CIS 680, Vision & Learning, Fall 2018: HW3

Abhijeet Singh, Jagtar Singh, Shashank Garg

December 13, 2018

1 Variational Autoencoders

1.1 Autoencoder

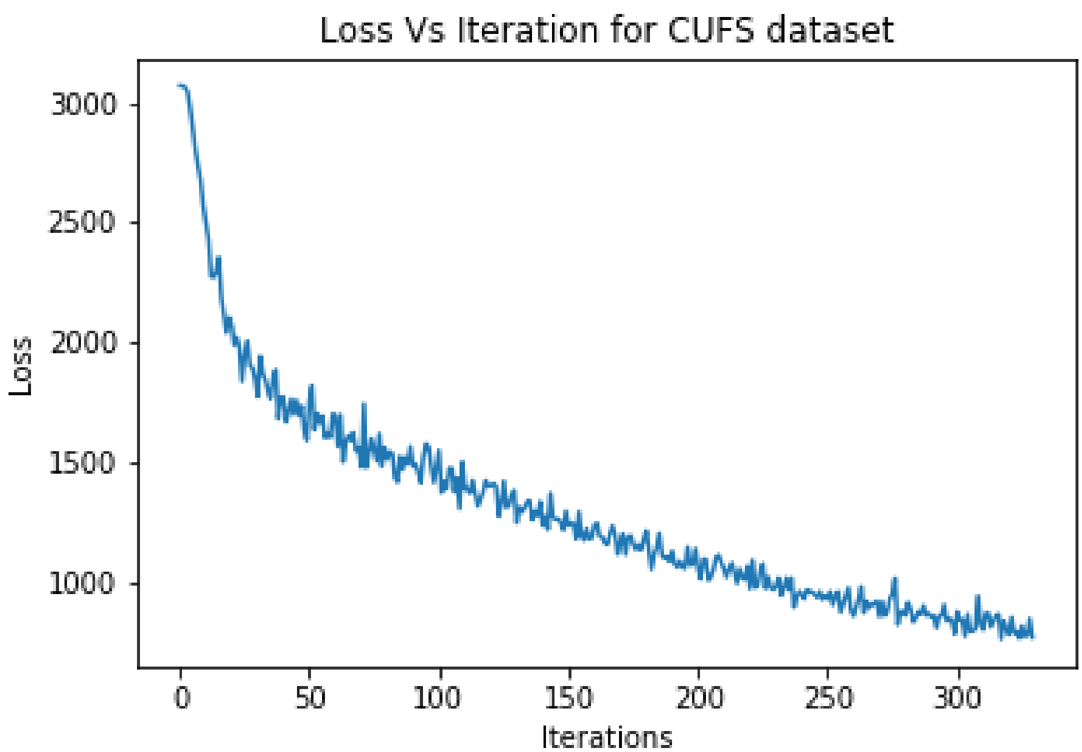


Figure 1: Autoencoder loss

Figure 2: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 3: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 4: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 5: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 6: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 7: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 8: Final Result(First Two Rows: Actual Images, Last Two Row: Generated Images)

1.2 Variational Autoencoder

1.2.1 Test Images(CUFS)

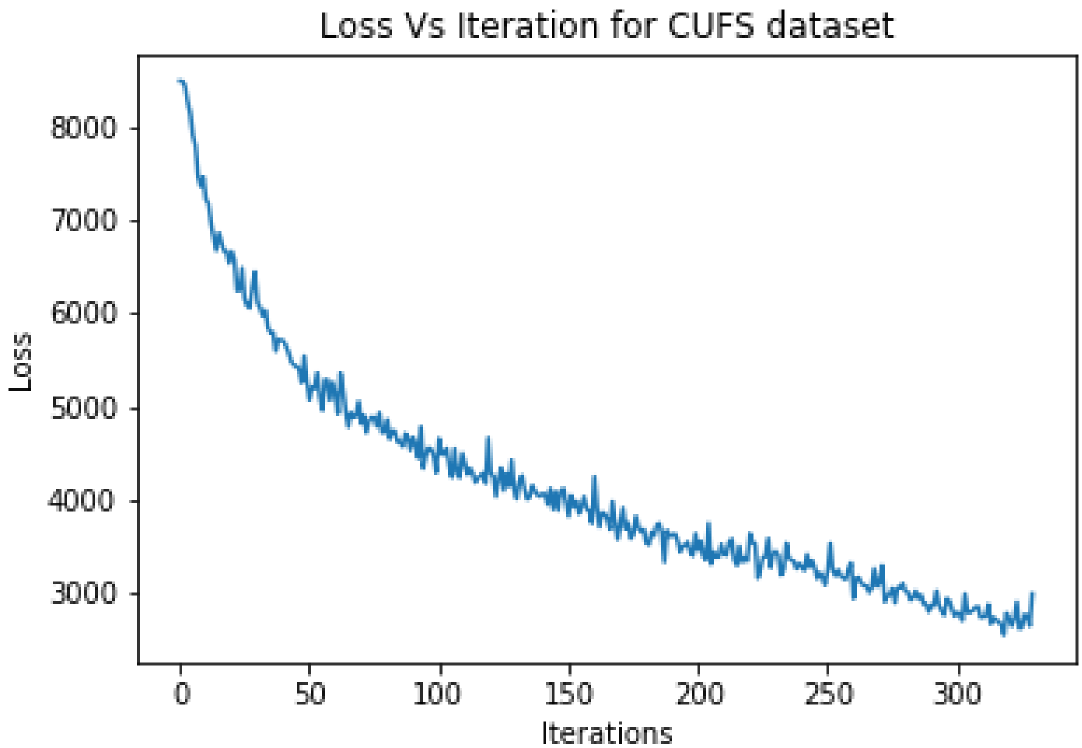


Figure 9: Variational Autoencoder loss

Figure 10: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 11: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 12: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 13: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 14: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 15: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 16: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

