# Notebook Results

See the png files for screenshots after 1, 2, 3, 4, 5 and 6 epochs. The raw results are still in the Jupyter Notebook

The only noticed bug was that the average FID distance didn't work (as the `train_epoch` doesn't return anything). I was still able to verify that my version worked up to standard by just looking at the FID distance after every 100 iterations. Unfortunately I got kicked out of COLAB after my successful test and it didn't seem worth rectifying such a minor bug.

# Conceptual Quesitons

## 1a) Discriminator loss

The discriminator loss is measuring how "bad" the discriminator is a recognising real and fake images. It is calculated by summing the binary cross-entropy loss from the real and fake images.

```
# How "bad" the discriminator was at identifying real images (ie related to how many reals it thought were fake)
real_loss = cross_entropy(tf.ones_like(real_output), real_output)

# How "bad" the discriminator was at identifying fake images (ie related to how fakes it thought were real)
fake_loss = cross_entropy(tf.zeros_like(fake_output), fake_output)

# Sum both cross-entropy losses and return
total_loss = real_loss + fake_loss
return total_loss
```

## 1b) Generator loss

As the generator is trying to create fake images that seem real to the discriminator, the generator loss measures how "bad" the generator's fakes are. The generator loss will increase with the number of fake images that the discriminator correctly recognises as fake. The loss is computed using binary cross-entropy:

```
# Compute how "bad" the generator is at creating fake images
return cross_entropy(tf.ones_like(fake_output), fake_output)
```

## 2)

Both the generator and discriminator are "competing" aggainst each other. The generator is trying to create fake images that seem real and the discriminator is trying to recognise the fakes. Alternating between the generator and discriminator is needed because the discriminator requires the output from the generator in order for it to attempt to recognise fakes.

## 3)

Mode collapse is when the generator gets "stuck" producing a single output image as opposed to mulitple variations. If the discriminator is unable to recognise that this one input is a fake, it becomes too easy for the generator to find another output image in the next iteration that fools the disriminator. The discriminator never learns its way out of the trap the the generator becomes over optimised producing a single output image.

## 4)

The Fréchet Inception Distance is used to determine how good the generator is by measuring "how similar the generated images are to the real ones". A lower FID is better as it means the real and generated images are very similar. The FID measures the distance between the respective activation distributions.

## 5)

A GAN is a type of "AI" which involves two Neural Networks which compete aggainst each other. One is called the generator which tries to learn how to create fake images. The other is the discriminator which tries to learn how to recognise fake images amoung real ones. Both networks are trained at the same time and compete with each other. GANs can be used to produce deepfakes by configuring the discriminator to compare the generators output to real version of someone's face. Eventually, given enough training and tuning, the generator is able to create a convincing version of any photo or video.