EE698R: Advanced Topics in ML
Project Report

Rishi Agarwal (210849)
Shivam Sharma (210983)

1 Introduction

The course content focused on advanced techniques in Machine Learning, specifically on Generative Learning. The introduction of GANs has been revolutionary in the area of unpaired image to image translation. Given the focus on research-centric content in the course and the stress of implementing models from scratch, we decided to evaluate three different models from past literature in the field of generative learning. Section 2 details the datasets used in the project and a description of the target task. In sections 3, 4, and 5, we give a brief theoretical overview of the three models and their tasks. Section 6 compares these models' performance on the same task and data based on qualitative assessment, as in the original papers.

2 Target task and Data

2.1 Models Used

The three models that we have used are:

- VAE-GAN (Ref1)
- CycleGAN (Ref2)
- StarGAN (Ref3)

2.2 Task

We have explored how the area of unpaired image-to-image translation has evolved over the years. We perform image-to-image translation on the datasets given in Section 2.3 for all the three models described above. Specifically, we train the models to learn certain visual attributes from the data distribution and then augment images accordingly to perform style transfer. The models are listed in order of their public release.

2.3 Datasets

We have used two datasets for our experimentation.

- CelebA - The popular CelebA dataset contains celebrity faces in the form of images with 40 annotated features which are one-hot encoded. Of the three models, VAE-GAN and StarGAN were already tested on this data. We implemented CycleGAN on this as a novel addition and verified the results for the other two papers. Here we attempt to alter facial features in the translation objective.
- DeepFashion - First appeared in 2022, this data contains full-size images of models in various types of clothing. This task forms the core of the project where we aim to test the extent to which old models stand up to newer, more complex data. The translation objective here is to alter attributes related to clothing (tops, jeans, shorts etc.) while keeping the models intact.

Preprint. Under review.

2.4 Compute

All the models were trained within 12 hours on the GPU02 IITK server with Nvidia A100s.

3 VAE-GAN

3.1 Overview

Introduced in 2016, this paper is a modification to the original GAN architecture. It replaces the generator of the GAN by a VAE. Formally, this paper introduces a new autoencoder architecture called the Adversarial Autoencoder (AAE) that can learn a probabilistic latent representation of data in an unsupervised manner. The key idea is to combine the traditional autoencoder objective with an adversarial training procedure that forces the latent distribution to match a desired prior, such as a Gaussian.

Let $\mathbf{x}$ be the input data, $\mathbf{z}$ be the latent representation, and $p(\mathbf{z})$ be the desired prior distribution. The AAE consists of three components:

- An encoder $q_\phi(\mathbf{z}|\mathbf{x})$ that maps the input to the latent representation.
- A decoder $p_\theta(\mathbf{x}|\mathbf{z})$ that reconstructs the input from the latent code.
- A discriminator $D_\psi(\mathbf{z})$ that tries to distinguish latent codes sampled from the aggregate posterior $q_\phi(\mathbf{z})$ from those sampled from the prior $p(\mathbf{z})$.

The AAE is trained by minimizing the reconstruction loss $\mathcal{L}rec = -\mathbb{E}q_\phi[\log p_\theta(\mathbf{x}|\mathbf{z})]$ while simultaneously maximizing the adversarial loss $\mathcal{L}adv = -\mathbb{E}q_\phi[\log D_\psi(\mathbf{z})] - \mathbb{E}p(\mathbf{z})[\log(1 - D\psi(\mathbf{z}))]$.

This allows the AAE to learn a latent representation that is both useful for reconstruction and matches the desired prior distribution, enabling tasks like generation, interpolation, and disentanglement.

To perform the style transfer, latent representation of the images is averaged according to the presence/absence of class labels. The idea is that by finding out the average encoding of the different class labels in the latent space and finding the difference in the averages of images which do and do not have that label represents a certain "direction" in the latent space that nudges an image towards having the attributes encoded by this particular label.