1.2.2 Images from Noise(CUFS)

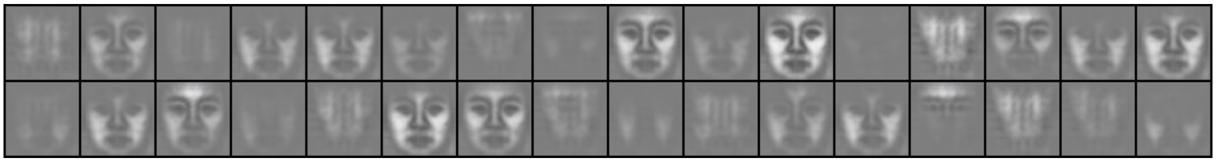


Figure 17: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

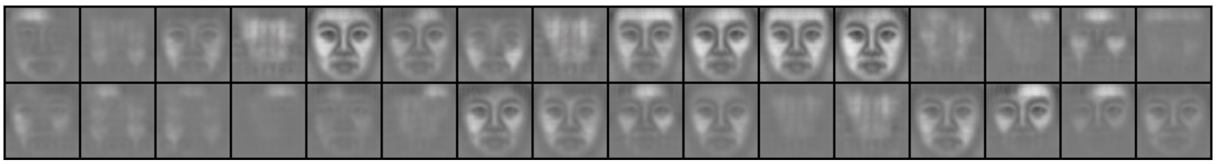


Figure 18: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

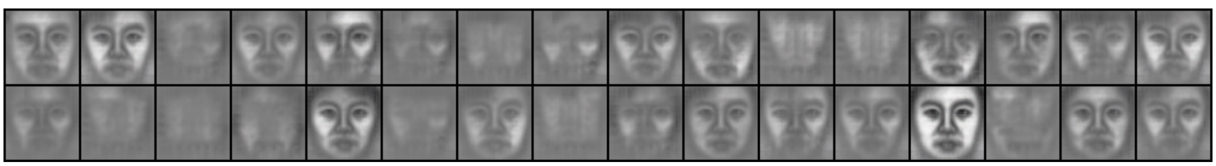


Figure 19: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

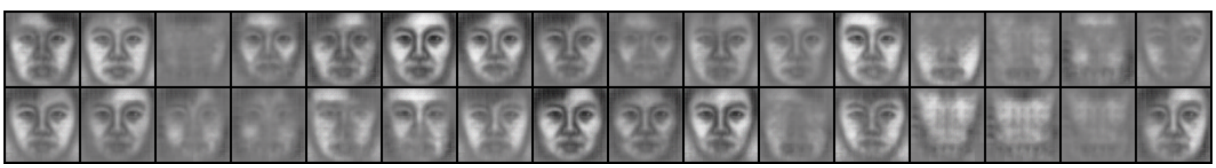


Figure 20: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

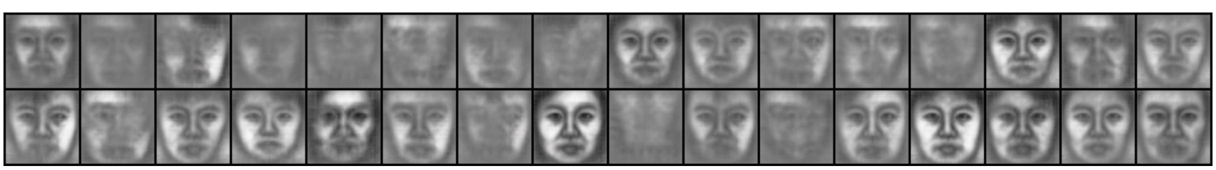


Figure 21: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

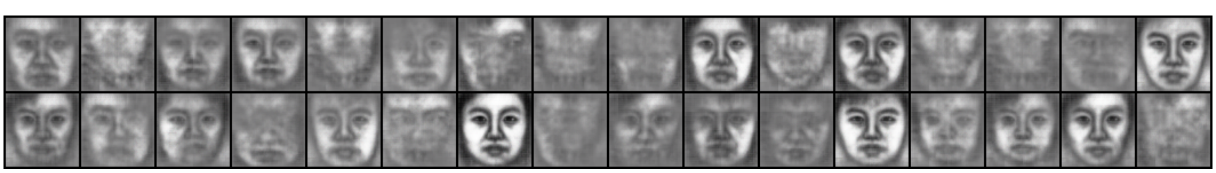


Figure 22: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

1.3 VAE on CelebA Dataset

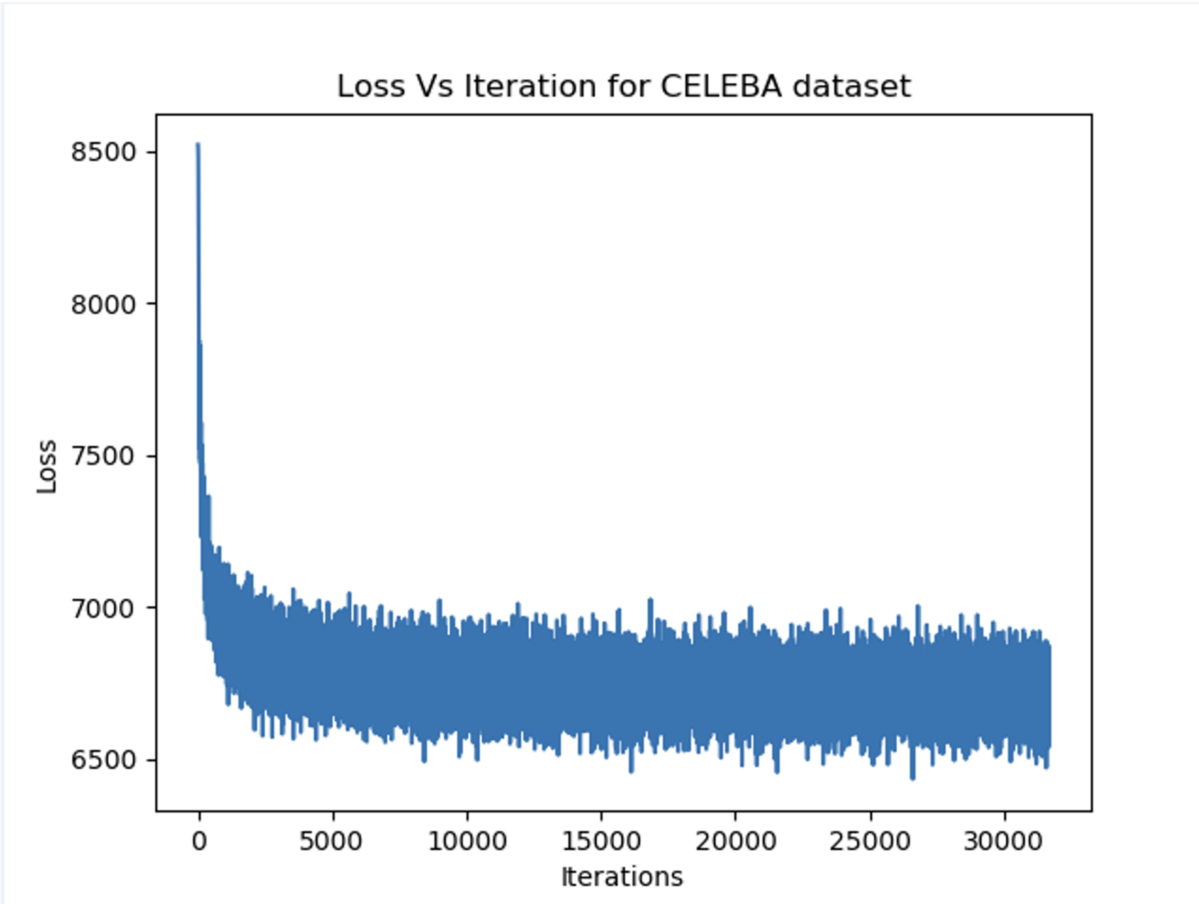


Figure 23: Loss Vs Iteration

1.3.1 Test Images

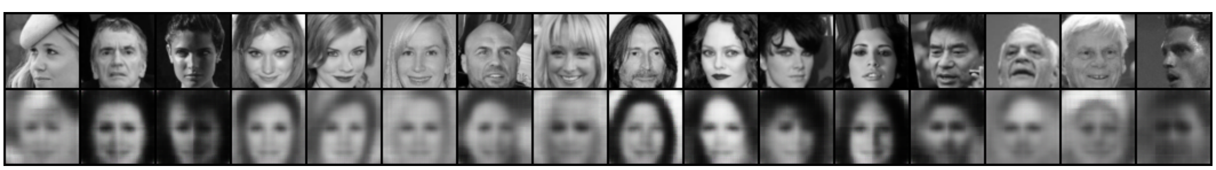


Figure 24: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

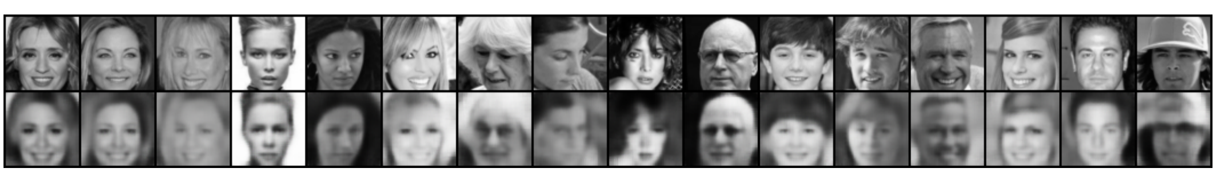


Figure 25: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 26: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 27: Intermediate results(First Row: Actual Images, Second Row: Generated Images)

Figure 28: Final results(First Row: Actual Images, Second Row: Generated Images)

