

Assignment-2

S20170010085

Question-1

Input images have been reduced to 64*64 size to decrease computing time. Training has been done for 200 epochs with batch size of 128. Below are models for generator and discriminator

Generator

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 32768)	3276800
batch_normalization_40 (Batch Normalization)	(None, 32768)	131072
leaky_re_lu_40 (LeakyReLU)	(None, 32768)	0
reshape_4 (Reshape)	(None, 8, 8, 512)	0
conv2d_transpose_20 (Conv2DTranspose)	(None, 16, 16, 256)	3276800
batch_normalization_41 (Batch Normalization)	(None, 16, 16, 256)	1024
leaky_re_lu_41 (LeakyReLU)	(None, 16, 16, 256)	0
conv2d_transpose_21 (Conv2DTranspose)	(None, 32, 32, 128)	819200
batch_normalization_42 (Batch Normalization)	(None, 32, 32, 128)	512
leaky_re_lu_42 (LeakyReLU)	(None, 32, 32, 128)	0
conv2d_transpose_22 (Conv2DTranspose)	(None, 64, 64, 64)	204800
batch_normalization_43 (Batch Normalization)	(None, 64, 64, 64)	256
leaky_re_lu_43 (LeakyReLU)	(None, 64, 64, 64)	0
conv2d_transpose_23 (Conv2DTranspose)	(None, 64, 64, 32)	51200
batch_normalization_44 (Batch Normalization)	(None, 64, 64, 32)	128
leaky_re_lu_44 (LeakyReLU)	(None, 64, 64, 32)	0
conv2d_transpose_24 (Conv2DTranspose)	(None, 64, 64, 3)	2400
Total params: 7,764,192		
Trainable params: 7,697,696		
Non-trainable params: 66,496		

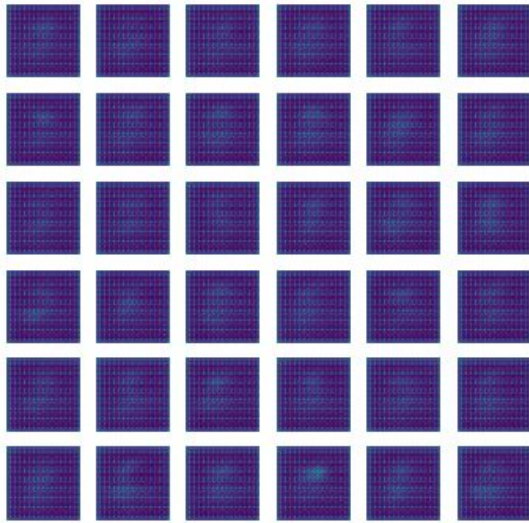
Discriminator

Layer (type)	Output Shape	Param #
TER (Conv2D)	(None, 32, 32, 64)	4864
batch_normalization_36 (Batch Normalization)	(None, 32, 32, 64)	256
leaky_re_lu_36 (LeakyReLU)	(None, 32, 32, 64)	0
dropout_16 (Dropout)	(None, 32, 32, 64)	0
2 (Conv2D)	(None, 16, 16, 128)	204928
batch_normalization_37 (Batch Normalization)	(None, 16, 16, 128)	512
leaky_re_lu_37 (LeakyReLU)	(None, 16, 16, 128)	0
dropout_17 (Dropout)	(None, 16, 16, 128)	0
3 (Conv2D)	(None, 8, 8, 256)	819456
batch_normalization_38 (Batch Normalization)	(None, 8, 8, 256)	1024
leaky_re_lu_38 (LeakyReLU)	(None, 8, 8, 256)	0
dropout_18 (Dropout)	(None, 8, 8, 256)	0
4 (Conv2D)	(None, 4, 4, 512)	3277312
batch_normalization_39 (Batch Normalization)	(None, 4, 4, 512)	2048
leaky_re_lu_39 (LeakyReLU)	(None, 4, 4, 512)	0
dropout_19 (Dropout)	(None, 4, 4, 512)	0
flatten_4 (Flatten)	(None, 8192)	0
dense_8 (Dense)	(None, 1)	8193
Total params: 4,318,593		
Trainable params: 4,316,673		
Non-trainable params: 1,920		

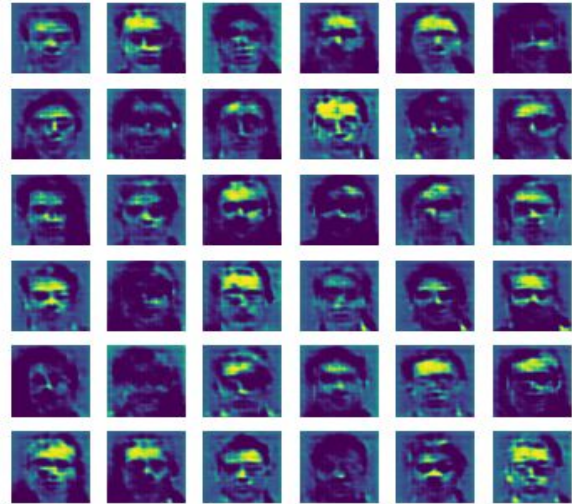
For generator noise of 100*1 is randomly generated and fed through network.

