ECE 285 - Deep Generative Models - Assignment 4

Saqib Azim (A59010162)

sazim@ucsd.edu

ASSIGNMENT REPOSITORY: https://github.com/saqib1707/ECE285_Assignments/tree/master/assignment4

1 Generative Adversarial Networks (GAN) Introduction

GANs belong to the set of deep generative models which can be used for teaching a deep learning network to learn the training data distribution in order to generate new data from that same distribution. With huge training data, sophisticated model architectures and optimization algorithms, GANs can generate highly realistic content of various kinds of signals such as images, text, speech, etc. GAN consists of two separate networks - a generator G and a discriminator D. The objective of the generator is to generate fake synthetic images similar to real training images whereas the task of the discriminator is to learn to correctly classify between real and fake images. During training, the generator tries to improve in generating better fake images while the discriminator, working opposite to improve itself to become a better classifier. Theoretically, the training process reaches equilibrium when the generator is generating perfect fake synthetic images that can be perceived as real data, while the discriminator outputs 0.5 probabilities for the generated image being real and fake.

Let x denote a real training image. $D(x)$ represents the discriminator network producing a scalar probability that the image x came from training data distribution rather than the generator fake output. Ideally, $D(x)$ should be high (close to 1) when the input x comes from training data and low (close to 0) when it comes from the generator. Let z denote the latent space tensor sampled from standard normal distribution $\mathcal{N}(0, 1)$, generator function $G(z)$ maps this latent vector z to image space MxMx3 in order to learn the training data distribution p_d so that later it can generate fake sample images from the estimated distribution p_g.

The generator G and the discriminator D play a minimax game where the discriminator tries to maximize the probability of classifying between real and fake images, whereas the generator tries to minimize the probability that the discriminator will classify its generated fake images as fake.

$D(G(z))$: probability that the output $G(z)$ of the generator G is a real image.

2 Network Architecture

Deep Convolutional GAN (aka DC-GAN) model has been used to train the chest x-ray data in order to learn to generate synthetic fake images. The DC-GAN architecture consists of a generator and a discriminator network. The generator takes a random vector z of shape D_Lx1x1 sampled from a latent space $\mathcal{N}(0, 1)$, and passes it through a series of transposed convolution layer $\longrightarrow$ *BatchNorm* layer $\longrightarrow$ *ReLU* layer. Figure 1 illustrates the DC-GAN generator network architecture. The final layer consists of transposed convolution layer $\longrightarrow$ *tanh* activation function instead of *BatchNorm* and *ReLU* in order to project the generated tensor image values between [-1,1]. For each latent space input vector z, the output of the generator is a fake image of size MxMx3 where M is a hyperparameter.

Preprint. Under review.

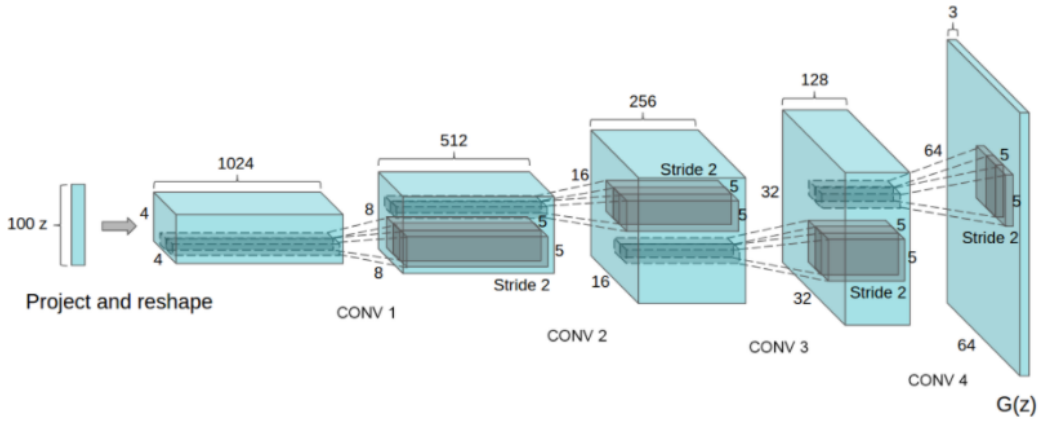

Figure 1: Generator Network Architecture

The discriminator D, as shown in Figure 2, is a binary classification network that takes an input image of size MxMx3 and outputs a scalar probability of the confidence of the input image being real. It consists of a series of convolution layer $\longrightarrow$ *BatchNorm* layer $\longrightarrow$ *Leaky ReLU* layer. The final layer replaces the *BatchNorm* and *Leaky ReLU* with *sigmoid* layer in order to generate probability values between [0,1]. The usage of *BatchNorm* and *Leaky ReLU* facilitates the gradient flow during GAN training.