1.3.2 Image generated from Noise

Figure 29: Intermediate results

Figure 30: Intermediate results

Figure 31: Intermediate results

Figure 32: Intermediate results

Figure 33: Intermediate results

Figure 34: Final results

2 Generative Adversarial Networks

2.1 DCGAN on CUFS dataset

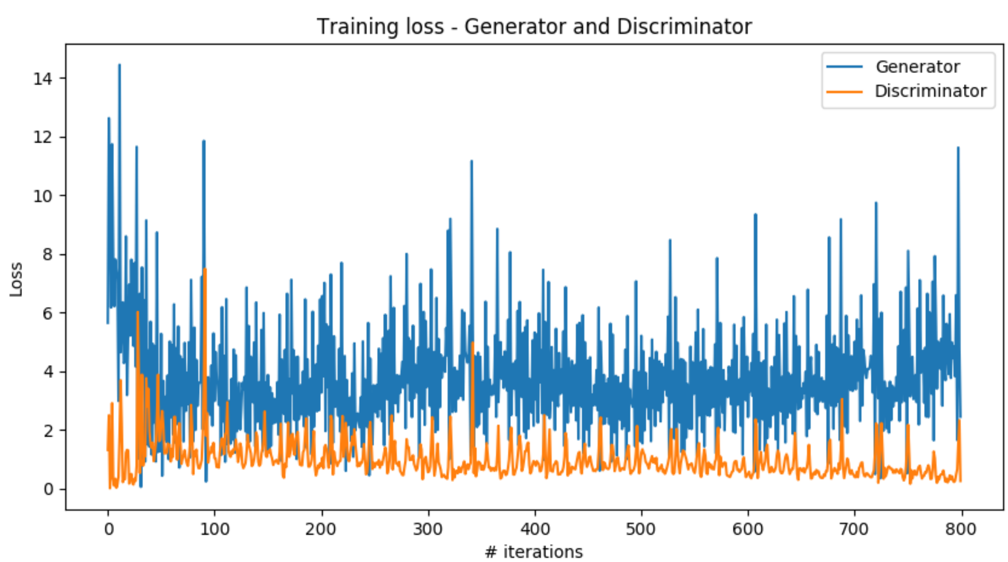


Figure 35: GAN loss

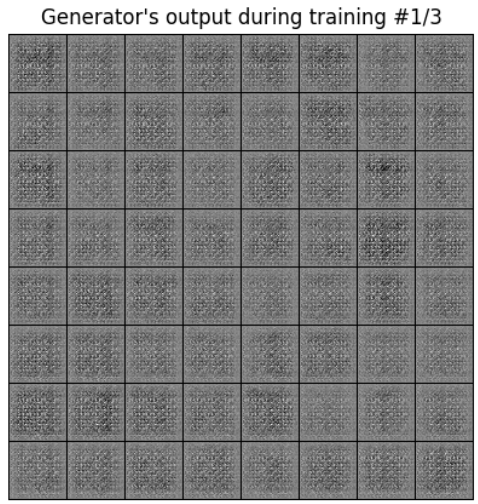


Figure 36: DCGAN on CUFS dataset - Intermediate results

Figure 37: DCGAN on CUFS dataset - Intermediate results

Figure 38: DCGAN on CUFS dataset - Intermediate results

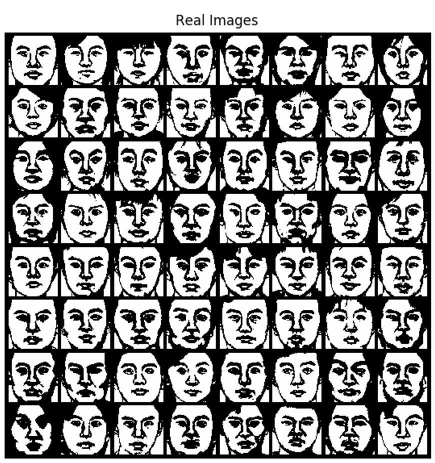


Figure 39: DCGAN on CUFS dataset - Final results

The observation here is that the generated fake images are kind of similar looking (not exactly same though). This might be due to the less number of training examples present for the CUFS dataset, because of which the network is maybe not able to capture all kinds of variability in the images effectively.

