# DIP-HW14

January 22, 2020

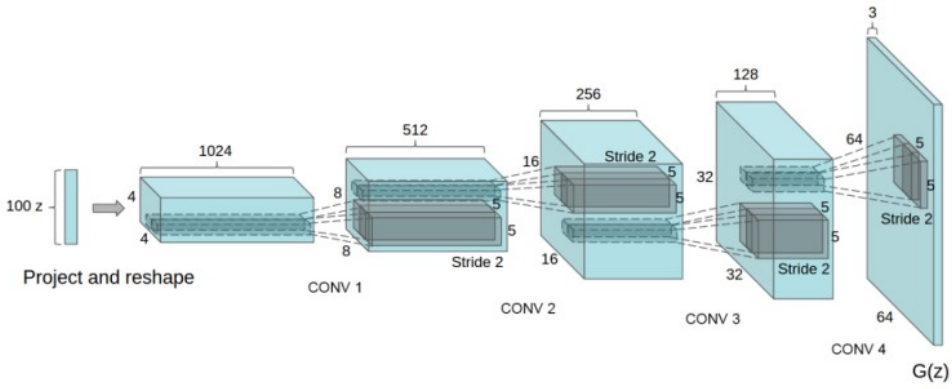
## 1 Digital Image Processing - HW14 - 98722278 - Mohammad Doosti Lakhani

In this notebook, I have solved the assignment's problems which are as follows:

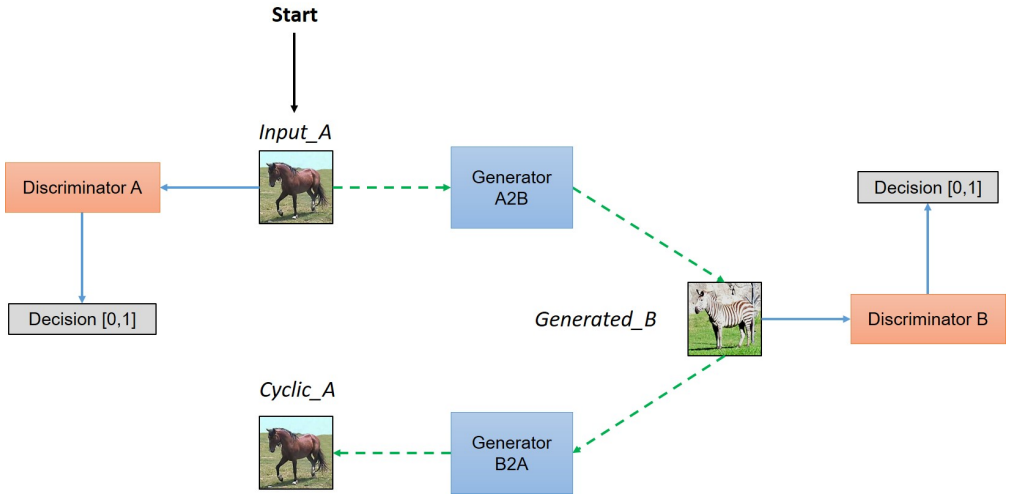
1. Read Cycle GAN paper and report a summary.
2. Answer these questions about PCA and AutoEncoders:
    1. Supervision aspect.
    2. Trainable aspect.
    3. Compare from different aspects too.
3. Train a GAN network that uses below structure as the generator and train it on any dataset (except MNIST)

### 1.1 1 Cycle GAN Summary

CycleGAN is an image to image translation tasks that works when paired label image for input dataset is not available. Previous models in image to image translation task such as Pix2Pix needs paired label as they try to find a one to one mapping between each pair but CycleGAN omitted this need by trying to first learning a map from X to Y and then forcing network to be able to construct input image which means learning a map from Y to X and this is the cycle in CycleGAN name.




generator




CycleGAN arch

As we know in GANs there is a generator and a discriminator (standard models) where generator tries to create images based on what mapping it learned and the duty of discriminator is to distinguish between images which being generated by the generator and real images from dataset. Both networks are trained simultaneously. When training finished, generator model learned the underlying distribution of data and can create new images that are similar to dataset in term of style.

But there is flaw in the aforementioned approach. In simplified works generator can learn only a mapping from a group of input images that causes discriminator not to be able to classify fake images from real ones and generator produces same images all the time no matter what input is. Actually, generator discard input image and discriminator fails on particular images and loss function still decreasing as no constraintion violated.

The idea of forcing network to learn a reverse mapping from Y to X in CycleGAN cause the loss to incorporate the reconstruction of input image to be similar at the end.

G1(x) = y, G2(y)=z —> x = z

The above notations says that if a generator cannot obtain input image from the generated image from the original generator, the loss should be high. Here is the network architecture:

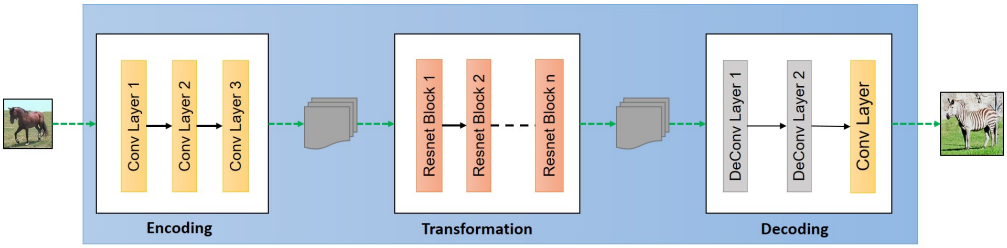
Based on the architecture CycleGAN model works in this way:

First input image `img` is fed to first generator called `g1` and `g1` tries to map `img` to another image called `gen`. `g1` is learning a map from domain `dm1` which is for `img` to domain `dm2` which stands for `gen`. Now as we do not have any paired image in dataset, we cannot ensure that `dm2` is meaningful which means the mapping learned by `g1` can be anything. So to ensure about it, the new generated image `g1` is fed to another generator `g2` (where is the reverse of `g1` when trained) to convert back `gen` to image `img`. This forces the model to learn only the meaningful domain represented by the input `img`. Also there are two discriminator, one for each generator to enforce each generator to work well in theirs single task transformation. In other words, discriminators try to enforce two objectives: 1. Generators generate images similar to real data by generating meaningful images 2. By generating images similar to input.