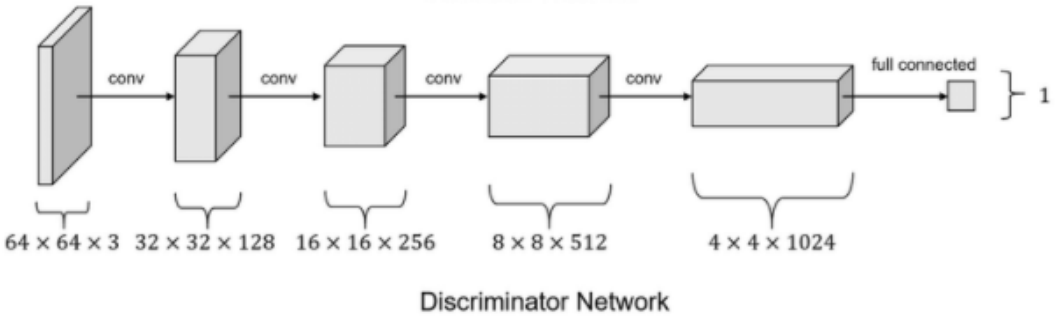

Figure 2: Discriminator Network Architecture

3 Dataset Information

Name: Chest X-Ray Dataset
Download Path: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

Number of Train Images (NORMAL + PNEUMONIA): 5216
Number of Validation Images (NORMAL + PNEUMONIA): 16
Number of Test Images (NORMAL + PNEUMONIA): 624

The goal of this assignment is to learn to generate new synthetic chest x-ray images from random input vectors sampled from the standard normal distribution space using GAN models (specifically DC-GAN). Since, we are required to only generate synthetic fake images, there is no need for classification labels. Therefore, all the images belonging to NORMAL and PNEUMONIA class labels have been merged together. The raw chest x-ray images are of varying sizes / resolution, hence all the images are resized to a fixed size (M x M) before feeding to the GAN network.

4 Training Procedure

We have used the binary cross-entropy loss during training defined as -

$$l_n = -[y_n . \log x_n + (1 - y_n) . \log(1 - x_n)] \tag{1}$$

Adam optimizer has been used with a varying learning rate α in different experiments. In each training iteration, we construct different mini-batches of real (from training data) and fake images (generator G output), and update the discriminator network followed by updating the generator network.

For the discriminator, the goal is to maximize $\log(D(x)) + \log(1 - D(G(z)))$. First, we construct a batch of real training images, forward pass through the discriminator, calculate the loss $\log D(x)$, and then calculate the gradients in backward pass. Secondly, we construct a batch of fake images generated using generator from random samples, forward pass through discriminator, compute the loss $\log(1 - D(G(z)))$, and compute the gradients using backward pass. Now, we update the weights parameters of the discriminator network using real-image gradients and fake-image gradients.

For the generator, the objective is to generate better fake images by minimizing $log(1 - D(G(z)))$. First, we classify the batch of fake images generated using generator, calculate the loss $\log D(G(z))$, compute the gradients with backward pass, and finally update the weights of generator network.

$$\min \max L(D, G) = \mathtt{E}_{x \sim p_d(x)}[\log D(x)] + \mathtt{E}_{z \sim p(z)}[\log(1 - D(G(z)))] \tag{2}$$

5 Experiments

In these experiments, the following hyperparameters and their effect on GAN training and performance has been observed using metrics such as inception score (IS) and frechet inception distance (FID).

- input image size (M x M)
- latent space channels L_{ch}
- latent space spatial size L_wxL_w
- network feature map depth factor N_{gf} and N_{df}
- Number of epochs N_E
- learning rate α

Fixed hyperparameters: Following hyperparameters have not been modified during these experiments -

- Number of input image channels = 3
- Training batch size = 64
- Leaky ReLU weight = 0.2
- Adam optimizer parameter β = 0.5

Inception Score (IS): This metric is used for evaluating the quality of generated images by generative models (such as GAN, VAE, etc.). It uses the classification probabilities provided by a pre-trained InceptionV3 model to measure performance of the GAN. It passes each generated sample image through an InceptionV3 classification network and generates classification probability distribution. If this distribution is narrow, then this implies that the image has a distinct perceivable class. Similarly, it also computes marginal score for a set of 1000 predefined classes using all the images. If this marginal distribution is uniform, then it means the GAN is capable of generating variety of images belonging to different classes. Using these concepts, the inception score is computed. **Higher the inception score, better the generative capability of a generative network**.

Frechet Inception Distance (FID): This metric calculates the distance between feature

vectors extracted for real and synthetic generated images. Similar to IS, it uses a pre-trained InceptionV3 model, and computes the mean and co-variance between the feature vectors of real and generated images to quantify and measure GAN performance. Since the FID score measures distance between the latent gaussian distribution and the standard normal distribution, lower the FID score, the better the performance of the generative model.

The chest x-ray dataset and the generated images are both MxMx3 (where M = 64, 128, 256), but the InceptionV3 model (used in FID and IS computation) requires images of minimum size 299x299x3. Hence, the dataset test images and VAE generated sample images have been interpolated to the specified required size while calculating these metrics. In addition, the IS and FID scores are reported using samples generated after the training is complete.

	Exp-1 5.1	Exp-2 5.2	Exp-3 5.3	Exp-4 5.4	Exp-5 5.5
M	64	64	64	64	128
L_{ch}xL_wxL_w	100x1x1	200x1x1	100x1x1	100x1x1	100x5x5
N_{gf}, N_{df}	64, 64	64, 64	64, 64	64, 64	128, 32
N_E	50	50	50	50	20
α	$1e^{-4}$	$1e^{-4}$	$2e^{-4}$	$10e^{-4}$	$2e^{-4}$
IS	**2.0149**	1.7195	1.5632	1.8693	1.0088
FID	1.0403	**0.7332**	0.4905	1.4619	248.79

	Exp-6 5.6
M	128
L_{ch}xL_wxL_w	100x1x1
N_{gf}, N_{df}	64, 16
N_E	20
α	$2e^{-4}$
IS	1.0010
FID	148.63

Table 1: Hyperparameter settings and Metrics Score for different experiments performed

- M represents the image size
- L_{ch}xL_wxL_w represents the latent space tensor of width and height L_w and number of channels L_{ch} sampled from $\mathcal{N}(0, 1)$
- N_{gf} & N_{df} represents the factor controlling the depth of feature maps inside generator and discriminator network resp.
- N_E is the number of training epochs
- α: is the learning rate during training

5.1 Experiment-1

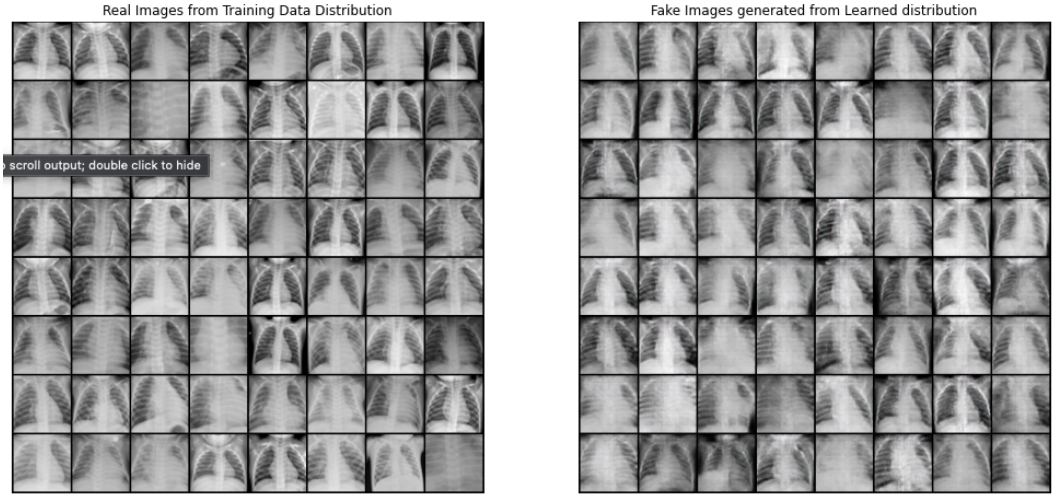

Figure 3: Real Training Images (left) and Fake Images generated by the network (right)