2.2 DCGAN on CelebA dataset

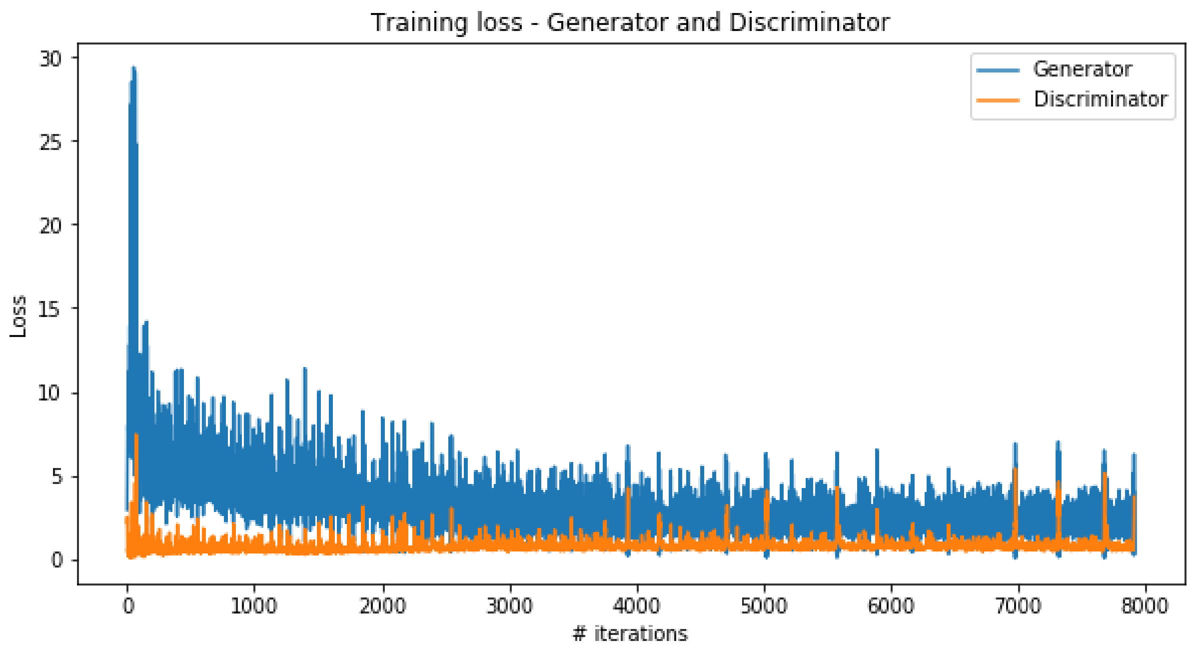


Figure 40: GAN loss

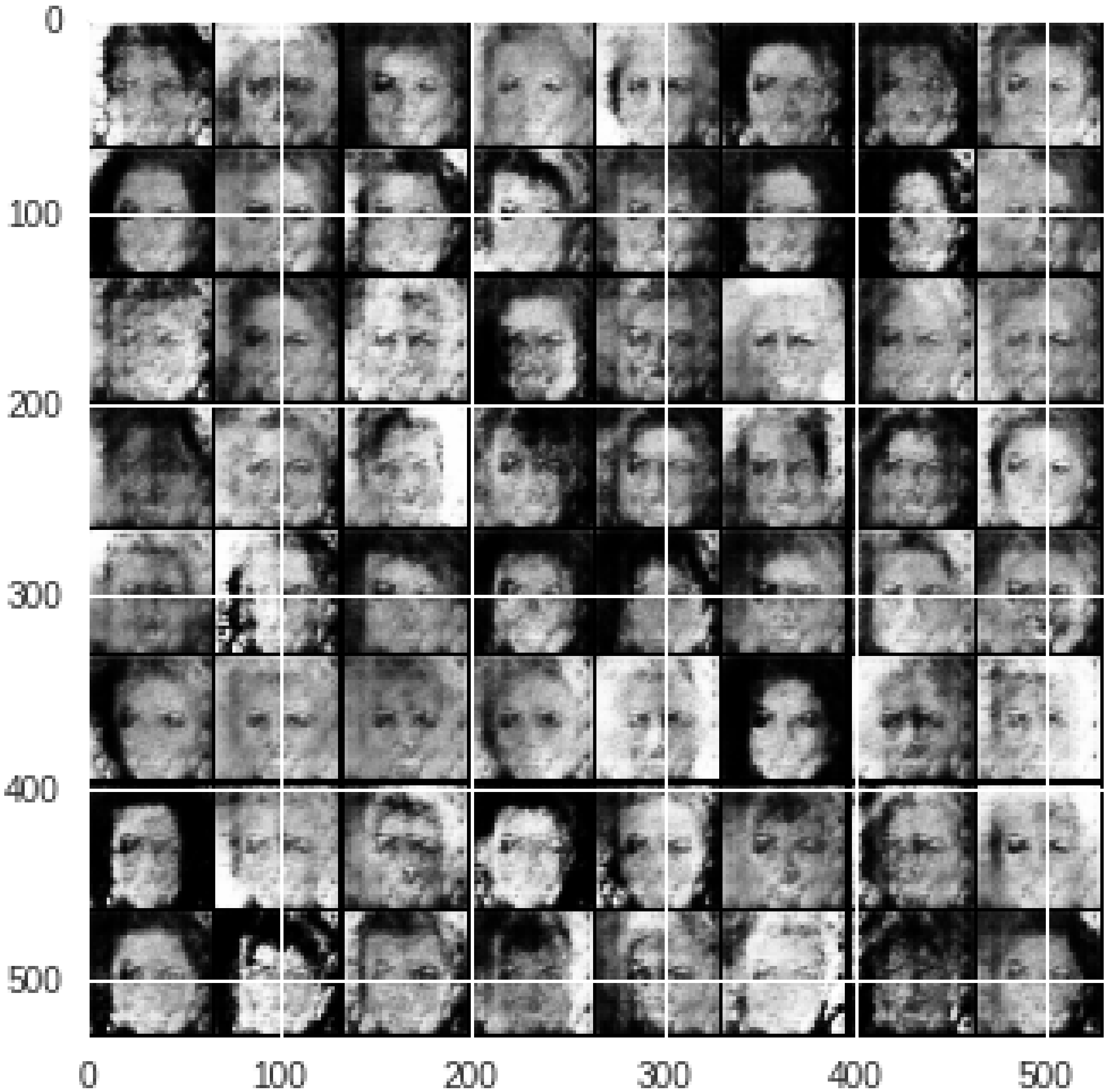


Figure 41: DCGAN on CelebA dataset - Intermediate results

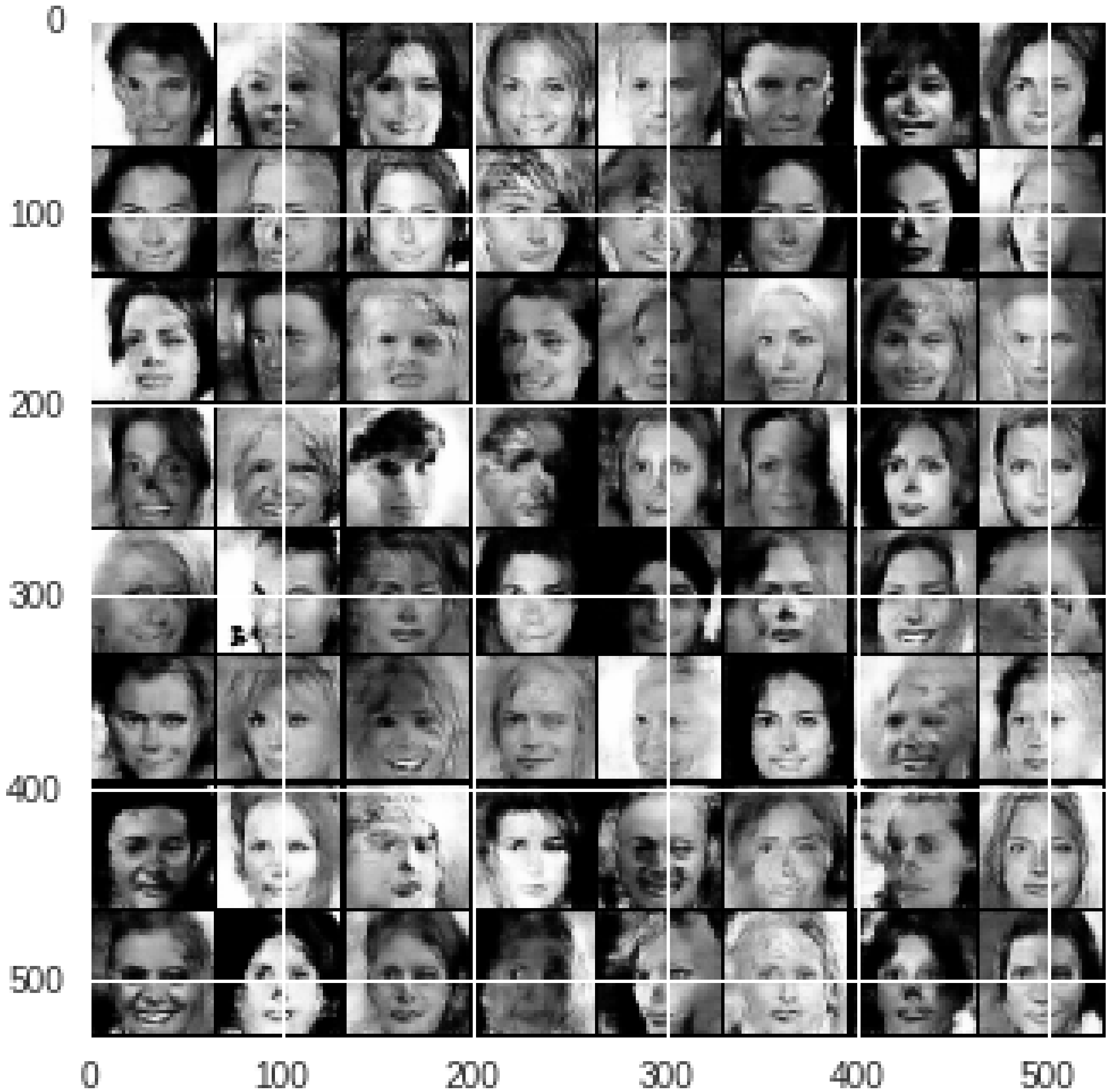


Figure 42: DCGAN on CelebA dataset - Intermediate results

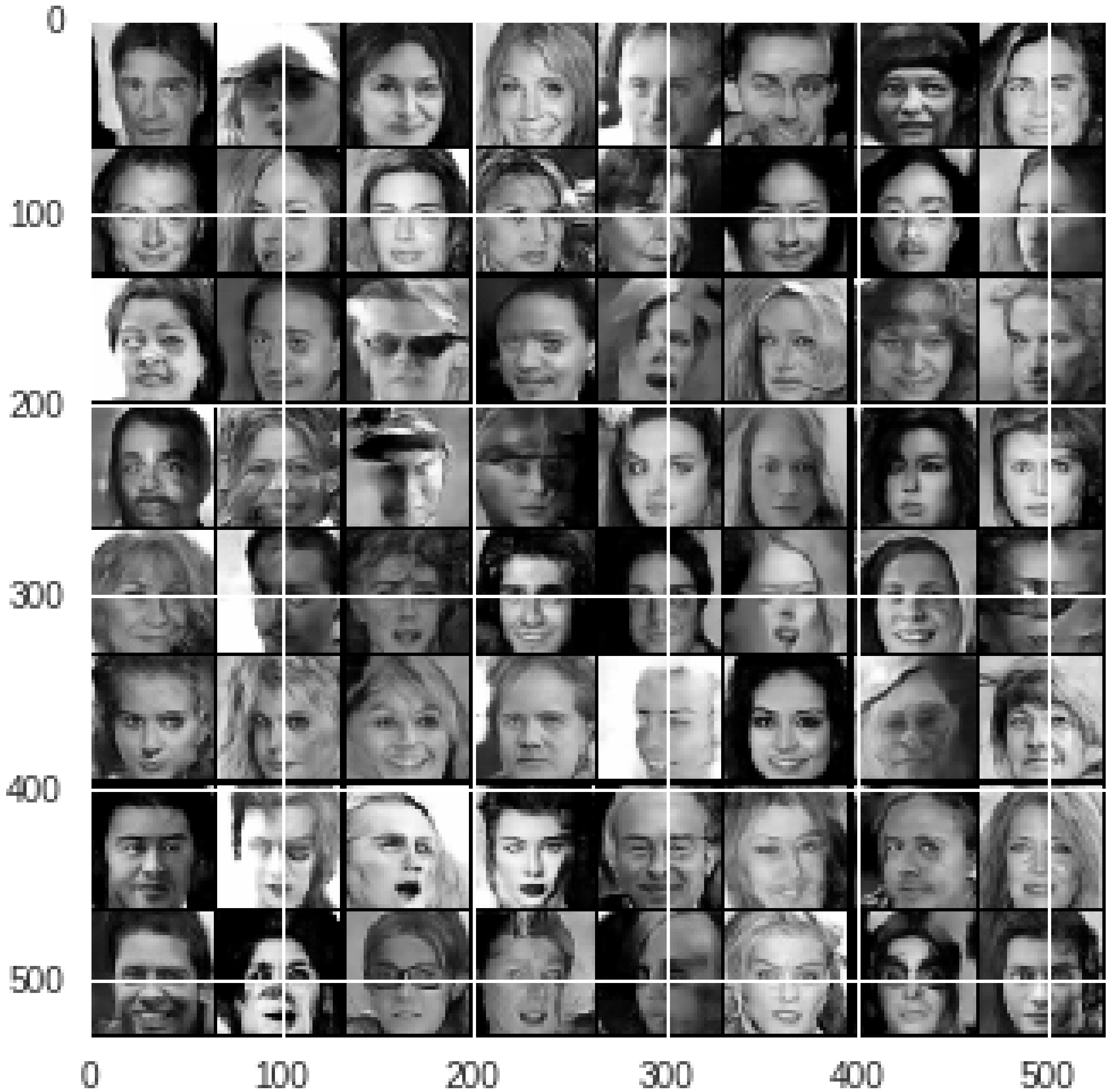


Figure 43: DCGAN on CelebA dataset - Intermediate results