Results

Epoch 0:



Epoch 50:



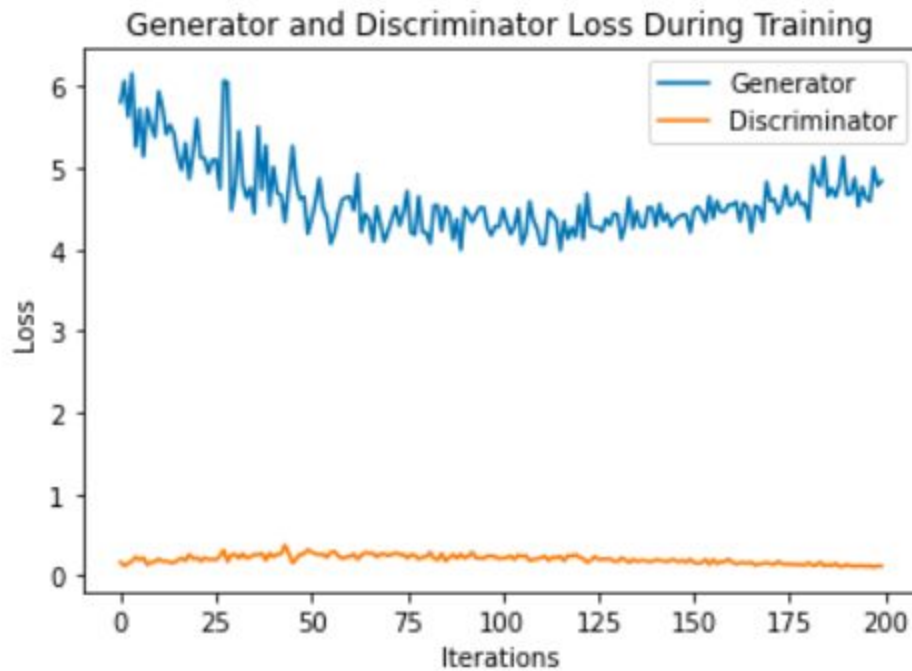
Epoch 100:



Epoch 200:



LOSS:



Question-2

Results

Results after running autoencoder for 65 epochs on the dataset resized to 64*64 size :



Upper row are original and the lower row are generated images. Both are of same resolution but generated images are blurry and we can see in many cases person's spectacles are not generated.

Model's architecture:

Encoder

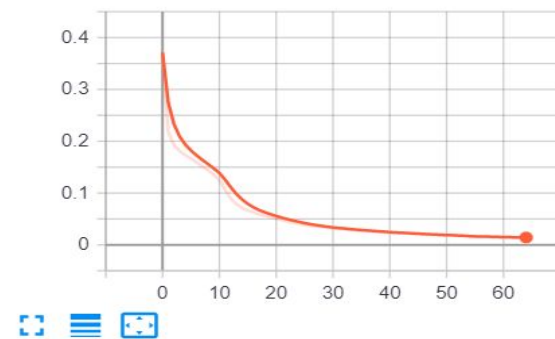
Layer (type)	Output Shape	Param #
1 (Conv2D)	(None, 32, 32, 64)	4864
batch_normalization_9 (Batch Normalization)	(None, 32, 32, 64)	256
leaky_re_lu_9 (LeakyReLU)	(None, 32, 32, 64)	0
dropout_4 (Dropout)	(None, 32, 32, 64)	0
2 (Conv2D)	(None, 16, 16, 128)	204928
batch_normalization_10 (Batch Normalization)	(None, 16, 16, 128)	512
leaky_re_lu_10 (LeakyReLU)	(None, 16, 16, 128)	0
dropout_5 (Dropout)	(None, 16, 16, 128)	0
3 (Conv2D)	(None, 8, 8, 256)	819456
batch_normalization_11 (Batch Normalization)	(None, 8, 8, 256)	1024
leaky_re_lu_11 (LeakyReLU)	(None, 8, 8, 256)	0
dropout_6 (Dropout)	(None, 8, 8, 256)	0
4 (Conv2D)	(None, 4, 4, 512)	3277312
batch_normalization_12 (Batch Normalization)	(None, 4, 4, 512)	2048
leaky_re_lu_12 (LeakyReLU)	(None, 4, 4, 512)	0
dropout_7 (Dropout)	(None, 4, 4, 512)	0
flatten_1 (Flatten)	(None, 8192)	0
dense_2 (Dense)	(None, 100)	819300
Total params: 5,129,700		
Trainable params: 5,127,780		
Non-trainable params: 1,920		