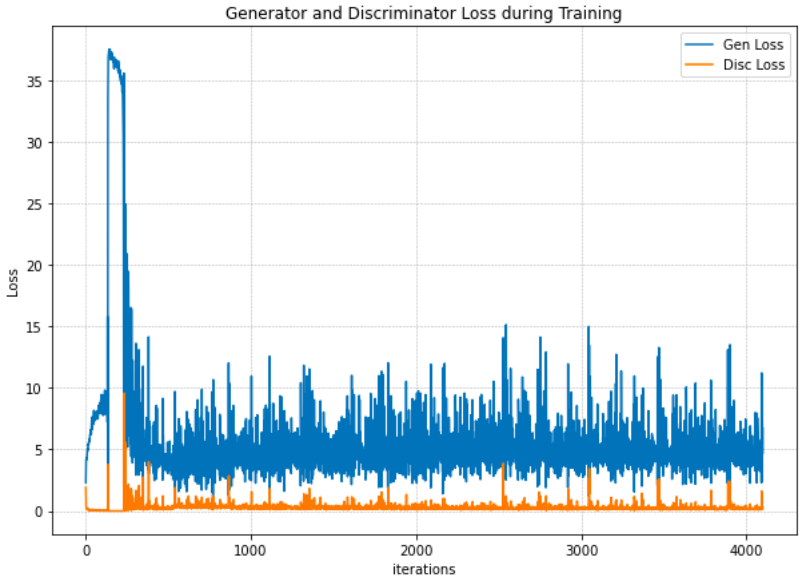

Figure 4: Training Loss with Iterations

5.2 Experiment-2

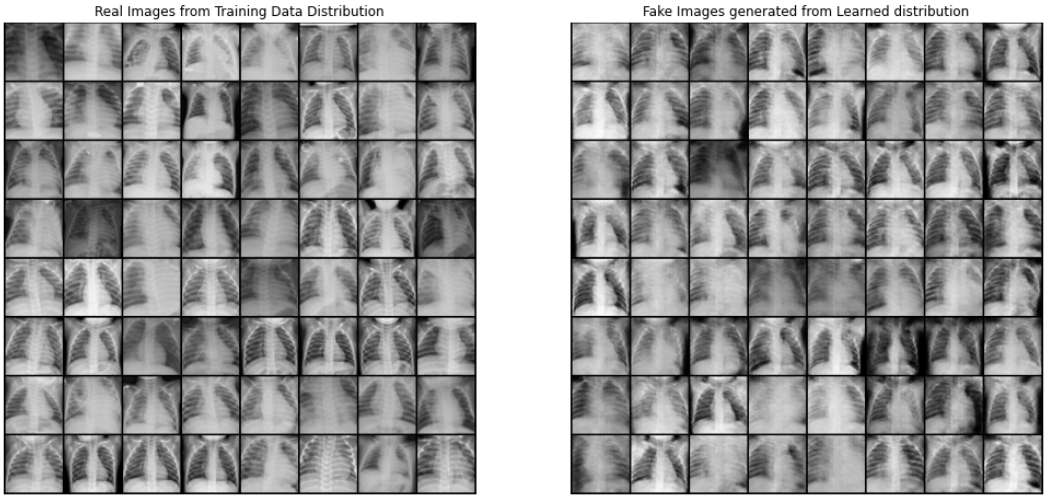

Figure 5: Real Training Images (left) and Fake Images generated by the network (right)