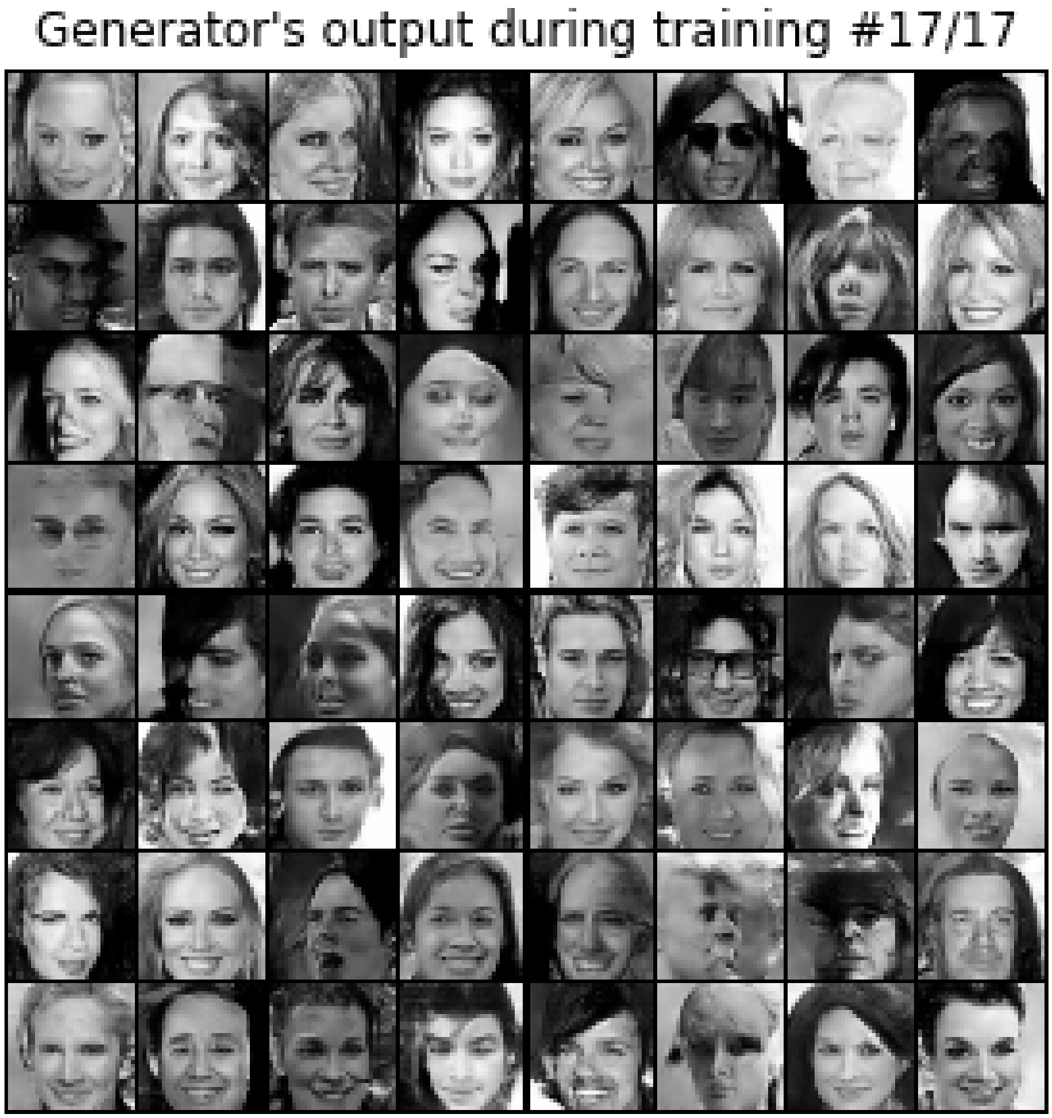


Figure 44: DCGAN on CelebA dataset - Final fake results

The observation here is that the generated fake images are quite different from each other, unlike the first part with CUFS data. This might be because there are a lot of training images in the CelebA dataset and hence the network is able to learn and effectively capture the varying faces.