Decoder

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 32768)	3276800
batch_normalization_13 (Batch Normalization)	(None, 32768)	131072
leaky_re_lu_13 (LeakyReLU)	(None, 32768)	0
reshape_1 (Reshape)	(None, 8, 8, 512)	0
conv2d_transpose_5 (Conv2DTranspose)	(None, 16, 16, 256)	3276800
batch_normalization_14 (Batch Normalization)	(None, 16, 16, 256)	1024
leaky_re_lu_14 (LeakyReLU)	(None, 16, 16, 256)	0
conv2d_transpose_6 (Conv2DTranspose)	(None, 32, 32, 128)	819200
batch_normalization_15 (Batch Normalization)	(None, 32, 32, 128)	512
leaky_re_lu_15 (LeakyReLU)	(None, 32, 32, 128)	0
conv2d_transpose_7 (Conv2DTranspose)	(None, 64, 64, 64)	204800
batch_normalization_16 (Batch Normalization)	(None, 64, 64, 64)	256
leaky_re_lu_16 (LeakyReLU)	(None, 64, 64, 64)	0
conv2d_transpose_8 (Conv2DTranspose)	(None, 64, 64, 32)	51200
batch_normalization_17 (Batch Normalization)	(None, 64, 64, 32)	128
leaky_re_lu_17 (LeakyReLU)	(None, 64, 64, 32)	0
conv2d_transpose_9 (Conv2DTranspose)	(None, 64, 64, 3)	2400
Total params: 7,764,192		
Trainable params: 7,697,696		
Non-trainable params: 66,496		

Loss :

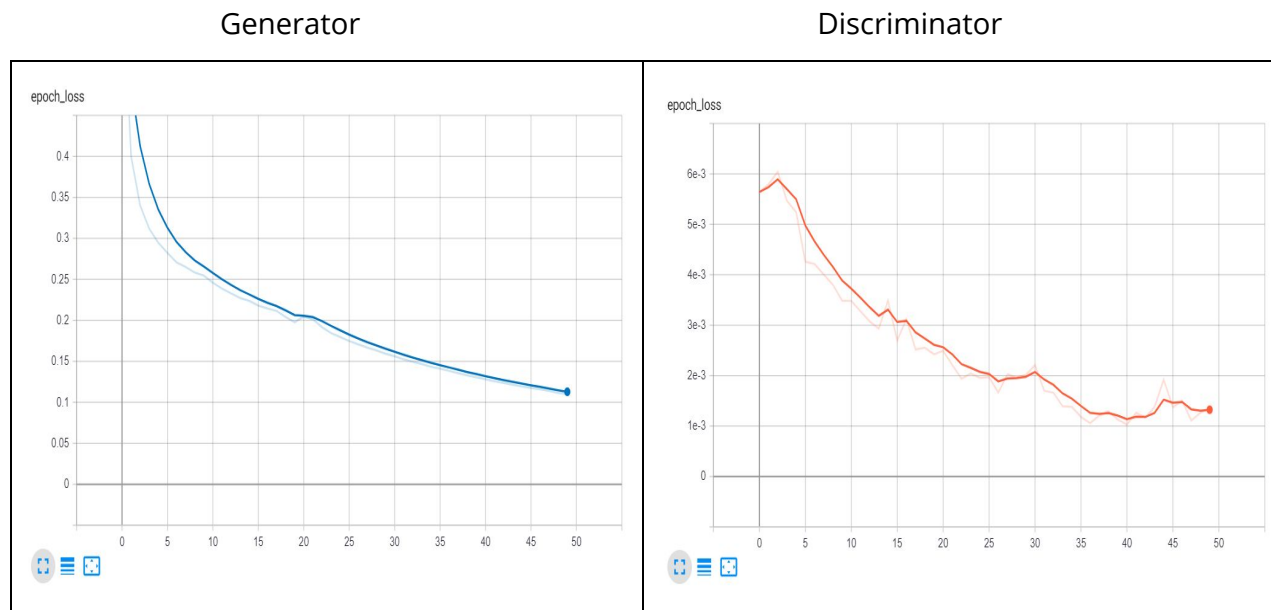
epoch_loss



Question - 3

Here, Above autoencoder is used as generator for GAN. Same architecture is used as above. Here we no longer have to generate random vector because autoencoder generates images and we have to compute the loss against original images and improve the model.

Results:



Output:

