MR.1. Data Generation using Generative Adversarial Networks: Final Report

Assignment Group "Yes We Gan"
Jake Langton — Yiannis Maxwell

June 5, 2020

1 Introduction

The aim of this project is to benchmark a Generative Adversarial Network (GAN) on a dataset of celebrity face images to generate new face images. A GAN consists of two competing neural networks: a 'generator' and 'discriminator'. The generator creates random synthetic outputs (such as an image of a face), while the discriminator attempts to distinguish the generated outputs from authentic outputs (such as celebrity face images), with the goal of generating a new, realistic output. The dataset used in this project is the CelebFaces Attributes Dataset (CelebA) made available by Liu et al. (2015). It consists of 202,499 images of celebrity faces, with annotations for 5 landmark locations and 40 binary attributes (such as facial features, expression, hair type) per image.

2 Related Work

Generative Adversarial Networks were introduced by Goodfellow et al. (2014). The main challenges in designing GANs derive from their inherent instability due to training two competing networks. They are vulnerable to non-convergence and issues such as instability and mode collapse, as highlighted by Goodfellow (2016). A fundamental step toward training stable GAN models was the introduction of DCGANs by Radford et al. (2015), combining deep convolutional and generative adversarial networks. DCGANs are more stable and have become the de facto starting point for GAN models. The issue of mode collapse occurs when the generator outputs a limited range of outputs for any latent space input. Arjovsky et al. (2017) improved stability and prevented mode collapse in their Wasserstein GAN (WGAN). By using a critic rather than discriminator, and approximating the continuous and differentiable Wasserstein distance, the WGAN could be trained to optimality and eliminate collapsed modes. While methods of preventing mode collapse have been proposed, effectively evaluating mode collapse is a current problem in GAN research. Lala et al. (2018) examined several metrics including Wasserstein distance, coverage and number of captured

modes. It was found that single metrics were not sufficient to measure mode collapse, nor do they necessarily give consistent results. Further to evaluating mode collapse, methods to evaluate generated image quality are of significant interest. A popular quality metric is the Fréchet Inception Distance (FID), introduced by Heusel et al. (2017). FID uses an Inception classifier measure similarity between sets of images, and correlates with human judgement. Recent state-of-the-art GANs have generated facial images near indistinguishable from real images by human judgment. Karras et al. (2017) produced high resolution images using a progressively growing GAN, while demonstrating that growing each network progressively with increasing resolution sped up and stabilised training. Their research has been followed-up by GAN image generation with using style transfer. StyleGAN, described by Karras et al. (2018) used elements of style transfer to generate new face images using CelebA-HQ and a new dataset, Flickr-Faces-HQ. Karras et al. (2019) improved on this with StyleGAN2, with redesigned generator normalisation that improved image quality and removed artifacts.
Despite a variety of GANs having been proposed of improving performance, a study by Lucic et al. (2017) suggests that GAN performance is more dependent on computational budget and tuning than fundamental algorithmic changes. This report therefore aims to be a systematic and objective evaluation of GAN tuning. We examine the effect of adjusting learning rate and label smoothing on a DCGAN, and use FID to evaluate image quality.

3 Methods

The philosophy underpinning our experimental methodology was as such. Given that GANs are notoriously complex, with numerous hyper-parameters it would be best to start simple and increase complexity gradually. We started with the $28 \times 28 \times 1$ MNIST dataset, then used the CelebA dataset scaled to the same dimensions before expanding the model to use $64 \times 64 \times 3$ CelebA images.
This workflow was intended to allow us to migrate the basic model Tensorflow (2020) to a more complex model with elements from Radford et al. (2015) and Miyato et al. (2018). We experimented with batch normalisation, learning rate, output activation functions, network depth, dropout layers and batch size, primarily on the discriminator networks. The changes to these models are summarised in Table 2 and Table 3 of the Appendix, along with the results of different training runs on each of these models. Rather than blindly improving the model through trial and error these experiments were constructed with the explicit aim of illustrating the effect of changing individual hyper-parameters.

4 Experiments

4.1 Fréchet Inception Distance (FID)

FID is a method of measuring the similarity of 2 batches of images. The last layer prior to the output layer of a world class image recognition model Inception V3 is used to generate a feature space. Each real and fake image is mapped into this feature space. Each feature is summarised with respect to all other images in the batch (real or fake) as Gaussian with a

mean and co-variance matrix. The distance between these two distributions (real and fake) is then calculated using the Fréchet distance, also called the Wasserstein-2 distance.

Data	Batch Size	FID Relative Standard deviation
28x28x1	4	15.7
28x28x1	64	8.5
28x28x1	256	10.3
28x28x1	1024	24.0
64x64x1	64	8.0
64x64x1	1024	6.3

Table 1: Metrics for quality of generated images

The results indicate the inherent trade-offs in calculating FIDs. When evaluating a small batch size the results deviate greatly each time as only a small sample is being compared. However, for large sample sizes standard deviation increases again for low resolution images. This is likely because single pixel deviations could drastically impact the batches. The results of these experiments informed our decision to only implement FID on the $64 \times 64 \times 3$ dataset. For the smaller dataset FID was used as a guide but with full knowledge that they cannot be fully trusted. FID will be calculated with size with batch sizes of 64 as the improvement for calculating larger image batches is not worth the extra time required to do so.

4.2 CelebA $28 \times 28 \times 1$

Experimenting with the reduced sized images proved difficult and resulted in lots of challenges. Firstly, due to the greater feature complexity the MNIST GAN structure did not work well as the discriminator was far too good at separating the true and false images. In an effort to migrate a generic network structure as exhibited in Radford et al. (2015) and Miyato et al. (2018) we experimented with batch normalisation, learning rate, sigmoid discriminator outputs, network depth, dropout layers and batch size.

4.3 CelebA $64 \times 64 \times 3$

The network models were extended from Miyato et al. (2018) and adjusted to suit the input size of $64 \times 64 \times 3$ with a linear activation of the discriminator output and an adjusted learning rate for the generator.

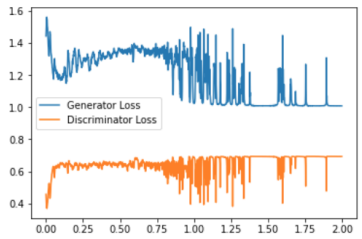
5 Results

5.1 Summary Of Results

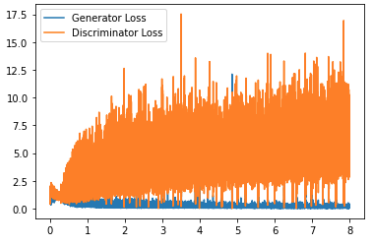
- Changing batch size from 64 to 256 made little difference. A lower batch size may have increased the stability slightly
- Deviations from the original MNIST GAN yielded no improvements until network depth is increased as was required for the $64 \times 64 \times 3$ images, where the generator learning rate was

increased from 1e-4 to 5e-4.
- Tuning hyperparameters in the discriminator was more effectual than tuning the Generators in general.
- Increasing the amount of label smoothing improved results when the discriminator was too strong but decreased performance when too high.

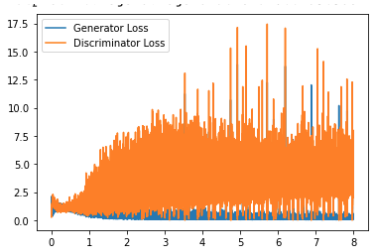
Discriminator Complexity Output Activation
- Deeper Generators made little difference to the results.
- Deeper discriminators could be used with a "linear" output activation when "sigmoid" outputs failed. These allow for prolonged stable high quality training.
- Dropout networks allow for deeper discriminator networks with more complex parameters by slowing inital training so that it does not overpower the generator.
- Sigmoid activation output on the final layer of the discriminator makes the discriminator too strong. This issue was drastically reduced by implementing a "linear" output activation function. See model #4 (sigmoid) and model #5 (linear).
- Increasing discriminator depth can stop loss from growing constantly and stabilise learning. See model #5 (shallow) and model #6 (deep)



(a) Model #4



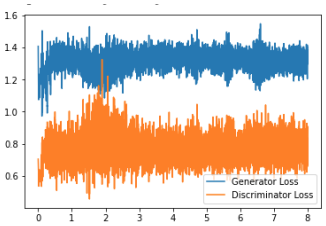
(b) Model #5



(c) Model #6

Batch Normalisation

- Batch Normalisation increased variability in the loss function when added to the discriminator for 28x28x1 datasets. This often results in unstable training. Additional methods must be implemented to ensure stability. See the difference between model #1 and model #2 (Batch Normalisation).
- This increased variability is useful when a discriminator is too good and distinguishing results and learning plateaus.

