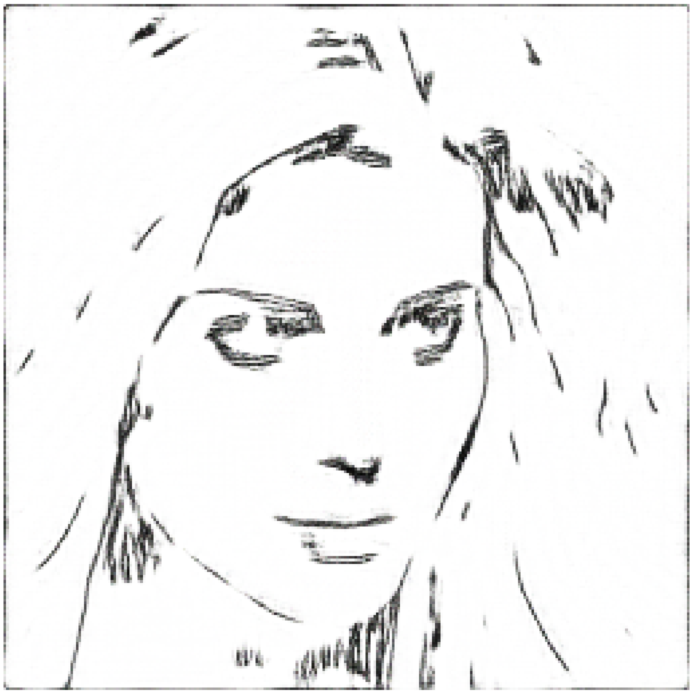


Fig. 7. This is a sample of the Conditional Input

B. Model Architecture

The Model architecture was quite complex it self however i will try my best to explaing the model. This model had the Following main compenents:

1) **Generator for Human Faces**
2) **Generator for Human Face Sketches**
3) **Discriminator for Human Faces**
4) **Discriminator for Human Face Sketches**

C. Dataset

The dataset used in this question was the person image sketches dataset. This was provided to us in the assignment and this dataset is available in the Kaggle Datasets area. It consisted of Human Face Images and their Corresponding Face Sketches.

D. The entire Concept

The entire concept was to implement Image-to-Image Translation in this question. For a provided Human Sketch, the model has to learn the Real Human face and generate the specified Human Face and for a provided Human Face, the model has to find out the Human Sketch by observing the Real Human Images.