Formally, let $\mathbf{x}$ be an input image and $\mathbf{z} = q_\phi(\mathbf{x})$ be its latent representation. Let $\bar{\mathbf{z}}_{\text{yes}}$ be the average latent representation of images with a certain attribute (e.g. "smiling"), and $\bar{\mathbf{z}}_{\text{no}}$ be the average latent representation of images without that attribute. Then, the translation of the input image $\mathbf{x}$ to have (or not have) the given attribute can be done by adding (or subtracting) the difference between the two average latent representations: $\hat{\mathbf{z}} = \mathbf{z} \pm (\bar{\mathbf{z}}_{\text{yes}} - \bar{\mathbf{z}}_{\text{no}})$

Where the sign of the difference term depends on whether we want to add or remove the attribute. The translated image is then generated by passing $\hat{\mathbf{z}}$ through the decoder: $\hat{\mathbf{x}} = p_\theta(\hat{\mathbf{z}})$. This allows the model to perform attribute-based image translation by directly manipulating the latent representations, without needing to train a separate translation model.

3.2 Model Architecture

Following is an illustration of the general training loop for the model

There are two separate architectures used for the two datasets that have been used for evaluation. The CelebA architecture is taken directly from the original paper, while the architecture for DeepFashion dataset has been handcrafted by taking inspiration from the VAE-GAN paper as well as the paper that introduced the DeepFashion Dataset. We have tried various layer sizes and latent space dimensions for the latter and the following is the one that gave the best results.

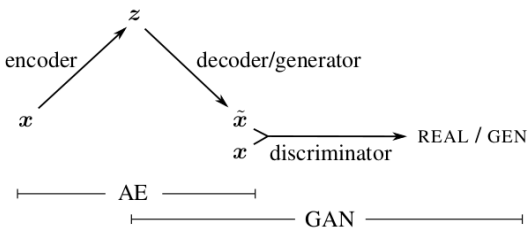


Figure 1: Network Overview, taken from (Ref1)

3.2.1 CelebA

Here the images were originally 178*218 in dimensions which were center-cropped and resized to 64*64. The latent space dimension used was $\mathbf{z}_{\text{size}}$=128.

Encoder	Decoder	Discriminator
5×5 64 conv. ↓, BNorm, ReLU	8·8·256 fully-connected, BNorm, ReLU	5×5 32 conv., ReLU
5×5 128 conv. ↓, BNorm, ReLU	5×5 256 conv. ↑, BNorm, ReLU	5×5 128 conv. ↓, BNorm, ReLU
5×5 256 conv. ↓, BNorm, ReLU	5×5 128 conv. ↑, BNorm, ReLU	5×5 256 conv. ↓, BNorm, ReLU
2048 fully-connected, BNorm, ReLU	5×5 32 conv. ↑, BNorm, ReLU	5×5 256 conv. ↓, BNorm, ReLU
	5×5 3 conv., tanh	512 fully-connected, BNorm, ReLU
		1 fully-connected, sigmoid

Table 1: Architectures for the three networks that comprise CelebA VAE/GAN. ↓ and ↑ represent down- and upsampling respectively

3.2.2 DeepFashion

Here the images were originally 512*1028 in dimensions which were resized to 128*256. The latent space dimension used was $\mathbf{z}_{\text{size}}$=256.

Encoder	Decoder	Discriminator
5×5 64 conv. ↓, BNorm, L-ReLU	32·16·256 fully-connected, BNorm, L-ReLU	5×5 32 conv., L-ReLU
5×5 128 conv. ↓, BNorm, L-ReLU	5×5 256 conv. ↑, BNorm, L-ReLU	5×5 128 conv. ↓, BNorm, L-ReLU
5×5 256 conv. ↓, BNorm, L-ReLU	5×5 128 conv. ↑, BNorm, L-ReLU	5×5 256 conv. ↓, BNorm, L-ReLU
1024 fully-connected, BNorm, L-ReLU	5×5 32 conv. ↑, BNorm, L-ReLU	5×5 256 conv. ↓, BNorm, L-ReLU
	5×5 3 conv., tanh	512 fully-connected, BNorm, L-ReLU
		1 fully-connected, sigmoid