2.3 WGAN on CUFS dataset

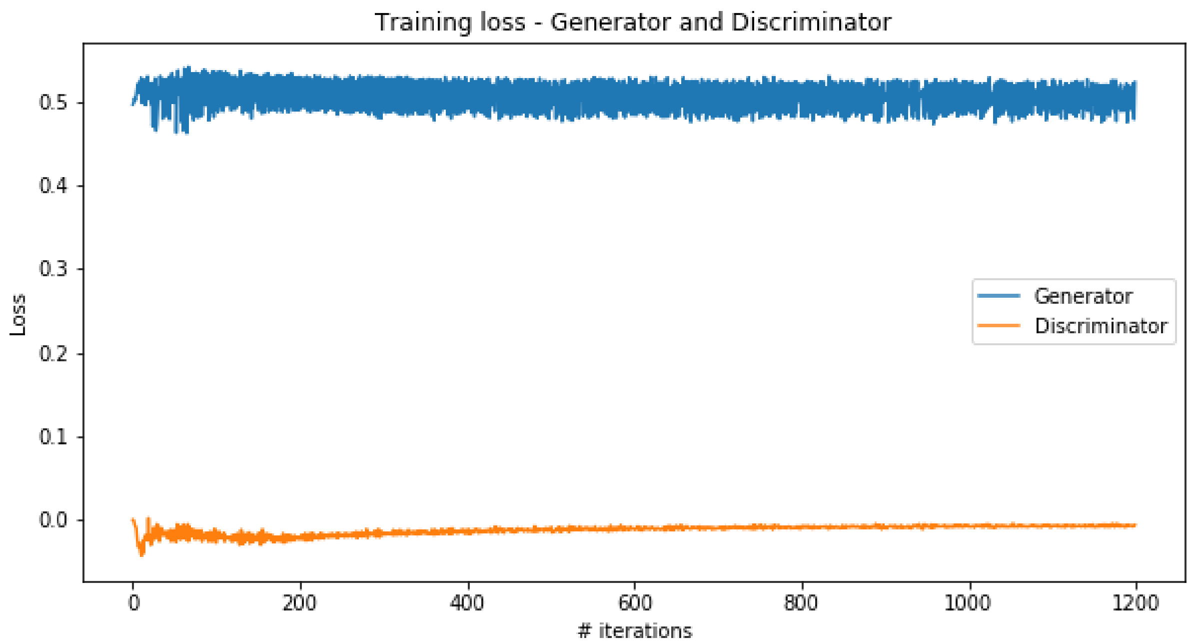


Figure 45: GAN loss

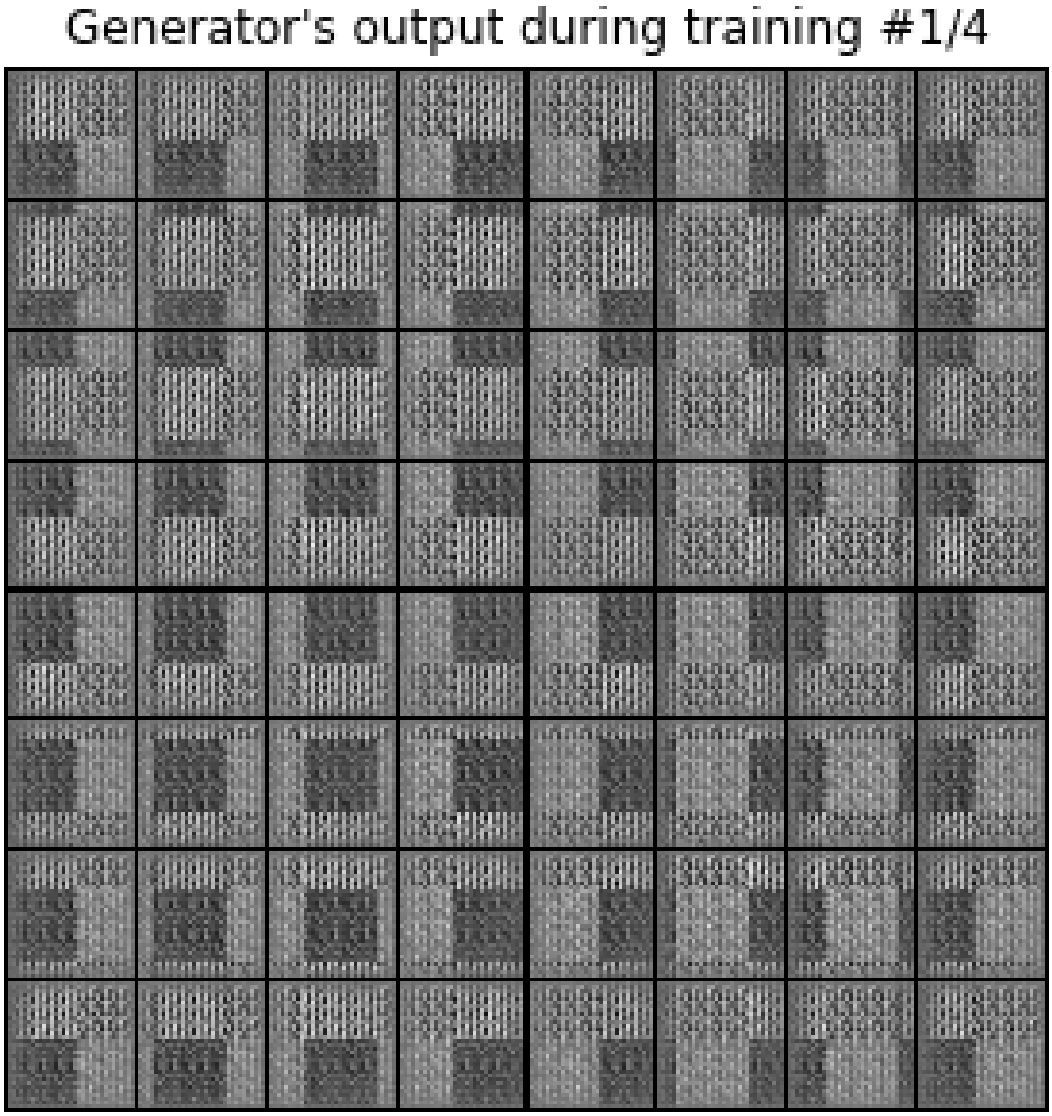


Figure 46: WGAN on CUFS dataset - Intermediate results

Figure 47: WGAN on CUFS dataset - Intermediate results

Figure 48: WGAN on CUFS dataset - Intermediate results

Figure 49: WGAN on CUFS dataset - Final results

2.4 WGAN on CelebA dataset

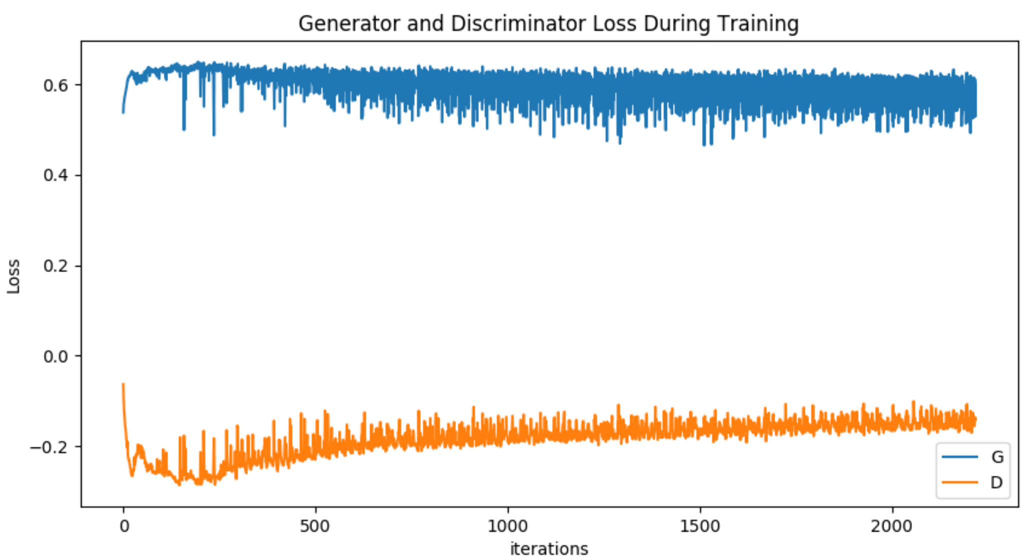


Figure 50: GAN loss

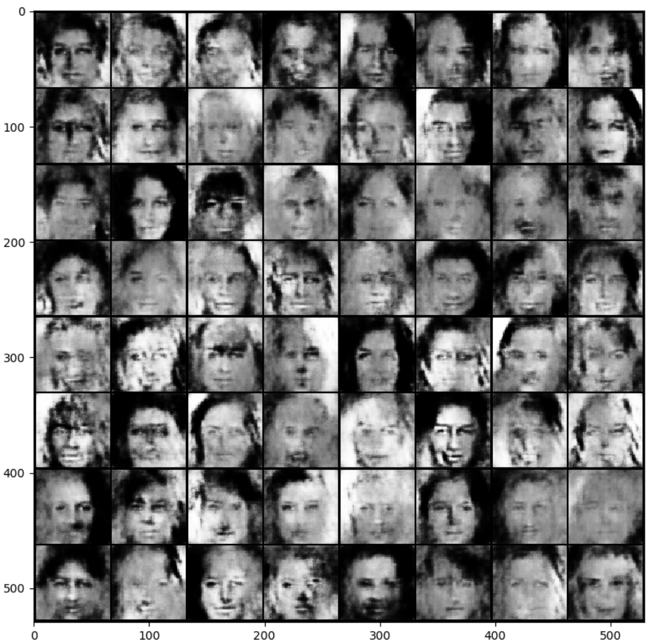


Figure 51: WGAN on CelebA dataset - Intermediate results

Figure 52: WGAN on CelebA dataset - Intermediate results

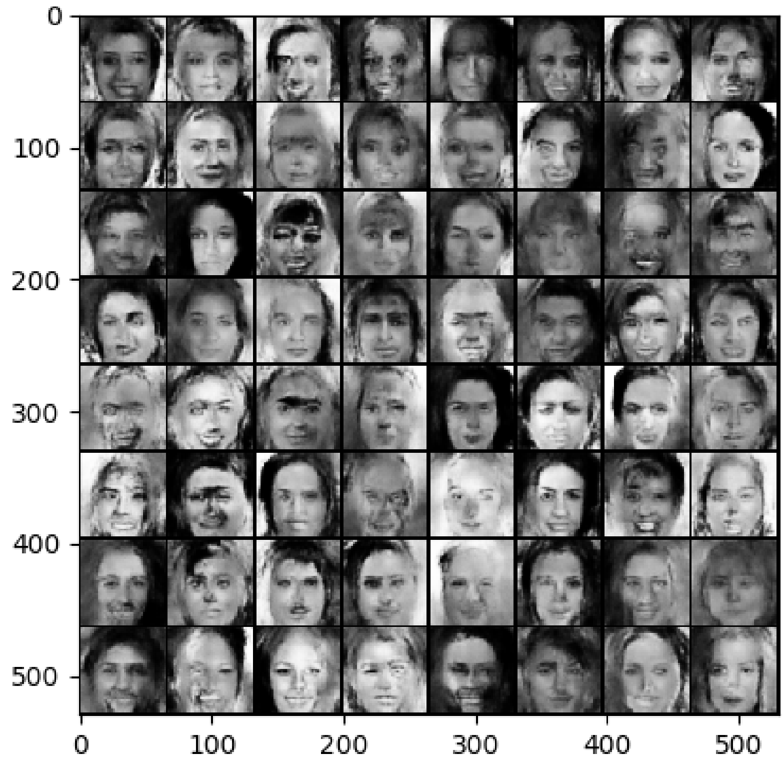


Figure 53: WGAN on CelebA dataset - Intermediate results

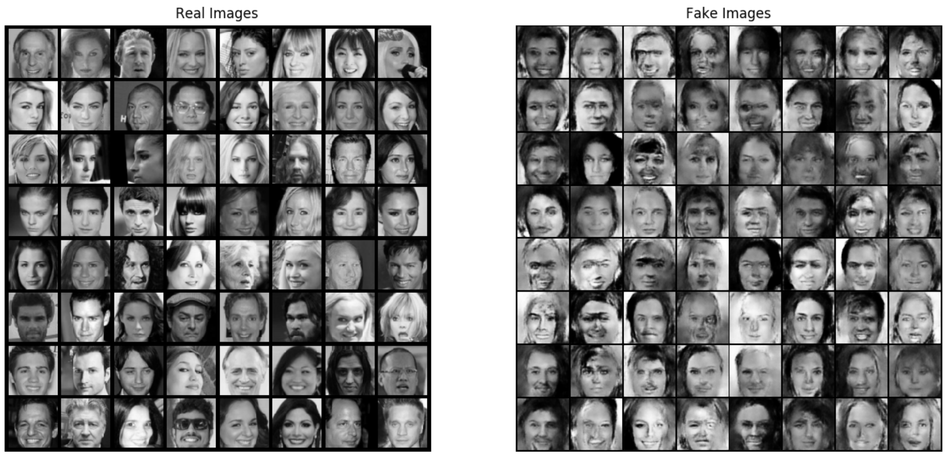


Figure 54: WGAN on CelebA dataset - Final results

3 Image-to-Image Translation

We have used the CUHK pre-processed dataset available on the course website.

The model is implemented based on paper of pix2pix by Isola et. al. In the middle of the generator, a dropout layer has been added to serve as noise. This complements the additional noise that otherwise need to be added with the conditional image.

We used the weighting parameter λ for L_1 loss as 100 based on the recommendation of the paper and it seems to work for the CUHK dataset as well.

Further, we have scaled each image to $256x256$ as implemented in the paper.

We ran our model to only 50 epochs with batch size of 1. We used skip connections for all the parts (apart from the fourth part where it is mentioned).