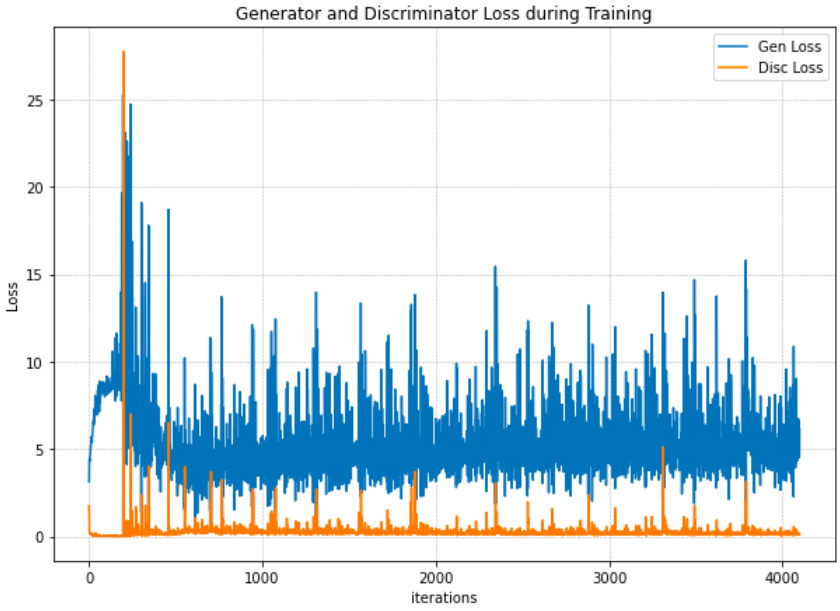

Figure 6: Training Loss with Iterations

5.3 Experiment-3

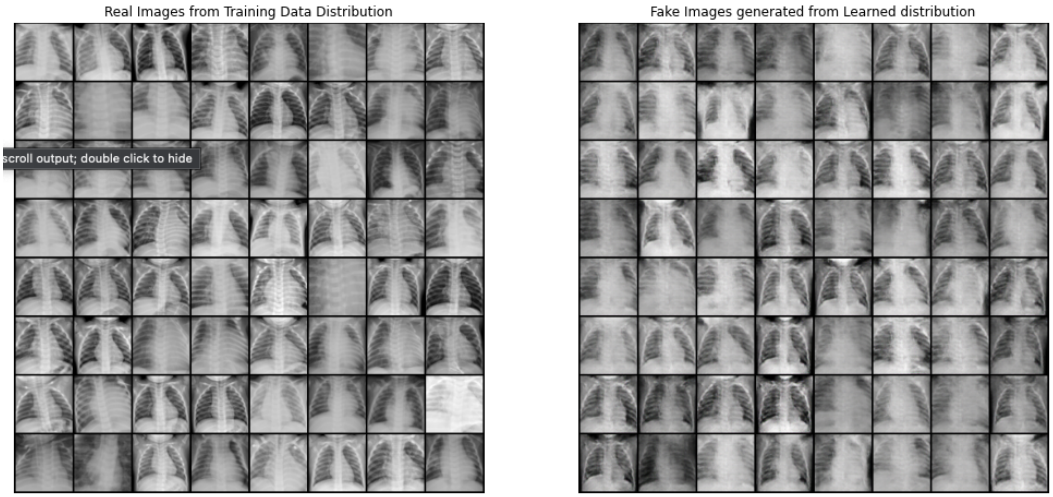

Figure 7: Real Training Images (left) and Fake Images generated by the network (right)