Here is image of CycleGAN with implementation terms:

Loss function needs to meet following constraints:

```
1. Discriminator must approve all the original images
```




CycleGAN implem

2. Discriminator must reject all the images which are generated by corresponding Generators
3. Generators must make the discriminators approve all the generated images
4. The generated image must retain the property of original image, let's say if we get `gen` fr

Both discriminators want to decreas their loss regarding their real input images `D1(a)` and `D2(a)` Both discriminators want to predict 0 for generated images so D wants to maximize `D1(G2(b))` and `D2(G1(a))` Both generators want to fool their discriminators so G wants to minimize `D2(G1(a))` and `D1(G2(b))` And cycle loss that enables network the generate images similar to their inputs which means the difference between input image to a generator and its output should be small as possible so we want to minimize `mean(a-ab, b-ba)` where `ab` is the output of giving `a` to `g1` then `g2` and `ba` in reverse manner.

## 1.2 2 Answer these questions about PCA and AutoEncoders:

1. Supervision aspect.
2. Trainable aspect.
3. Compare from other aspects too.

### 1.2.1 2.A Supervision

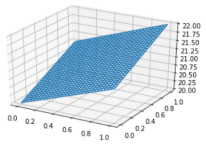
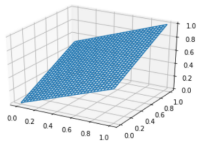
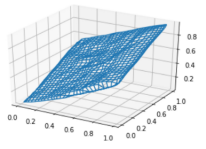
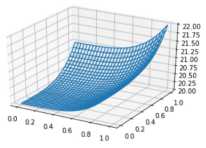
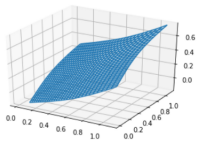
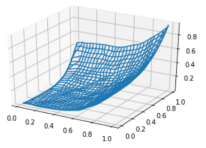
In PCA only features (X) are needed as the only computations are eigen values and coovariance matrix. But about Autoencoders it can be both supervised or unsupervised depending on task but generally and as what see in main AutoEncoders, they are unsupervised as they used to find hidden structure in given data distributions.

### 1.2.2 2.B Trianable

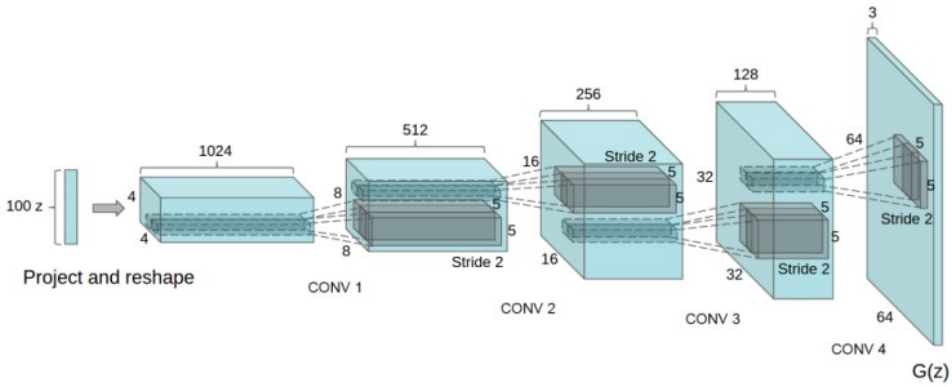
PCA uses eigen values and eigen vectors to find most dominant factors in given data and then just transform them into new axis system using orthogonal transforms so it is not trainable. But AutoEncoders are neural network architectures that use optimizer to train model iteratively.

### 1.2.3 2.C Other Aspects

First thing about PCA is that it only works on data where features are **linearly coorelated** to each other which for most of the tasks in real world it fails. But on the other hand, AutoEncoders uses conv (or fc) layers and non-linearity among them which enables it capture almost any type of corelation between two arbitrary features and based on the pattern recognition theorem, the capacity of AutoEncoders are unlimited which means they can learn any kind of corelations but the appropriate config needs to be found which is complex in practice.

| Function | Feature Space | PCA Reconstruction | Auto Encoder Reconstruction |
|---|---|---|---|
| **Plane** |  |  |  |
| **Curved Surface** |  |  |  |

pca vs ae




generator

About similarities, they both can find correlations in features so we can skip some of features and use the **dimensionality reduction* aspect of these two approaches.

About computations, as we know that PCA is linear and AutoEncoder is non-linear, and PCA is not trainable and iterative but AutoEncoder is, so it can be obtained that AutoEncoder needs much more computational power and time.

Finding correct config for AutoEncoders is hard and almost there is no fixed approach in general and it have to be learned for different data distributins while PCA only uses statistical features of data and no configuration is needed.

### 1.3 3 Train a GAN Network

1. Import Library
2. Set Hyperparameters
3. Load Data
4. Generator
5. Discriminator
6. Compile Models
7. Train Model

```
Epoch: 0        D Loss: 0.6993019580841064      D Acc: 0.29443359375 --------------- G Loss: 0.6881529092788696
Epoch: 1        D Loss: 0.696714460849762       D Acc: 0.41162109375 --------------- G Loss: 0.7033244371414185
Epoch: 2        D Loss: 0.6949199438095093      D Acc: 0.50244140625 --------------- G Loss: 0.7416453957557678
Epoch: 3        D Loss: 0.6912110447883606      D Acc: 0.49658203125 --------------- G Loss: 0.6737056970596313
Epoch: 4        D Loss: 0.6943858861923218      D Acc: 0.5546875 --------------- G Loss: 0.6980124115943909
Epoch: 5        D Loss: 0.692566990852356       D Acc: 0.568359375 --------------- G Loss: 0.7127410769462585
Epoch: 6        D Loss: 0.7016289234161377      D Acc: 0.416015625 --------------- G Loss: 0.6698208451271057
Epoch: 7        D Loss: 0.6937599778175354      D Acc: 0.59375 --------------- G Loss: 0.6994764804840088
Epoch: 8        D Loss: 0.6934203505516052      D Acc: 0.47412109375 --------------- G Loss: 0.7040546536445618
Epoch: 9        D Loss: 0.6937201619148254      D Acc: 0.52392578125 --------------- G Loss: 0.705232560634613
Epoch: 10       D Loss: 0.6920966506004333      D Acc: 0.5185546875 --------------- G Loss: 0.6897913217544556
Epoch: 11       D Loss: 0.6934925317764282      D Acc: 0.49658203125 --------------- G Loss: 0.6870988011360168
Epoch: 12       D Loss: 0.6917318105697632      D Acc: 0.54541015625 --------------- G Loss: 0.6906412839889526
Epoch: 13       D Loss: 0.6920453310012817      D Acc: 0.54248046875 --------------- G Loss: 0.6967853903770447
Epoch: 14       D Loss: 0.6925050020217896      D Acc: 0.51513671875 --------------- G Loss: 0.7081882357597351
Epoch: 15       D Loss: 0.6913422346115112      D Acc: 0.51806640625 --------------- G Loss: 0.6866915822029114
Epoch: 16       D Loss: 0.6951287388801575      D Acc: 0.51318359375 --------------- G Loss: 0.681786298751831
Epoch: 17       D Loss: 0.6979186534881592      D Acc: 0.486328125 --------------- G Loss: 0.7036430239677429
Epoch: 18       D Loss: 0.6917332410812378      D Acc: 0.59716796875 --------------- G Loss: 0.6961584687232971
Epoch: 19       D Loss: 0.6949490904808044      D Acc: 0.494140625 --------------- G Loss: 0.6847312450408936
Epoch: 20       D Loss: 0.6910027861595154      D Acc: 0.50146484375 --------------- G Loss: 0.7286595106124878
Epoch: 21       D Loss: 0.6941105127334595      D Acc: 0.50244140625 --------------- G Loss: 0.6965697407722473
Epoch: 22       D Loss: 0.6896458268165588      D Acc: 0.5166015625 --------------- G Loss: 0.687188446521759
Epoch: 23       D Loss: 0.6918269395828247      D Acc: 0.4638671875 --------------- G Loss: 0.7151263356208801
Epoch: 24       D Loss: 0.6970396041870117      D Acc: 0.37158203125 --------------- G Loss: 0.6917413473129272
Epoch: 25       D Loss: 0.6908583045005798      D Acc: 0.56201171875 --------------- G Loss: 0.6944567561149597
Epoch: 26       D Loss: 0.6870847344398499      D Acc: 0.57275390625 --------------- G Loss: 0.7273502945899963
Epoch: 27       D Loss: 0.6927096843719482      D Acc: 0.48291015625 --------------- G Loss: 0.7067096829414368
Epoch: 28       D Loss: 0.6989367604255676      D Acc: 0.24853515625 --------------- G Loss: 0.6900167465209961
Epoch: 29       D Loss: 0.6941508650779724      D Acc: 0.421875 --------------- G Loss: 0.6897047758102417
Epoch: 30       D Loss: 0.6918132901191711      D Acc: 0.55615234375 --------------- G Loss: 0.6975005269050598
Epoch: 31       D Loss: 0.6864214539527893      D Acc: 0.5 --------------- G Loss: 0.670982837677002
Epoch: 32       D Loss: 0.6911531090736389      D Acc: 0.521484375 --------------- G Loss: 0.7466565370559692
Epoch: 33       D Loss: 0.6934551000595093      D Acc: 0.4814453125 --------------- G Loss: 0.6913794279098511
Epoch: 34       D Loss: 0.6890080571174622      D Acc: 0.56689453125 --------------- G Loss: 0.6925160884857178
Epoch: 35       D Loss: 0.6976221799850464      D Acc: 0.5126953125 --------------- G Loss: 0.7045781016349792
```

train fail on 1024

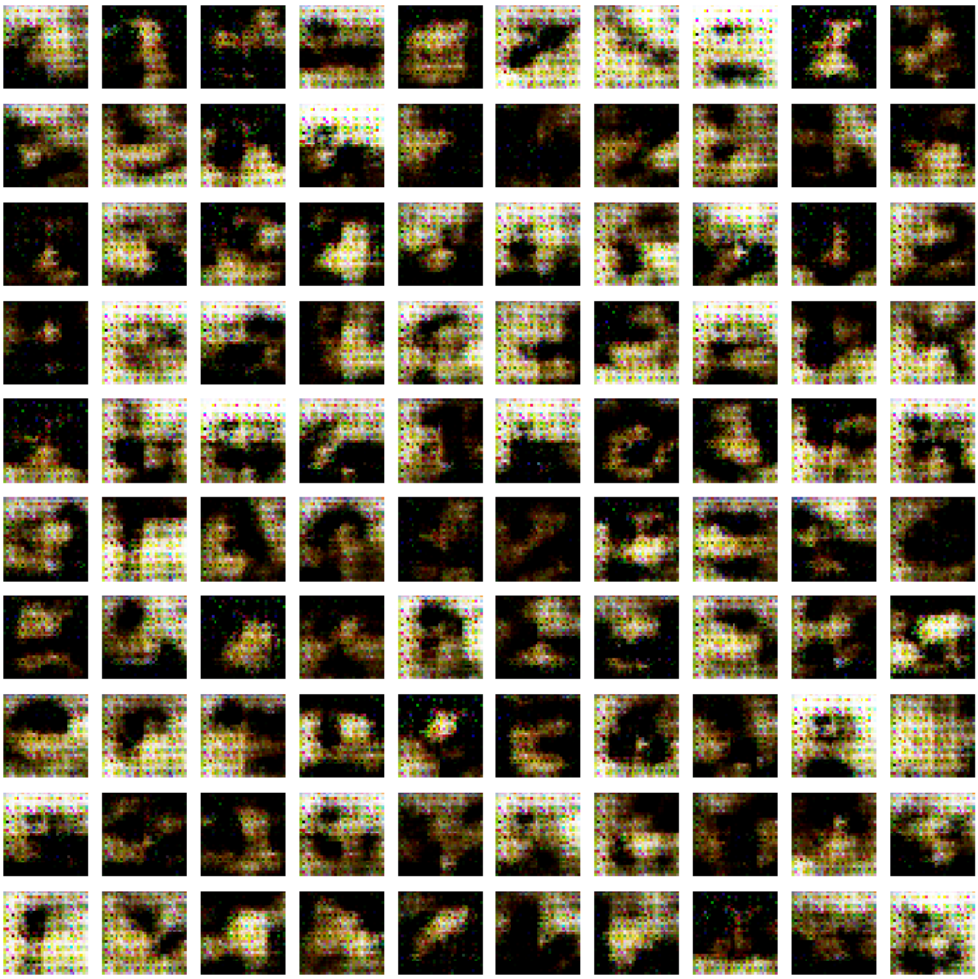
_** NOTE: in my experiments using 1024 channels in the first layer of generator hinders model to learn anything within 100 30 epochs so I changed it to smaller value such as 256 and model learned much better even in 5 epochs. **_

Trained generator after 30 epoch with 1024 channels:

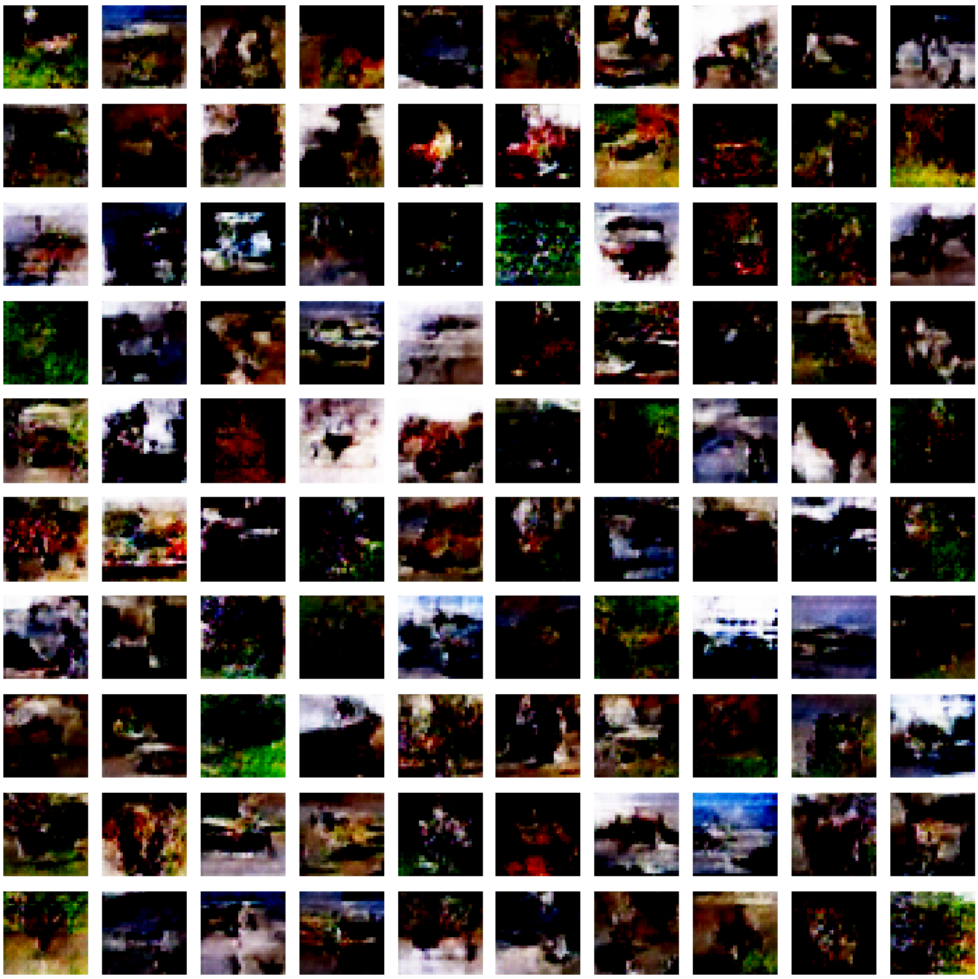
Trained generator after 30 epoch with 256 channels:

### 1.3.1 3.A Import Library

```
In [ ]: %tensorflow_version 1.x

        import os

        import matplotlib.pyplot as plt
        import numpy as np

        from keras.datasets import cifar10
        from keras.layers import Activation
        from keras.layers.advanced_activations import LeakyReLU
        from keras import initializers
        from keras.layers import (BatchNormalization, Conv2D, Conv2DTranspose, Dense,
                                  Dropout, Flatten, Input, Reshape, UpSampling2D,
                                  ZeroPadding2D)
        from keras.layers.advanced_activations import LeakyReLU
        from keras.models import Model, Sequential
        from keras.optimizers import Adam
```



output 30 epoch 1024



output 30 epoch 256

```
from PIL import Image, ImageDraw
```

### 1.3.2 3.B Set Hyperparameters

In [ ]:
```
latent_dim = 100

batch_size = 256
epochs = 200

img_rows = 32
img_cols = 32
channels = 3
kernel_size = 3

save_path = 'model'

img_rows, img_cols, channels = 32, 32, 3
img_shape = (img_rows, img_cols, channels)

optimizer = Adam(0.0002, 0.5)

if not os.path.isdir(save_path):
    os.mkdir(save_path)
```

### 1.3.3 3.C Load Data

In [23]:
```
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
x_train = (x_train.astype(np.float32) - 127.5) / 127.5
print(x_train.shape)
```

```
(50000, 32, 32, 3)
```

### 1.3.4 3.D Generator

In [24]:
```
def create_generator():
    model = Sequential()

    model.add(Dense(256 * 4 * 4, input_shape=(latent_dim,)))
    model.add(LeakyReLU(alpha=0.2))
    model.add(Reshape((4, 4, 256)))

    model.add(Conv2DTranspose(1024, kernel_size=(2, 2), padding='same'))
    model.add(LeakyReLU(alpha=0.2))

    model.add(Conv2DTranspose(128, kernel_size=(2, 2), strides=2, padding='same'))
    model.add(LeakyReLU(alpha=0.2))
```

```
            model.add(Conv2DTranspose(128, kernel_size=(2, 2), strides=2, padding='same'))
            model.add(LeakyReLU(alpha=0.2))

            model.add(Conv2DTranspose(64, kernel_size=(2, 2), strides=2, padding='same'))
            model.add(LeakyReLU(alpha=0.2))

            model.add(Conv2DTranspose(3, kernel_size=(2, 2), strides=1, padding='same'))
            model.add(Activation('tanh'))

            model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy
            return model
        generator = create_generator()
        generator.summary()

        # note as my dataset images are 32x32, I changed last layer of generator to output 32
        # as interpolating this size to 64x64 is not good for training a small model like thi
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_6 (Dense)              (None, 4096)              413696
_________________________________________________________________
leaky_re_lu_26 (LeakyReLU)   (None, 4096)              0
_________________________________________________________________
reshape_4 (Reshape)          (None, 4, 4, 256)         0
_________________________________________________________________
conv2d_transpose_16 (Conv2DT (None, 4, 4, 1024)        1049600
_________________________________________________________________
leaky_re_lu_27 (LeakyReLU)   (None, 4, 4, 1024)        0
_________________________________________________________________
conv2d_transpose_17 (Conv2DT (None, 8, 8, 128)         524416
_________________________________________________________________
leaky_re_lu_28 (LeakyReLU)   (None, 8, 8, 128)         0
_________________________________________________________________
conv2d_transpose_18 (Conv2DT (None, 16, 16, 128)       65664
_________________________________________________________________
leaky_re_lu_29 (LeakyReLU)   (None, 16, 16, 128)       0
_________________________________________________________________
conv2d_transpose_19 (Conv2DT (None, 32, 32, 64)        32832
_________________________________________________________________
leaky_re_lu_30 (LeakyReLU)   (None, 32, 32, 64)        0
_________________________________________________________________
conv2d_transpose_20 (Conv2DT (None, 32, 32, 3)         771
_________________________________________________________________
activation_4 (Activation)    (None, 32, 32, 3)         0
=================================================================
Total params: 2,086,979
```

```
Trainable params: 2,086,979
Non-trainable params: 0
_________________________________________________________________
```

### 1.3.5 3.E Discriminator

```
In [25]: def create_descriminator():
             model = Sequential()

             model.add(Conv2D(32, kernel_size=kernel_size, strides=2, input_shape=img_shape, pa
             model.add(LeakyReLU(alpha=0.2))
             model.add(Dropout(0.25))

             model.add(Conv2D(64, kernel_size=kernel_size, strides=2, padding="same"))
             model.add(ZeroPadding2D(padding=((0, 1), (0, 1))))
             model.add(LeakyReLU(alpha=0.2))
             model.add(Dropout(0.25))

             model.add(Conv2D(128, kernel_size=kernel_size, strides=2, padding="same"))
             model.add(LeakyReLU(alpha=0.2))
             model.add(Dropout(0.25))

             model.add(Conv2D(256, kernel_size=kernel_size, strides=1, padding="same"))
             model.add(LeakyReLU(alpha=0.2))
             model.add(Dropout(0.25))

             model.add(Conv2D(512, kernel_size=kernel_size, strides=1, padding="same"))
             model.add(LeakyReLU(alpha=0.2))
             model.add(Dropout(0.25))

             model.add(Flatten())
             model.add(Dense(1, activation='sigmoid'))

             model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy
             return model
         discriminator = create_descriminator()
         discriminator.summary()
```

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_11 (Conv2D)           (None, 16, 16, 32)        896
_________________________________________________________________
leaky_re_lu_31 (LeakyReLU)   (None, 16, 16, 32)        0
_________________________________________________________________
dropout_11 (Dropout)         (None, 16, 16, 32)        0
```

```
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 8, 8, 64)          18496
_________________________________________________________________
zero_padding2d_3 (ZeroPaddin (None, 9, 9, 64)          0
_________________________________________________________________
leaky_re_lu_32 (LeakyReLU)   (None, 9, 9, 64)          0
_________________________________________________________________
dropout_12 (Dropout)         (None, 9, 9, 64)          0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 5, 5, 128)         73856
_________________________________________________________________
leaky_re_lu_33 (LeakyReLU)   (None, 5, 5, 128)         0
_________________________________________________________________
dropout_13 (Dropout)         (None, 5, 5, 128)         0
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 5, 5, 256)         295168
_________________________________________________________________
leaky_re_lu_34 (LeakyReLU)   (None, 5, 5, 256)         0
_________________________________________________________________
dropout_14 (Dropout)         (None, 5, 5, 256)         0
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 5, 5, 512)         1180160
_________________________________________________________________
leaky_re_lu_35 (LeakyReLU)   (None, 5, 5, 512)         0
_________________________________________________________________
dropout_15 (Dropout)         (None, 5, 5, 512)         0
_________________________________________________________________
flatten_3 (Flatten)          (None, 12800)             0
_________________________________________________________________
dense_7 (Dense)              (None, 1)                 12801
=================================================================
Total params: 1,581,377
Trainable params: 1,581,377
Non-trainable params: 0
_________________________________________________________________
```

### 1.3.6 3.F Compile Models

In [ ]:
```
discriminator.trainable = False

        gan_input = Input(shape=(latent_dim,))
        fake_image = generator(gan_input)

        gan_output = discriminator(fake_image)

        gan = Model(gan_input, gan_output)
        gan.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
```

```
        def show_images(noise, epoch=None):
            generated_images = generator.predict(noise)
            plt.figure(figsize=(10, 10))

            for i, image in enumerate(generated_images):
                image = np.clip(image, 0, 1)
                plt.subplot(10, 10, i+1)
                plt.imshow(image.reshape((img_rows, img_cols, channels)))
                plt.axis('off')

            plt.tight_layout()

            if epoch != None:
                plt.savefig(f'{save_path}/gan-images_epoch-{epoch}.png')
```

### 1.3.7 3.G Train and Visualize Model

```
In [27]: steps_per_epoch = x_train.shape[0]//batch_size
         static_noise = np.random.normal(0, 1, size=(100, latent_dim))

         for epoch in range(epochs):
             for batch in range(steps_per_epoch):
                 noise = np.random.normal(0, 1, size=(batch_size, latent_dim))
                 real_x = x_train[np.random.randint(0, x_train.shape[0], size=batch_size)]

                 fake_x = generator.predict(noise)
                 x = np.concatenate((real_x, fake_x))
                 y = np.zeros(2*batch_size)
                 y[:batch_size] = 1

                 d_loss = discriminator.train_on_batch(x, y)

                 y_gen = np.ones(batch_size)
                 g_loss = gan.train_on_batch(noise, y_gen)

             print(f'Epoch: {epoch} \t D Loss: {d_loss[0]} \t D Acc: {d_loss[1]} ----------
             if epoch % 10 == 0:
                 show_images(static_noise, epoch)


         image_names = os.listdir(save_path)[:-1]

         frames = []
         for image in sorted(image_names, key=lambda name: int(''.join(i for i in name if i.isd
             frames.append(Image.open(save_path + '/' + image))
```

```
Epoch: 0          D Loss: 0.1408897191286087          D Acc: 0.966796875 ---------------- G Loss
```

```
Epoch: 1          D Loss: 0.6513555645942688          D Acc: 0.71875 --------------- G Loss: 1
Epoch: 2          D Loss: 0.42379525303840637          D Acc: 0.83203125 --------------- G Loss
Epoch: 3          D Loss: 0.31921884417533875          D Acc: 0.884765625 --------------- G Los
Epoch: 4          D Loss: 0.5647900700569153          D Acc: 0.701171875 --------------- G Loss
Epoch: 5          D Loss: 0.5118107795715332          D Acc: 0.771484375 --------------- G Loss
Epoch: 6          D Loss: 0.5986407995223999          D Acc: 0.66015625 --------------- G Loss
Epoch: 7          D Loss: 0.48800066113471985          D Acc: 0.78125 --------------- G Loss:
Epoch: 8          D Loss: 0.5552944540977478          D Acc: 0.716796875 --------------- G Loss
Epoch: 9          D Loss: 0.6581202745437622          D Acc: 0.6640625 --------------- G Loss:
Epoch: 10          D Loss: 0.5801175236701965          D Acc: 0.69921875 --------------- G Loss
Epoch: 11          D Loss: 0.5812733173370361          D Acc: 0.681640625 --------------- G Los
Epoch: 12          D Loss: 0.5604685544967651          D Acc: 0.71875 --------------- G Loss: 0
Epoch: 13          D Loss: 0.6033209562301636          D Acc: 0.68359375 --------------- G Loss
Epoch: 14          D Loss: 0.6348209381103516          D Acc: 0.671875 --------------- G Loss:
Epoch: 15          D Loss: 0.5678949952125549          D Acc: 0.6953125 --------------- G Loss
Epoch: 16          D Loss: 0.6112394332885742          D Acc: 0.681640625 --------------- G Los
Epoch: 17          D Loss: 0.5592082142829895          D Acc: 0.72265625 --------------- G Loss
Epoch: 18          D Loss: 0.5175609588623047          D Acc: 0.74609375 --------------- G Loss
Epoch: 19          D Loss: 0.4672866761684418          D Acc: 0.7734375 --------------- G Loss
Epoch: 20          D Loss: 0.5417904853820801          D Acc: 0.728515625 --------------- G Los
Epoch: 21          D Loss: 0.47376978397369385          D Acc: 0.77734375 --------------- G Los
Epoch: 22          D Loss: 0.5596259832382202          D Acc: 0.701171875 --------------- G Los
Epoch: 23          D Loss: 0.5604898929595947          D Acc: 0.71875 --------------- G Loss: 1
Epoch: 24          D Loss: 0.4999118447303772          D Acc: 0.736328125 --------------- G Los
Epoch: 25          D Loss: 0.5395944118499756          D Acc: 0.720703125 --------------- G Los
Epoch: 26          D Loss: 0.561856746673584          D Acc: 0.697265625 --------------- G Loss
Epoch: 27          D Loss: 0.5806962251663208          D Acc: 0.666015625 --------------- G Los
Epoch: 28          D Loss: 0.6255284547805786          D Acc: 0.626953125 --------------- G Los
Epoch: 29          D Loss: 0.6274042129516602          D Acc: 0.630859375 --------------- G Los
Epoch: 30          D Loss: 0.5663378238677979          D Acc: 0.693359375 --------------- G Los
Epoch: 31          D Loss: 0.5341119170188904          D Acc: 0.73046875 --------------- G Loss
Epoch: 32          D Loss: 0.6359353065490723          D Acc: 0.61328125 --------------- G Loss
Epoch: 33          D Loss: 0.5769007802009583          D Acc: 0.6953125 --------------- G Loss
Epoch: 34          D Loss: 0.5590680241584778          D Acc: 0.70703125 --------------- G Loss
Epoch: 35          D Loss: 0.6305510401725769          D Acc: 0.623046875 --------------- G Los
Epoch: 36          D Loss: 0.5834179520606995          D Acc: 0.6640625 --------------- G Loss
Epoch: 37          D Loss: 0.5690737962722778          D Acc: 0.69140625 --------------- G Loss
Epoch: 38          D Loss: 0.6442967653274536          D Acc: 0.59765625 --------------- G Loss
Epoch: 39          D Loss: 0.621629536151886          D Acc: 0.626953125 --------------- G Loss
Epoch: 40          D Loss: 0.6168941855430603          D Acc: 0.65625 --------------- G Loss: 1
Epoch: 41          D Loss: 0.5536731481552124          D Acc: 0.712890625 --------------- G Los
Epoch: 42          D Loss: 0.6024651527404785          D Acc: 0.65234375 --------------- G Loss
Epoch: 43          D Loss: 0.6426838636398315          D Acc: 0.642578125 --------------- G Los
Epoch: 44          D Loss: 0.5901310443878174          D Acc: 0.654296875 --------------- G Los
Epoch: 45          D Loss: 0.6112076640129089          D Acc: 0.673828125 --------------- G Los
Epoch: 46          D Loss: 0.614761471748352          D Acc: 0.6484375 --------------- G Loss:
Epoch: 47          D Loss: 0.6593266725540161          D Acc: 0.619140625 --------------- G Los
Epoch: 48          D Loss: 0.6312886476516724          D Acc: 0.6171875 --------------- G Loss
```

```
Epoch: 49          D Loss: 0.6259385347366333          D Acc: 0.599609375 --------------- G Los
Epoch: 50          D Loss: 0.6381340026855469          D Acc: 0.59765625 --------------- G Loss
Epoch: 51          D Loss: 0.616797685623169          D Acc: 0.640625 --------------- G Loss: 0
Epoch: 52          D Loss: 0.5856744050979614          D Acc: 0.697265625 --------------- G Los
Epoch: 53          D Loss: 0.6203483939170837          D Acc: 0.630859375 --------------- G Los
Epoch: 54          D Loss: 0.6542762517929077          D Acc: 0.59765625 --------------- G Loss
Epoch: 55          D Loss: 0.6311525106430054          D Acc: 0.62109375 --------------- G Loss
Epoch: 56          D Loss: 0.6179268956184387          D Acc: 0.638671875 --------------- G Los
Epoch: 57          D Loss: 0.6248496770858765          D Acc: 0.666015625 --------------- G Los
Epoch: 58          D Loss: 0.6146047711372375          D Acc: 0.6796875 --------------- G Loss
Epoch: 59          D Loss: 0.6326285004615784          D Acc: 0.63671875 --------------- G Loss
Epoch: 60          D Loss: 0.6265875697135925          D Acc: 0.64453125 --------------- G Loss
Epoch: 61          D Loss: 0.6299328804016113          D Acc: 0.595703125 --------------- G Los
Epoch: 62          D Loss: 0.6212470531463623          D Acc: 0.623046875 --------------- G Los
Epoch: 63          D Loss: 0.6279608607292175          D Acc: 0.66015625 --------------- G Loss
Epoch: 64          D Loss: 0.6593545079231262          D Acc: 0.580078125 --------------- G Los
Epoch: 65          D Loss: 0.642648458480835          D Acc: 0.625 --------------- G Loss: 0.84
Epoch: 66          D Loss: 0.6350326538085938          D Acc: 0.625 --------------- G Loss: 0.8
Epoch: 67          D Loss: 0.6746023297309875          D Acc: 0.6328125 --------------- G Loss
Epoch: 68          D Loss: 0.6527559757232666          D Acc: 0.619140625 --------------- G Los
Epoch: 69          D Loss: 0.611768364906311          D Acc: 0.63671875 --------------- G Loss
Epoch: 70          D Loss: 0.6212946772575378          D Acc: 0.634765625 --------------- G Los
Epoch: 71          D Loss: 0.6104242205619812          D Acc: 0.6484375 --------------- G Loss
Epoch: 72          D Loss: 0.60712069272995          D Acc: 0.658203125 --------------- G Loss
Epoch: 73          D Loss: 0.6361660957336426          D Acc: 0.623046875 --------------- G Los
Epoch: 74          D Loss: 0.6148809194564819          D Acc: 0.66015625 --------------- G Loss
Epoch: 75          D Loss: 0.6466186046600342          D Acc: 0.599609375 --------------- G Los
Epoch: 76          D Loss: 0.6516079306602478          D Acc: 0.56640625 --------------- G Loss
Epoch: 77          D Loss: 0.606248140335083          D Acc: 0.65234375 --------------- G Loss
Epoch: 78          D Loss: 0.6208204627037048          D Acc: 0.630859375 --------------- G Los
Epoch: 79          D Loss: 0.6448211669921875          D Acc: 0.6171875 --------------- G Loss
Epoch: 80          D Loss: 0.6287379264831543          D Acc: 0.6328125 --------------- G Loss
Epoch: 81          D Loss: 0.5998979210853577          D Acc: 0.662109375 --------------- G Los
Epoch: 82          D Loss: 0.611952006816864          D Acc: 0.62890625 --------------- G Loss
Epoch: 83          D Loss: 0.6314682960510254          D Acc: 0.62890625 --------------- G Loss
Epoch: 84          D Loss: 0.6220043897628784          D Acc: 0.634765625 --------------- G Los
Epoch: 85          D Loss: 0.6375966668128967          D Acc: 0.626953125 --------------- G Los
Epoch: 86          D Loss: 0.6361115574836731          D Acc: 0.640625 --------------- G Loss:
Epoch: 87          D Loss: 0.6174578666687012          D Acc: 0.646484375 --------------- G Los
Epoch: 88          D Loss: 0.6379070281982422          D Acc: 0.62890625 --------------- G Loss
Epoch: 89          D Loss: 0.6175389289855957          D Acc: 0.650390625 --------------- G Los
Epoch: 90          D Loss: 0.6122394800186157          D Acc: 0.62890625 --------------- G Loss
Epoch: 91          D Loss: 0.658196210861206          D Acc: 0.6015625 --------------- G Loss:
Epoch: 92          D Loss: 0.5822438597679138          D Acc: 0.693359375 --------------- G Los
Epoch: 93          D Loss: 0.6353254914283752          D Acc: 0.640625 --------------- G Loss:
Epoch: 94          D Loss: 0.6427758932113647          D Acc: 0.609375 --------------- G Loss:
Epoch: 95          D Loss: 0.6327270865440369          D Acc: 0.6171875 --------------- G Loss
Epoch: 96          D Loss: 0.6238670349121094          D Acc: 0.64453125 --------------- G Loss
```

