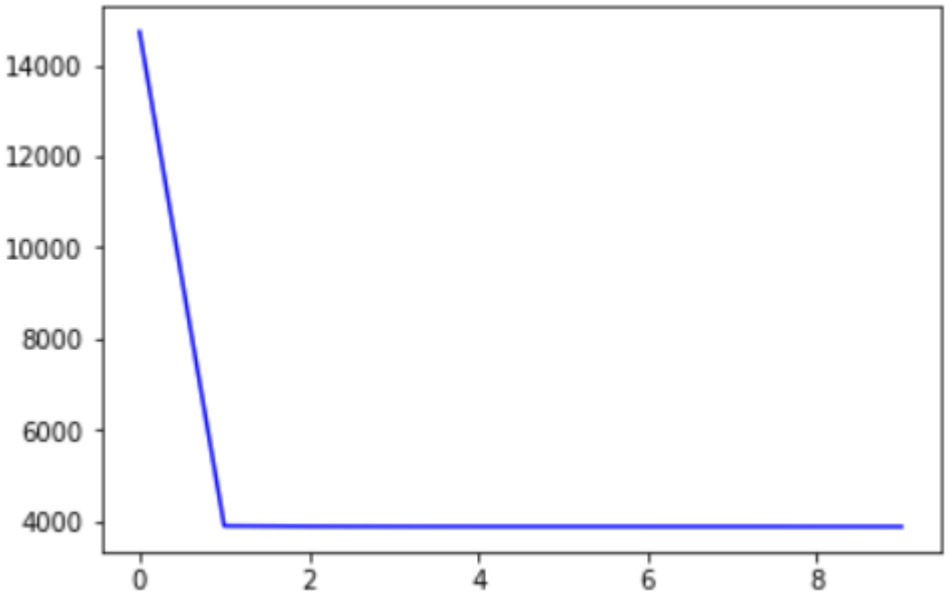
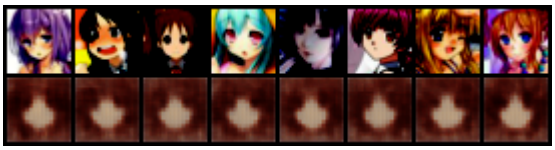
VAE

1. Resize all the images into 32 x 32. Using three conv layers for encoder, and the decoder is as versa. The encoder will output 2 hidden code, such as mean and log variance. The learning curve is as follows.

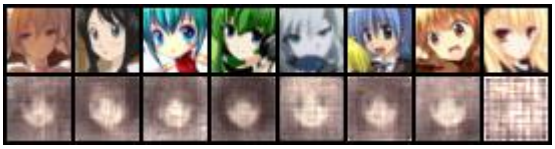


2. Reconstuction

   epoch 1:

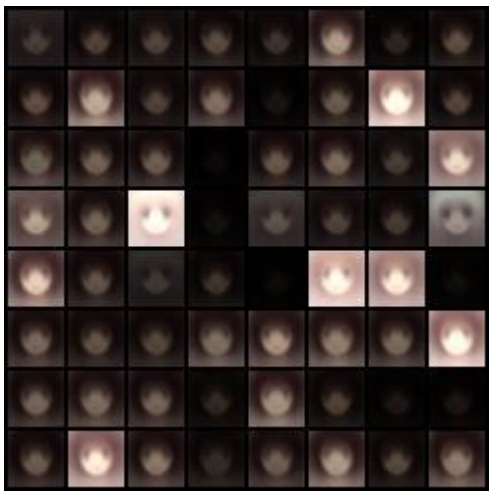


   Epoch 10:

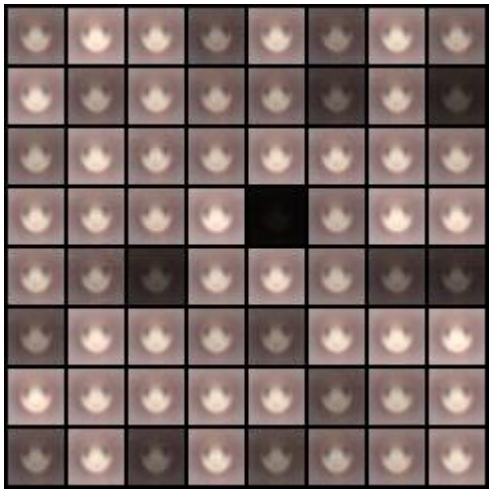


3. Examples

   epoch 1:



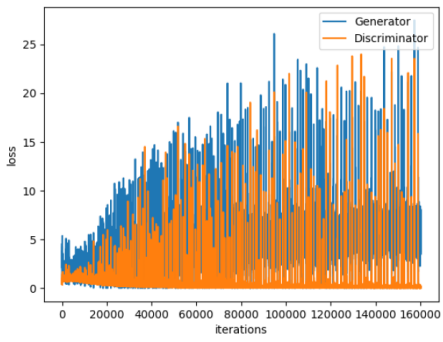
epoch 10:



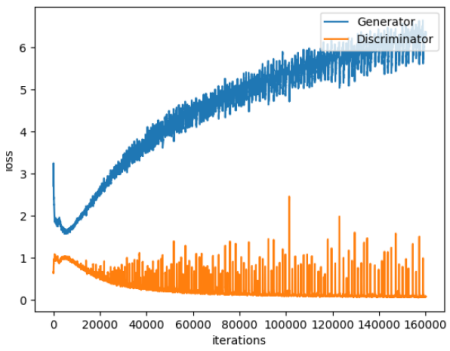
GAN

1. Resize the image to 32 x 32.
2. DCGAN with vanilla GAN objective

Loss:

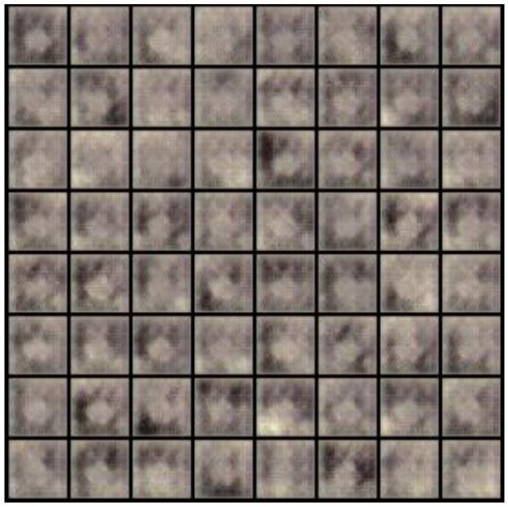


Loss smoothed:



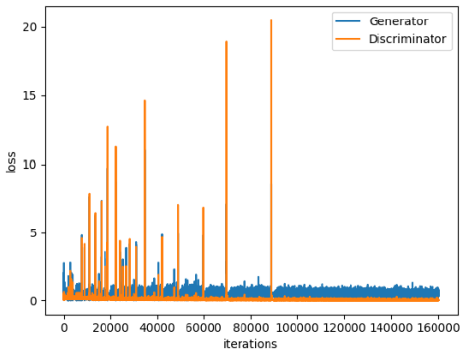
Samples:

Epoch 1 with 100th minibatch

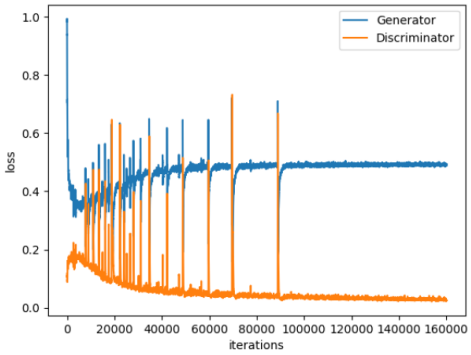


Epoch 100 with 100th minibatch

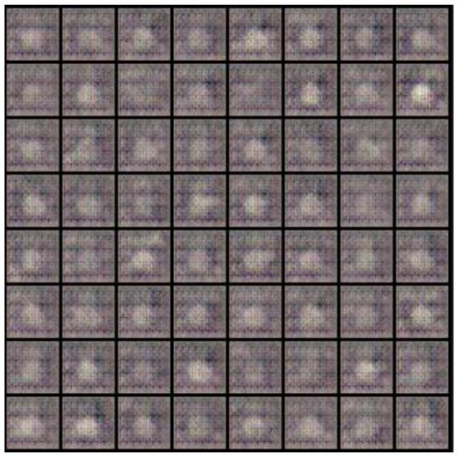
3. DCGAN with LSGAN objective loss:



Smoothed loss:



Epoch 1 with 100th iteration:



Epoch 100 with 100th iteration:

4. The results are very similar, but the training curves are very different. The l2 loss seems to have better training curve.
5. Both architectures are generative model. VAE uses its decoder as generator. The main problem is that the image generated by VAE looks blurred, but the image generated by GAN looks better. Compared to auto-encoder, the decoder of the VAE is stronger than auto-encoder. GAN uses two NN to train adverserially, and get better results.