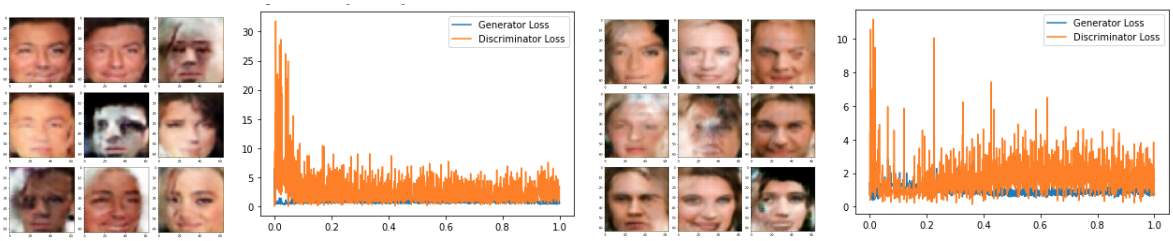


(d) Model #1

(e) Model #2

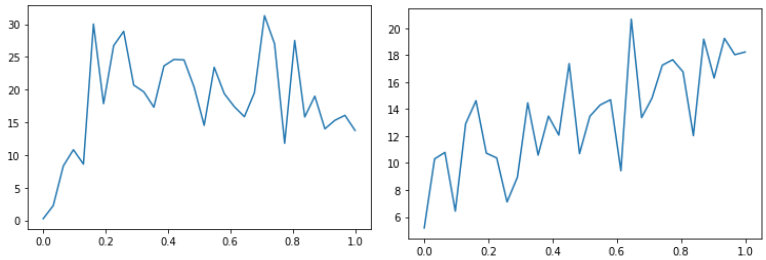
- Batch normalisation of the discriminator had a different effect on the larger image resolutions. It improved stability by reducing the oscillations of the discriminator.
- Batch normalisation of the discriminator did not improve mode collapse, if anything it

worsened it. Notice the output of model A has 3 similar faces in the top right hand corner. - FID which can be used to compare batches of generated samples for mode collapse is increasing steadily each batch when batch normalisation is not used. This indicates generated images are becoming less similar over time.



(f) Model#A (g) Model#A (h) Model #C (i) Model #C

Figure 1: Model# A uses Batch normalisation while model #C does not. The models are otherwise identical



(a) Model#A (b) Model#C

Figure 2: FID between output samples is used as a proxy for measuring mode collapse

6 Conclusion

In this report we benchmarked a DCGAN using the CelebA dataset. Different batch sizes, learning rate, label smoothing, network depth and image resolutions were tested. The results suggest batch normalisation does not significantly improve discriminator performance. Our results indicated that in fact batch normalisation in the discriminator increased mode collapse. More comprehensive and repeated testing would be required. When the discriminator is too strong, label smoothing was shown to help stabilise training. Our results support the use of lower learning rates for DCGANs, as this improved model stability. Although FID was a useful metric, the inability to accurately or objectively measure mode collapse or performance was a persistent problem throughout our experiments. Inspecting images manually is prone to sampling bias. Loss function graphs were the most useful information source in this regard, however not directly transferable to experiments when different loss functions are implemented.

References

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. arXiv:1701.07875.

Goodfellow, I. (2016). NIPS 2016 Tutorial: Generative Adversarial Networks. arXiv:1701.00160.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. arXiv:1406.2661.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. arXiv:1706.08500.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive Growing of GANs for Improved Quality, Stability, and Variation. arXiv:1710.10196.

Karras, T., Laine, S., and Aila, T. (2018). A Style-Based Generator Architecture for Generative Adversarial Networks. arXiv:1812.04948.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2019). Analyzing and Improving the Image Quality of StyleGAN. arXiv:1912.04958.

Lala, S., Shady, M., Belyaeva, A., and Liu, M. (2018). Evaluation of Mode Collapse in Generative Adversarial Networks. *High Performance Extreme Computing*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. (2017). Are GANs Created Equal? A Large-Scale Study. arXiv:1711.10337.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral Normalization for Generative Adversarial Networks. arXiv:1802.05957.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv:1511.06434.

Tensorflow (2020). Deep Convolutional Generative Adversarial Network.