```
Epoch: 97          D Loss: 0.6283705830574036          D Acc: 0.619140625 --------------- G Los
Epoch: 98          D Loss: 0.6361263394355774          D Acc: 0.626953125 --------------- G Los
Epoch: 99          D Loss: 0.6164047718048096          D Acc: 0.642578125 --------------- G Los
Epoch: 100          D Loss: 0.6373767852783203          D Acc: 0.650390625 --------------- G Lo
Epoch: 101          D Loss: 0.5890966653823853          D Acc: 0.6953125 --------------- G Loss
Epoch: 102          D Loss: 0.6404650807380676          D Acc: 0.615234375 --------------- G Lo
Epoch: 103          D Loss: 0.6285912990570068          D Acc: 0.642578125 --------------- G Lo
Epoch: 104          D Loss: 0.589664101600647          D Acc: 0.6953125 --------------- G Loss
Epoch: 105          D Loss: 0.6281032562255859          D Acc: 0.609375 --------------- G Loss
Epoch: 106          D Loss: 0.6015932559967041          D Acc: 0.669921875 --------------- G Lo
Epoch: 107          D Loss: 0.6246393322944641          D Acc: 0.640625 --------------- G Loss
Epoch: 108          D Loss: 0.634857177734375          D Acc: 0.6171875 --------------- G Loss
Epoch: 109          D Loss: 0.626362681388855          D Acc: 0.654296875 --------------- G Los
Epoch: 110          D Loss: 0.6123062968254089          D Acc: 0.646484375 --------------- G Lo
Epoch: 111          D Loss: 0.6520756483078003          D Acc: 0.587890625 --------------- G Lo
Epoch: 112          D Loss: 0.636002242565155          D Acc: 0.62109375 --------------- G Loss
Epoch: 113          D Loss: 0.6274649500846863          D Acc: 0.642578125 --------------- G Lo
Epoch: 114          D Loss: 0.5952185988426208          D Acc: 0.66796875 --------------- G Los
Epoch: 115          D Loss: 0.5889849662780762          D Acc: 0.673828125 --------------- G Lo
Epoch: 116          D Loss: 0.6102638244628906          D Acc: 0.666015625 --------------- G Lo
Epoch: 117          D Loss: 0.621018648147583          D Acc: 0.65234375 --------------- G Loss
Epoch: 118          D Loss: 0.5991860628128052          D Acc: 0.662109375 --------------- G Lo
Epoch: 119          D Loss: 0.6025831699371338          D Acc: 0.666015625 --------------- G Lo
Epoch: 120          D Loss: 0.6055877208709717          D Acc: 0.662109375 --------------- G Lo
Epoch: 121          D Loss: 0.5988187193870544          D Acc: 0.646484375 --------------- G Lo
Epoch: 122          D Loss: 0.6257717609405518          D Acc: 0.646484375 --------------- G Lo
Epoch: 123          D Loss: 0.6083186864852905          D Acc: 0.671875 --------------- G Loss
Epoch: 124          D Loss: 0.607736349105835          D Acc: 0.630859375 --------------- G Los
Epoch: 125          D Loss: 0.6274092197418213          D Acc: 0.642578125 --------------- G Lo
Epoch: 126          D Loss: 0.5599309802055359          D Acc: 0.697265625 --------------- G Lo
Epoch: 127          D Loss: 0.6367396116256714          D Acc: 0.638671875 --------------- G Lo
Epoch: 128          D Loss: 0.6076158881187439          D Acc: 0.650390625 --------------- G Lo
Epoch: 129          D Loss: 0.6230829954147339          D Acc: 0.625 --------------- G Loss: 0
Epoch: 130          D Loss: 0.6006773710250854          D Acc: 0.662109375 --------------- G Lo
Epoch: 131          D Loss: 0.5911728143692017          D Acc: 0.693359375 --------------- G Lo
Epoch: 132          D Loss: 0.5902551412582397          D Acc: 0.671875 --------------- G Loss
Epoch: 133          D Loss: 0.6015951037406921          D Acc: 0.65625 --------------- G Loss:
Epoch: 134          D Loss: 0.6080767512321472          D Acc: 0.673828125 --------------- G Lo
Epoch: 135          D Loss: 0.6115170121192932          D Acc: 0.658203125 --------------- G Lo
Epoch: 136          D Loss: 0.6226879954338074          D Acc: 0.6484375 --------------- G Loss
Epoch: 137          D Loss: 0.6253523230552673          D Acc: 0.62890625 --------------- G Los
Epoch: 138          D Loss: 0.6031507253646851          D Acc: 0.662109375 --------------- G Lo
Epoch: 139          D Loss: 0.6416237354278564          D Acc: 0.6171875 --------------- G Loss
Epoch: 140          D Loss: 0.5730687975883484          D Acc: 0.693359375 --------------- G Lo
Epoch: 141          D Loss: 0.5996060371398926          D Acc: 0.673828125 --------------- G Lo
Epoch: 142          D Loss: 0.6123886108398438          D Acc: 0.63671875 --------------- G Los
Epoch: 143          D Loss: 0.6074057221412659          D Acc: 0.658203125 --------------- G Lo
Epoch: 144          D Loss: 0.6154401302337646          D Acc: 0.646484375 --------------- G Lo
```