3.1 Photo to sketch, Generator only

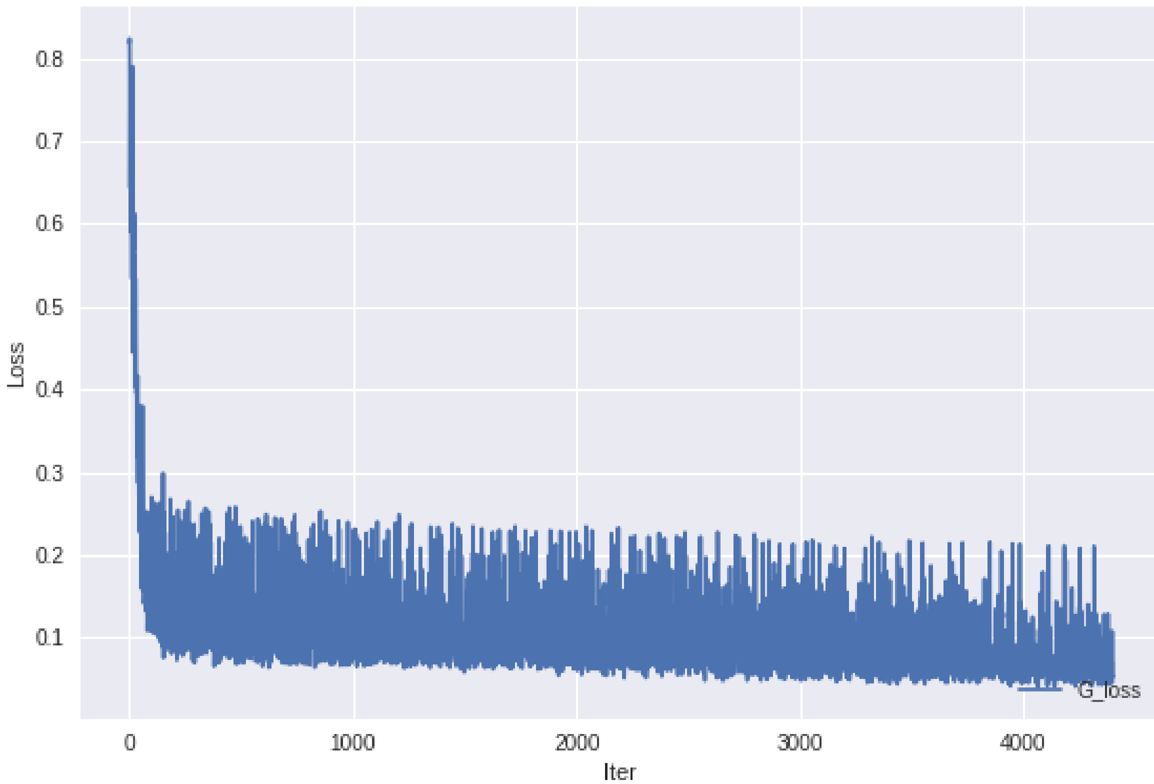


Figure 55: Loss

(a) Input (b) Prediction (c) Ground Truth

Figure 56: Results

(a) Input (b) Prediction (c) Ground Truth

Figure 57: Results

(a) Input (b) Prediction (c) Ground Truth

Figure 58: Results

3.2 Photo to sketch

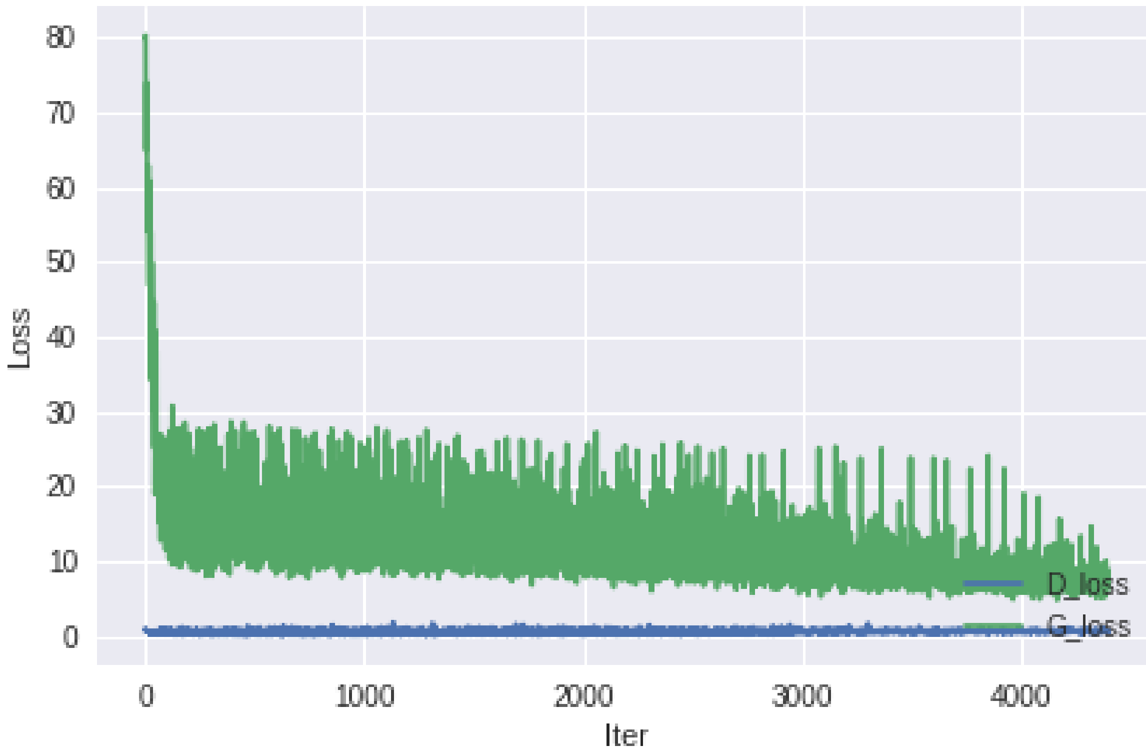


Figure 59: Loss

(a) Input (b) Prediction (c) Ground Truth

Figure 60: Results

(a) Input (b) Prediction (c) Ground Truth

Figure 61: Results

(a) Input (b) Prediction (c) Ground Truth

Figure 62: Results

3.3 Sketch to photo

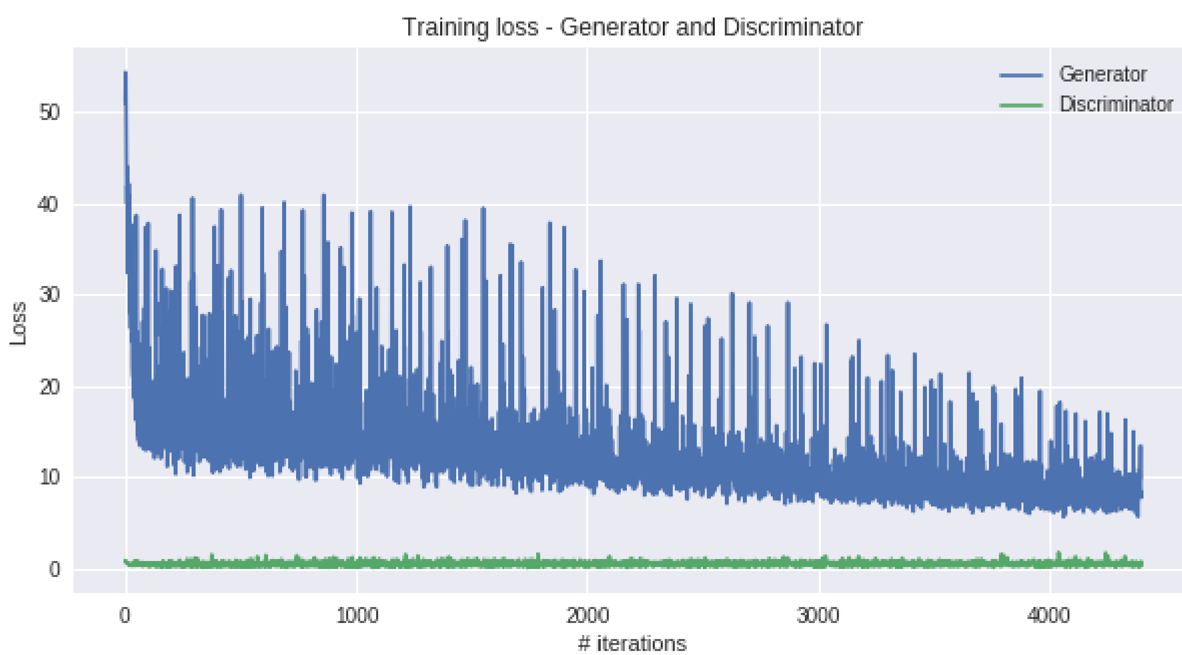


Figure 63: Loss

(a) Input (b) Prediction (c) Ground Truth

Figure 64: Results

(a) Input (b) Prediction (c) Ground Truth

Figure 65: Results

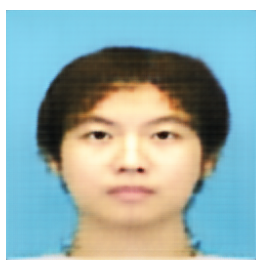
(a) Input (b) Prediction (c) Ground Truth

Figure 66: Results

3.4 Sketch to photo, without skip connections

Figure 67: Loss

(a) Input



(b) Prediction

(c) Ground Truth

Figure 68: Results

(a) Input

(b) Prediction

(c) Ground Truth

Figure 69: Results

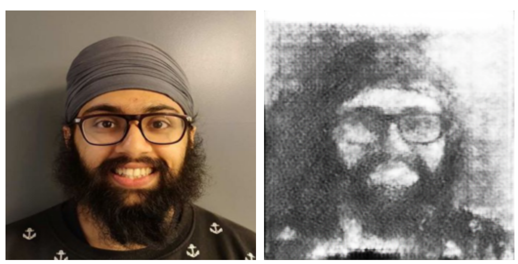
(a) Input (b) Prediction (c) Ground Truth

Figure 70: Results

3.5 Team members photos and sketches

(a) Input (b) Prediction

Figure 71: Results

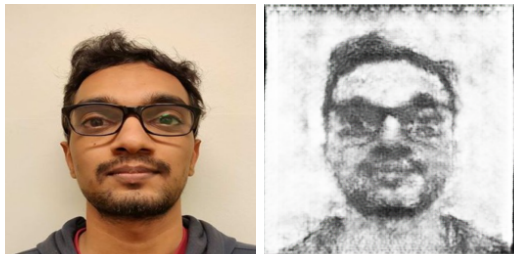


(a) Input (b) Prediction

Figure 72: Results

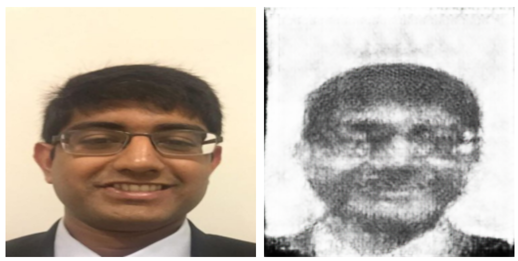
(a) Input (b) Prediction

Figure 73: Results



(a) Input (b) Prediction

Figure 74: Results

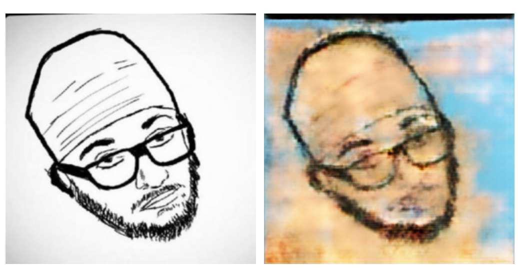


(a) Input (b) Prediction

Figure 75: Results

(a) Input (b) Prediction

Figure 76: Results



(a) Input (b) Prediction

Figure 77: Results

3.6 Comparison of results

3.6.1 Photo to sketch

The results are good for both the cases - Generator loss only and both Discriminator and Generator, though in the latter, it seems like the results are more sharper and slightly more detailed than in the case where we use the Generator only.

3.6.2 Sketch to photo

Here there is a significant difference in picture quality when we use skip connections. Using skip connections leads to more realistic photos (probably due to preserving of information across layers) whereas the results without skip connections tend to be more noisy and less accurate.