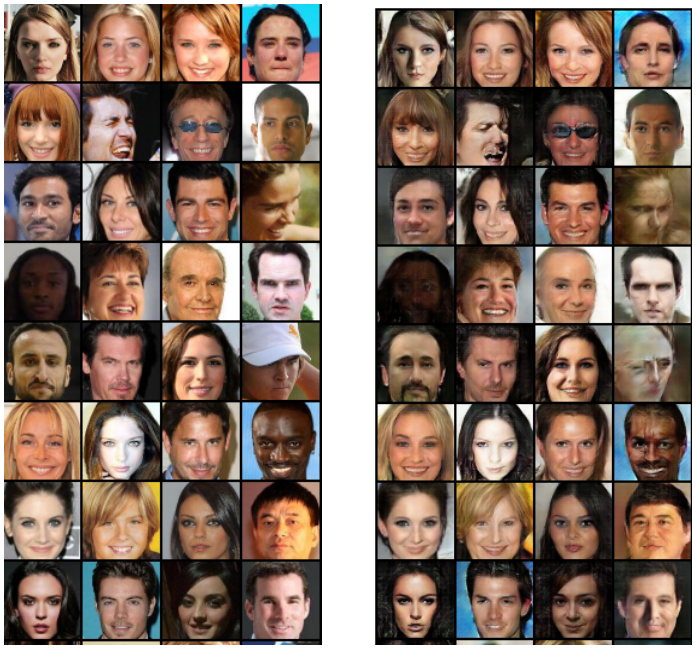
Table 2: Architectures for the three networks that comprise DeepFashion VAE/GAN. ↓ and ↑ represent down- and upsampling respectively. L-ReLU represents Leaky ReLU with linear region slope of 0.2

3.3 Results

Following are some selected results from the model for both the datasets

3.3.1 CelebA

The reconstruction of a random batch of 32 is given in the following image



(a) Original (b) Reconstructed

Figure 2: Original and Recreated Images from the CelebA dataset

We note that the results closely match those presented in the paper. This is expected as the model architecture is exactly the same and the feature space is limited.
Following is an example of style transfer task on the data.

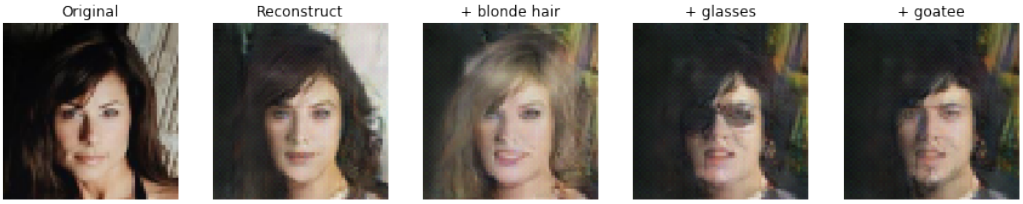


Figure 3: Attribute translation in CelebA

We note that again the model is able to accurately translate the target features and also maintain other attributes as more or less unchanged. Some effect of correlation among features in the data is observed on the translation.

3.3.2 DeepFashion

The reconstruction of a random batch of 16 is given in the following image

(a) Original (b) Reconstructed

Figure 4: Original and Recreated Images from the DeepFashion dataset

Here, we note that the model generalizes well on the samples but fails to capture the finer details, especially for the facial features and prints on the clothes. This is possibly due to the large input image sizes and too many features to be encoded in a relatively small latent space (256). We remark that increasing the latent space beyond 256 within computation capability available to us did not yield significantly better results.
Following is an example of style transfer task on the data.

Figure 5: Attribute translation in DeepFashion

Here we have taken a sample and translated it to WOMAN_Cardigan, WOMAN_Shorts, WOMAN_Jacket, WOMAN_Dress, MAN_Sweater repectively. We note that the vectors capture the general idea but the translation is not as crisp and there is significant interference from correlated attributes. We remark that this is probably because of the relatively small latent space dimensionality and also due to comparatively smaller label:sample ratio in the data for some classes.

4 CycleGAN

Introduced in 2017, this paper modifies the orignal GAN architecture to learn a specific characteristics from 2 sets of images and also to translate an image from one set to another.

Let $\mathbf{X}$ and $\mathbf{Y}$ be classes of images. We wish to learn two mappings, $\mathbf{X} \rightarrow \mathbf{Y}$ and $\mathbf{Y} \rightarrow \mathbf{X}$. Define generators $\mathbf{G}$ and $\mathbf{F}$ that do the same. Additionally, we have discriminators $\mathbf{D_x}$ and $\mathbf{D_y}$. Their job is to predict the probabilty of an image belonging to class $\mathbf{X}$ and $\mathbf{Y}$ respectively. The losses are defined as-

- **Adversarial Loss**

$$\mathcal{L}(G, D_Y) = \mathbb{E}_{y \sim p(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p(x)}[\log(1 - D_Y(G(x)))] \tag{1}$$

$$\mathcal{L}(F, D_X) = \mathbb{E}_{x \sim p(x)}[\log D_X(x)] + \mathbb{E}_{y \sim p(y)}[\log(1 - D_X(F(y)))] \tag{2}$$

  The generators try to minimize their respective objectives and the corresponding discriminators are trained to maximize it.

- **Cycle Consistency Loss**

$$\mathcal{L}(G, F) = \mathbb{E}_{x \sim p(x)}[\|F(G(x)) - x\|] + \mathbb{E}_{y \sim p(y)}[\|G(F(y)) - y\|] \tag{3}$$

  Both generators are made to minimize this loss. This ensures that the images generated by the models are not too dissimilar from the orignal input and retain enough information to be translated back.

Apart from these, there is also an optional identity loss, which is useful to preserve the color gradient in some datasets.

4.1 Network Architectures

The same architecture as described in the orignal paper, with changes made to the input and output shape.

4.2 Results

4.2.1 CelebA

The model was trained to convert between black and blonde hair. It was trained on a resolution of $256 * 256$ with a batch size of 8.

Below are results from a random batch-

- Black to Blond

Figure 6: Orignal

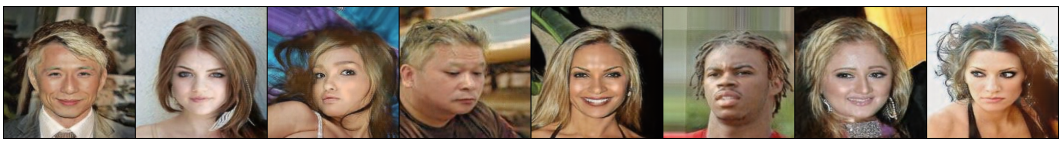


Figure 7: Translated

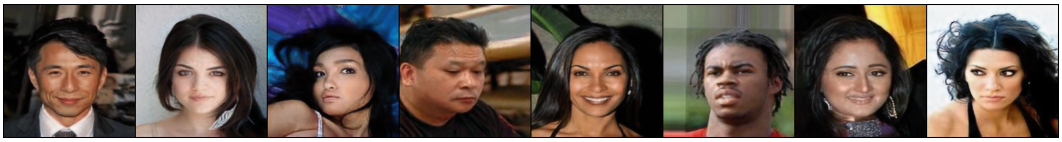


Figure 8: Recreated

- Blond to Black

Figure 9: Orignal

Figure 10: Translated

Figure 11: Recreated

Since changing the hair color only involves a slight change to the original image, the model performs very well and is able to generate good results.

4.2.2 DeepFashion

The model was trained to convert full sleeved clothing to sleeveless on a resolution of $256 * 128$ with a batch size of 32. Below are the results for a random batch

- Sleeved to Sleeveless

Figure 12: Orignal

Figure 13: Translated

Figure 14: Recreated

- Sleeveless to Sleeved

Figure 15: Orignal

Figure 16: Translated

Figure 17: Recreated

Below are some cherry picked results,

Figure 18: Sleeved to Sleeveless

Figure 19: Sleeveless to Sleeved

Also an interesting observation was that since there is high correlation between wearing full sleeves and full length lower clothing, the effect similar to the below instance was observed quite often-

Figure 20: Undesirable Change due to correlation

Overall, we can see that the GAN architecture is able to generate crisp and higher quality images.

5 StarGAN

StarGAN, introduced in 2018, is an attempt to generalize the idea of CycleGAN to mutli domain attribute translation.

In CycleGAN, 2 generators and 2 discriminators are needed to translate between two attributes. If the number of attributes is $\mathbf{N}$, to translate freely betweeen them, $2 * \mathbf{N} * (\mathbf{N} - \mathbf{1})$ networks need to be trained. This is not feasible to do for even small values of N.

StarGAN solves this issue by having just one generator and discriminator for translating between any attributes.

For $\mathbf{N}$ attributes, the presence of each attribute for an image can be represented in form of a one-hot-encoded vector. The generator takes has input the current image and the target attribute's one-hot-encoded vector as input. The discriminator takes as input an image and outputs an $\mathbf{N}$ length vector, representing the probabilities of presence of all attributes, along the the probability of the image being a real or a fake. The defined loss functions are,

- **Adversarial Loss**

$$\mathcal{L}_{adv}(G, D) = \mathbb{E}_x[\log D(x)] + \mathbb{E}_{x,c}[\log(1 - D(G(x, c)))] \tag{4}$$

  In classic adversarial pattern, $\mathbf{G}$ tries to minimize this while $\mathbf{D}$ tries to maximize it.

- **Domain Classification Loss**

$$\mathcal{L}_{cls}(D) = \mathbb{E}_{x,c_r}[-\log D(c_r|x)] \tag{5}$$

$$\mathcal{L}_{cls}(G) = \mathbb{E}_{x,c}[-\log D(c|G(x, c))] \tag{6}$$

  This is the loss introduced to deal with domain classification. The discriminator minimizes its loss to ensure that it classifies the real images into the correct classes and the generator minimizes its loss to so that the fake images created by it are classified correctly by the discriminator.

- **Cycle Consistency**

$$\mathcal{L}_{cyc}(G) = \mathbb{E}_{x,c_r,c}[\|x - G(G(x, c), c_r)\|] \tag{7}$$

  This loss is adopted from the CycleGAN paper, to ensure that the transformed images are not too different from the originals.

5.1 Network Architecture

The architecture implemented is exactly as described in the original paper and is similar to CyleGAN. The attribute vector is repeated and concatenated with the image along the first dimension.

There are several engineering optimizations used in the paper, such as using the WGAN adversarial objective with gradient penalty instead of the traditional GAN objective for more stable training. Additionally, in order to get a valid target attribute vector, the original vectors in the batch are shuffled.

5.2 Results

5.2.1 CelebA

The model was trained with the following($\mathbf{N} = 7$) attributes - Black-Hair, Blond-Hair, Male/Female, Young/Old, Bangs, Mustache, Smiling. The resolution was $\mathbf{128 * 128}$ and the batch size was 8.

Below are some results for single and multi attribute translation.

- Black to Blond(for comparision with CycleGAN)

Figure 21: Black to Blond Translation

- Blond to Black(for comparision with CycleGAN)

Figure 22: Blond to Black Translation

Figure 23: Multi Attribute Translation

- Male, Not Smiling to Black Hair, Female, Smiling

We find that the model is suitable for making small texture and color based changes to the original image but does not do as well at doing more involved, structural changes, For example below is the example for $(BlackHair, NoBangs, Smiling) \rightarrow (Blond, Bangs, Nosmile)$.

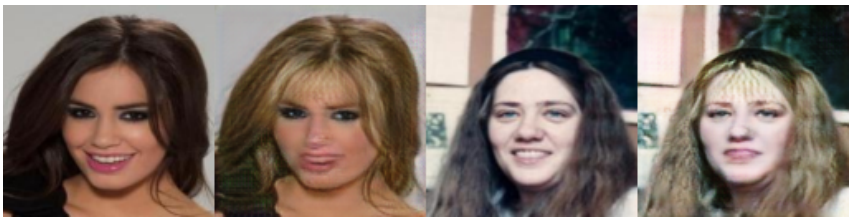


Figure 24: More involved changes

6 Future Work

- Due to time constraint, an orignal task on the DeepFashion dataset could not be performed on the StarGAN model.
- Explore more recent latent diffusion and flow based stragegies for image to image translation.