```
Epoch: 145          D Loss: 0.5829581618309021          D Acc: 0.689453125 --------------- G Lo
Epoch: 146          D Loss: 0.6004339456558228          D Acc: 0.65625 --------------- G Loss:
Epoch: 147          D Loss: 0.5882567763328552          D Acc: 0.705078125 --------------- G Lo
Epoch: 148          D Loss: 0.5860982537269592          D Acc: 0.65625 --------------- G Loss:
Epoch: 149          D Loss: 0.5908358693122864          D Acc: 0.66796875 --------------- G Los
Epoch: 150          D Loss: 0.5958592891693115          D Acc: 0.681640625 --------------- G Lo
Epoch: 151          D Loss: 0.6024646162986755          D Acc: 0.6875 --------------- G Loss: 0
Epoch: 152          D Loss: 0.5767725706100464          D Acc: 0.6875 --------------- G Loss: 0
Epoch: 153          D Loss: 0.6102381944656372          D Acc: 0.654296875 --------------- G Lo
Epoch: 154          D Loss: 0.5945796966552734          D Acc: 0.6796875 --------------- G Loss
Epoch: 155          D Loss: 0.615073561668396          D Acc: 0.642578125 --------------- G Los
Epoch: 156          D Loss: 0.5946037173271179          D Acc: 0.66796875 --------------- G Los
Epoch: 157          D Loss: 0.6075536012649536          D Acc: 0.6640625 --------------- G Loss
Epoch: 158          D Loss: 0.6094079613685608          D Acc: 0.6640625 --------------- G Loss
Epoch: 159          D Loss: 0.5913312435150146          D Acc: 0.666015625 --------------- G Lo
Epoch: 160          D Loss: 0.5890749096870422          D Acc: 0.689453125 --------------- G Lo
Epoch: 161          D Loss: 0.6228742003440857          D Acc: 0.64453125 --------------- G Los
Epoch: 162          D Loss: 0.5968111157417297          D Acc: 0.677734375 --------------- G Lo
Epoch: 163          D Loss: 0.6343427300453186          D Acc: 0.646484375 --------------- G Lo
Epoch: 164          D Loss: 0.6172620058059692          D Acc: 0.65625 --------------- G Loss:
Epoch: 165          D Loss: 0.5918434262275696          D Acc: 0.66796875 --------------- G Los
Epoch: 166          D Loss: 0.5665353536605835          D Acc: 0.697265625 --------------- G Lo
Epoch: 167          D Loss: 0.6408085823059082          D Acc: 0.607421875 --------------- G Lo
Epoch: 168          D Loss: 0.6031060218811035          D Acc: 0.65625 --------------- G Loss:
Epoch: 169          D Loss: 0.637352466583252          D Acc: 0.640625 --------------- G Loss:
Epoch: 170          D Loss: 0.6019368171691895          D Acc: 0.67578125 --------------- G Los
Epoch: 171          D Loss: 0.6117047071456909          D Acc: 0.673828125 --------------- G Lo
Epoch: 172          D Loss: 0.5506842732429504          D Acc: 0.71484375 --------------- G Los
Epoch: 173          D Loss: 0.5998166799545288          D Acc: 0.673828125 --------------- G Lo
Epoch: 174          D Loss: 0.5866829752922058          D Acc: 0.671875 --------------- G Loss
Epoch: 175          D Loss: 0.5803964734077454          D Acc: 0.669921875 --------------- G Lo
Epoch: 176          D Loss: 0.6170734167098999          D Acc: 0.65234375 --------------- G Los
Epoch: 177          D Loss: 0.6149045825004578          D Acc: 0.6484375 --------------- G Loss
Epoch: 178          D Loss: 0.5991706252098083          D Acc: 0.65234375 --------------- G Los
Epoch: 179          D Loss: 0.603866457939148          D Acc: 0.66015625 --------------- G Loss
Epoch: 180          D Loss: 0.5985166430473328          D Acc: 0.67578125 --------------- G Los
Epoch: 181          D Loss: 0.6045410633087158          D Acc: 0.640625 --------------- G Loss
Epoch: 182          D Loss: 0.5849644541740417          D Acc: 0.69921875 --------------- G Los
Epoch: 183          D Loss: 0.5767671465873718          D Acc: 0.6875 --------------- G Loss: 0
Epoch: 184          D Loss: 0.6070204377174377          D Acc: 0.65234375 --------------- G Los
Epoch: 185          D Loss: 0.6069952249526978          D Acc: 0.65234375 --------------- G Los
Epoch: 186          D Loss: 0.593257486820221          D Acc: 0.66796875 --------------- G Loss
Epoch: 187          D Loss: 0.604248583316803          D Acc: 0.669921875 --------------- G Los
Epoch: 188          D Loss: 0.5618681907653809          D Acc: 0.7109375 --------------- G Loss
Epoch: 189          D Loss: 0.5786077976226807          D Acc: 0.716796875 --------------- G Lo
Epoch: 190          D Loss: 0.5892927050590515          D Acc: 0.689453125 --------------- G Lo
Epoch: 191          D Loss: 0.5765275359153748          D Acc: 0.69921875 --------------- G Los
Epoch: 192          D Loss: 0.5875688195228577          D Acc: 0.69921875 --------------- G Los
```

```
Epoch: 193          D Loss: 0.613261878490448          D Acc: 0.669921875 --------------- G Los
Epoch: 194          D Loss: 0.5765141248703003          D Acc: 0.69921875 --------------- G Los
Epoch: 195          D Loss: 0.6062048077583313          D Acc: 0.64453125 --------------- G Los
Epoch: 196          D Loss: 0.5531201958656311          D Acc: 0.712890625 --------------- G Lo
Epoch: 197          D Loss: 0.6178140044212341          D Acc: 0.638671875 --------------- G Lo
Epoch: 198          D Loss: 0.6372928023338318          D Acc: 0.65234375 --------------- G Los
Epoch: 199          D Loss: 0.5954123735427856          D Acc: 0.65234375 --------------- G Los
```