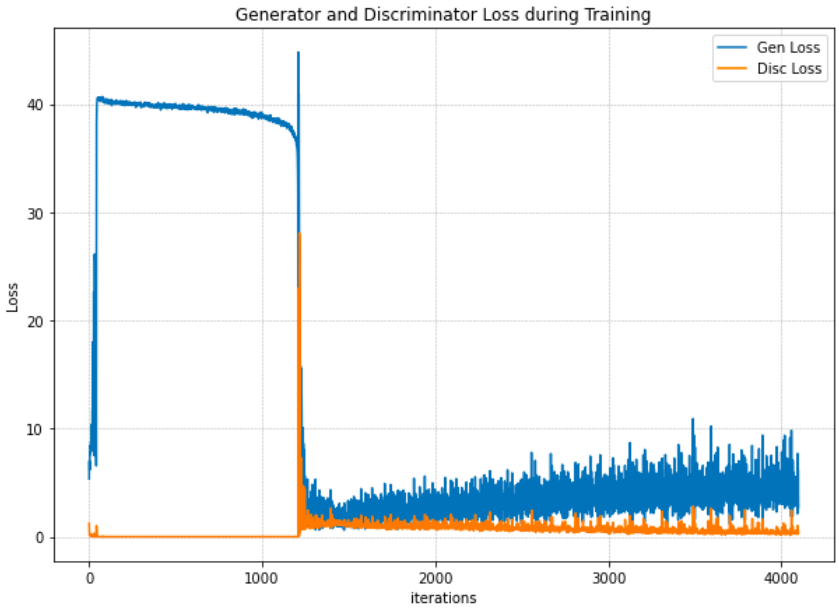

Figure 8: Training Loss with Iterations

5.4 Experiment-4

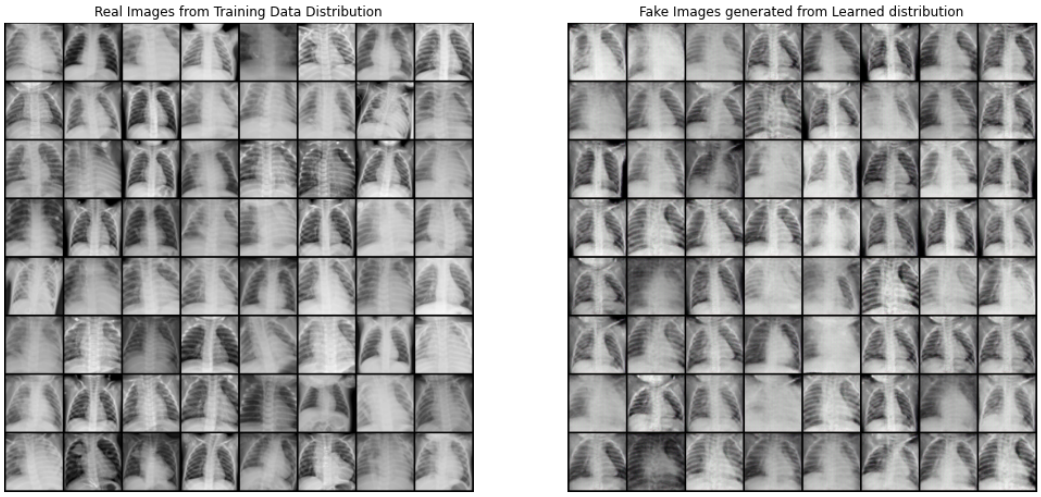

Figure 9: Real Training Images (left) and Fake Images generated by the network (right)

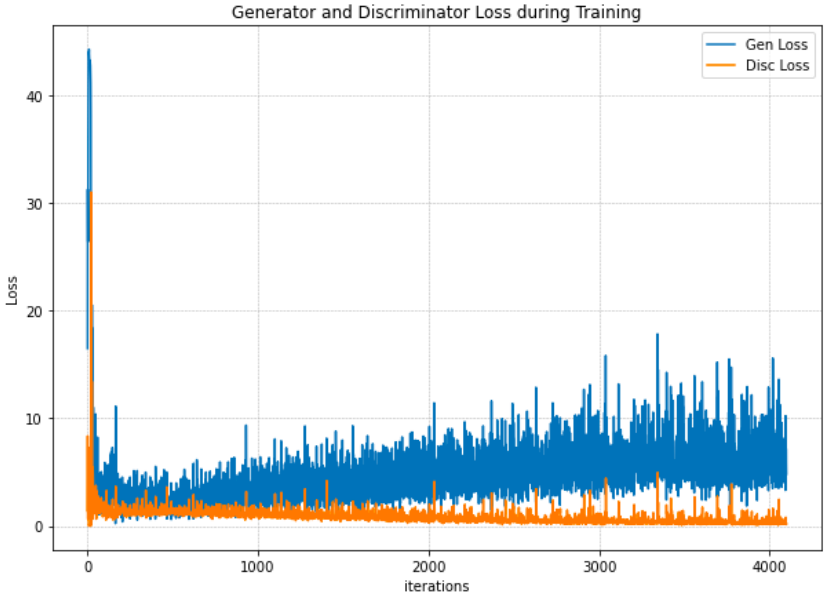

Figure 10: Training Loss with Iterations

5.5 Experiment-5

Added an extra conv $\longrightarrow$ BatchNorm $\longrightarrow$ ReLU layer in the discriminator to match dimensions along the network.

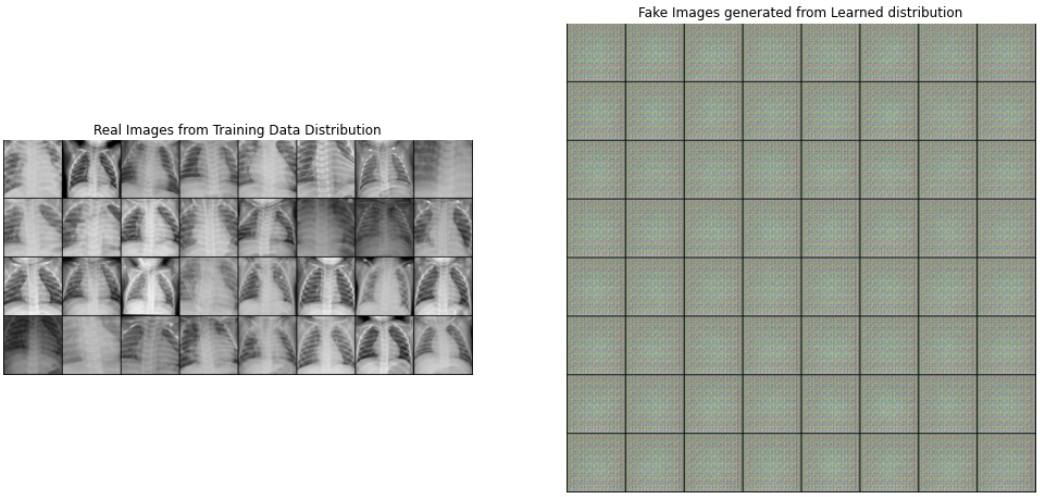

Figure 11: Real Training Images (left) and Fake Images generated by the network (right)

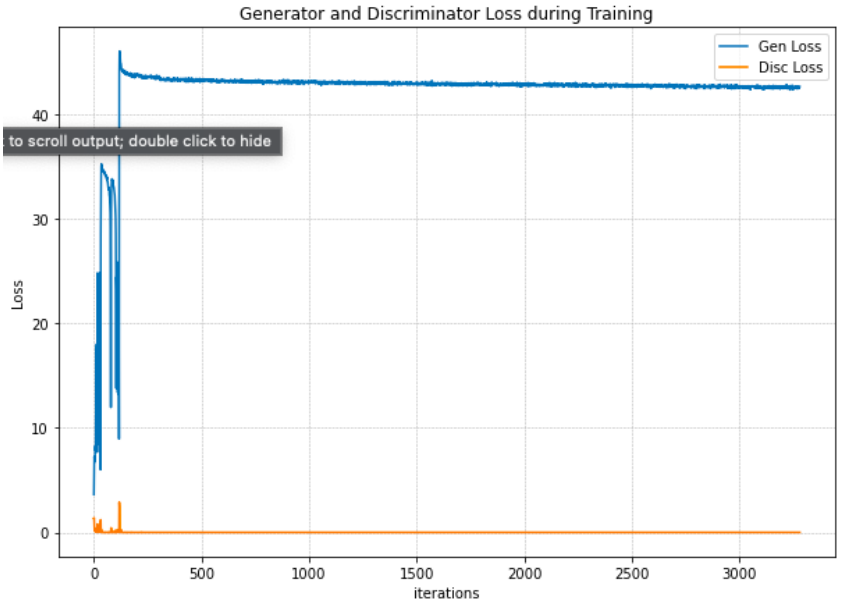

Figure 12: Training Loss with Iterations

5.6 Experiment-6

Added an extra conv $\longrightarrow$ BatchNorm $\longrightarrow$ Leaky ReLU layer in the generator network to match the dimensions along the network.

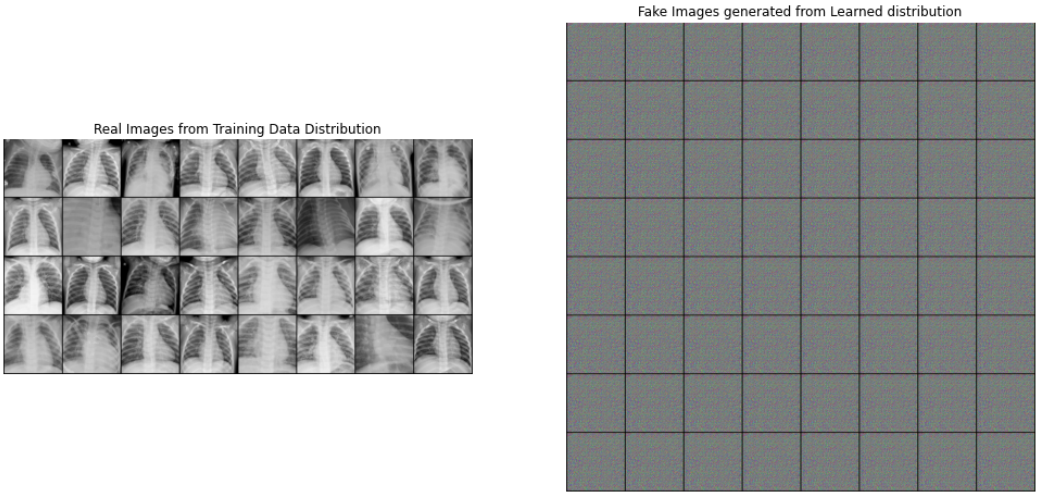

Figure 13: Real Training Images (left) and Fake Images generated by the network (right)

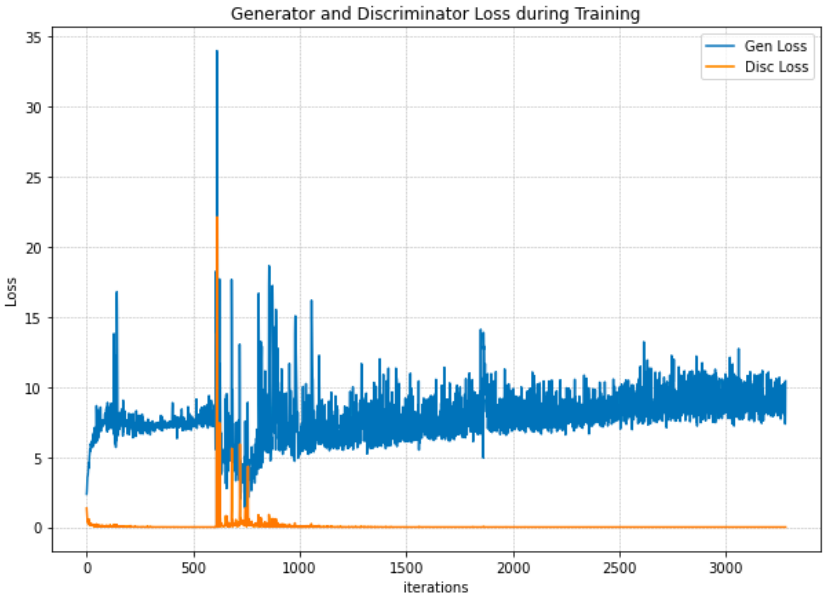

Figure 14: Training Loss with Iterations

6 Summary & Conclusion

In this report, we have trained a DCGAN network to learn to generate chest x-ray images. A DCGAN consists of a convolution-based a generator and a discriminator network. From the experimental results, we have observed that GAN training is very unstable and a small change in hyperparameter settings can lead to training collapse with little explanation as to what went wrong. In table 1 experiment 1 and 2, we observe that changing number of latent space channels L_{ch} has little effect on the metrics as well as on the generated fake samples. In experiments 1, 3, & 4, we have changed the learning rate α keeping other hyperparameters fixed. $\alpha = 2e^{-4}$ has the lowest FID, while $\alpha = 1e^{-4}$ has the highest IS. Once again, it is difficult to understand and estimate optimal hyperparameters just based on metrics. In experiment 5, we have made small modification to the discriminator network by adding an extra convolution layer, using a 5x5 spatial size latent tensor instead of 1x1, resized the image to 128 during loading instead of 64, and also modified the feature depth factor N_{gf} and N_{df}. But due to unstable training, the generated samples are just noise which is corroborated from the poor IS and FID scores. Similarly, in experiment 6 as well, changing the N_{gf} and N_{df} resulted in poor training and results. Based on our experiments, the hyperparameters selected in experiment 1, 2, 3, 4 resulted in meaningful fake chest x-ray images (hardly perceivable as fake) while experiment 5, 6 result in poor training and results. In addition, I performed multiple other experiments by changing M, L_{ch}, L_w, N_{gf} and N_{df} but everytime the training was unsuccessful and it produced fake images similar to Figure 13 (right).

References

- https://github.com/vdumoulin/conv_arithmetic/blob/master/README.md

- https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

- https://github.com/soumith/ganhacks

- https://github.com/pytorch/examples/issues/70

- DCGAN paper: https://arxiv.org/abs/1511.06434