7 Appendix

7.1 Models

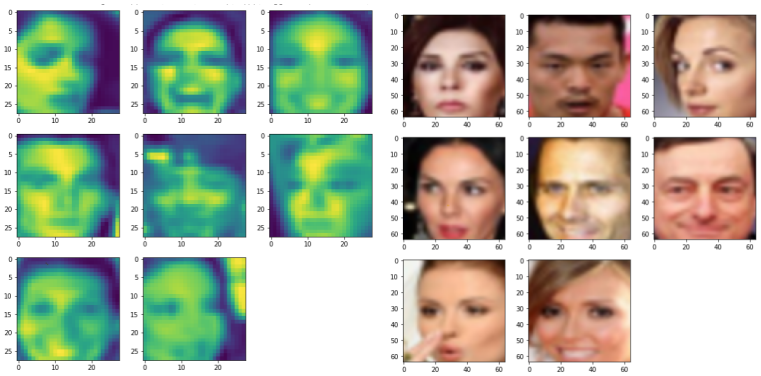
Model	Batch Normalisation (Disc)	Dropout (Disc)	Output (Disc)	Parameters (Disc)
#1	0	1	Linear	200,000
#2	1	0	Linear	200,000
#3	1	0	Sigmoid	200,000
#4	1	1	Sigmoid	200,000
#5	1	1	Linear	200,000
#6	1	1	Linear	1,900,000

Table 2: 28×28×1 Models: Model #1 - https://www.tensorflow.org/tutorials/generative/dcgan Tensorflow (2020)

Model	Label Smoothing	Batch Norm (Disc.)
#A	0.1	Yes
#B	0	Yes
#C	0.1	No

Table 3: $64 \times 64 \times 3$ Models: Base model adapted from Miyato et al. (2018)

7.2 Training Data



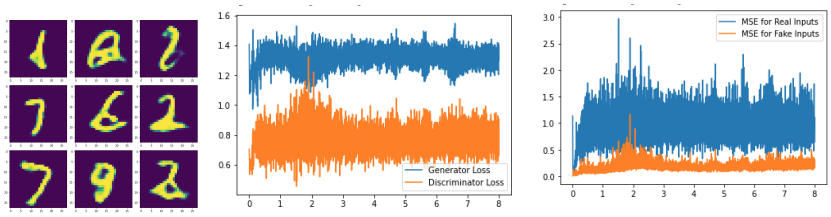
(a) Target Image Set CelebA $28 \times 28 \times 1$

(b) Target Image Set CelebA $64 \times 64 \times 3$

Figure 3: Training Data

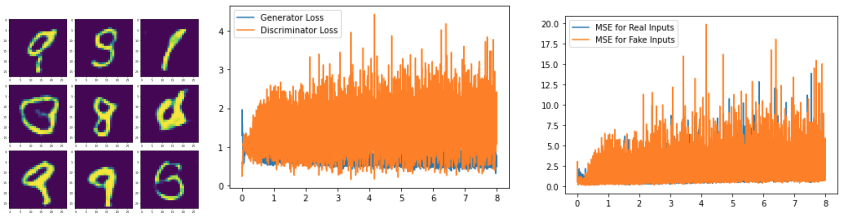
7.3 Visual Results

7.3.1 MNIST $(28 \times 28 \times 1)$



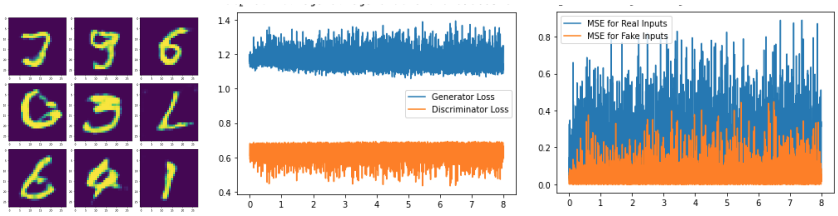
(a) Output, 8 Epochs (b) Non-Saturating Loss (c) Mean Squared Error

Figure 4: Model #1:



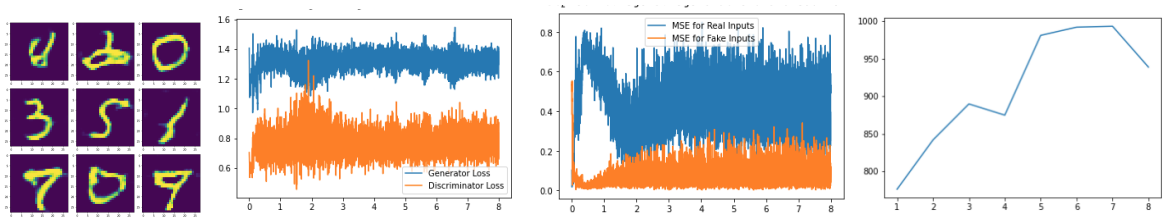
(a) Output, 8 Epochs (b) Non-Saturating Loss (c) Mean Squared Error

Figure 5: Model #2:



(a) Output, 8 Epochs (b) Non-Saturating Loss (c) Mean Squared Error

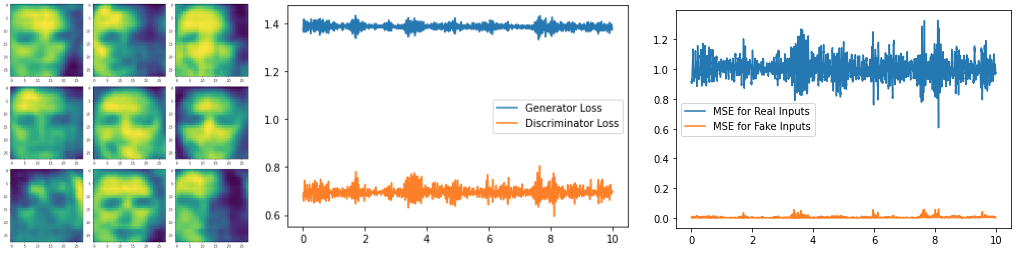
Figure 6: Model #3:



(a) Output, 8 Epochs (b) Mean Squared Error (c) Non-Saturating Loss (d) Frechet Inception Distance

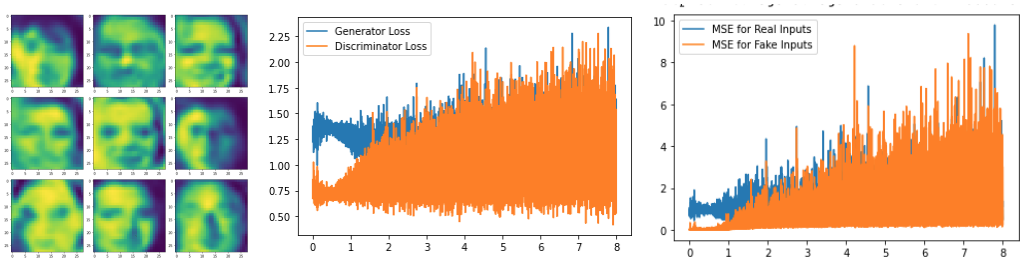
Figure 7: Model #4

7.3.2 CelebA ($28 \times 28 \times 1$)



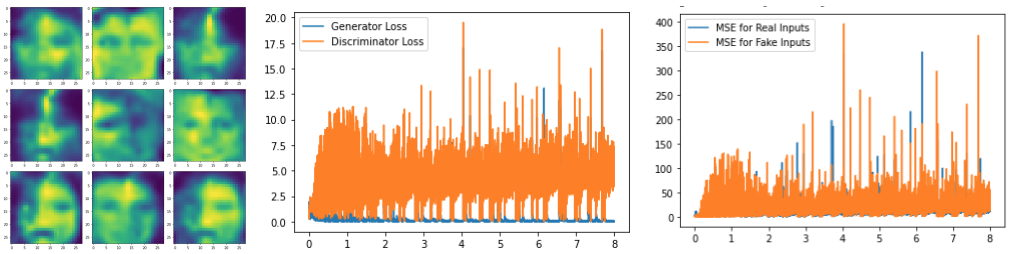
(a) Output, 1 Epochs (b) Non-Saturating Loss (c) Mean Squared Error

Figure 8: Model #0: Poor Learning Rate, Discriminator too good at distinguishing fake from real



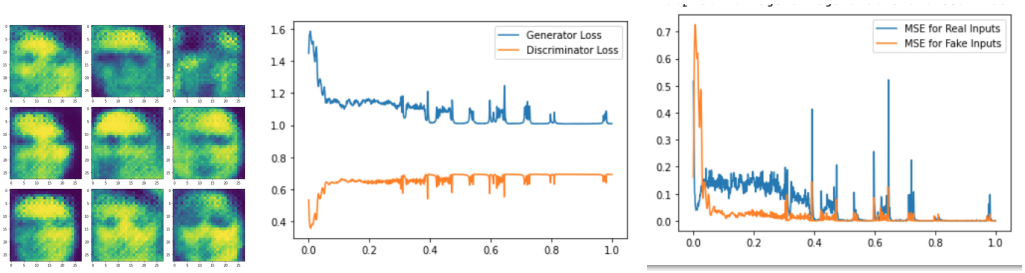
(a) Output, 1 Epochs (b) Non-Saturating Loss (c) Mean Squared Error

Figure 9: Model #1:Slowing the learning rate improved performance but the GAN loss is constantly growing and is unstable



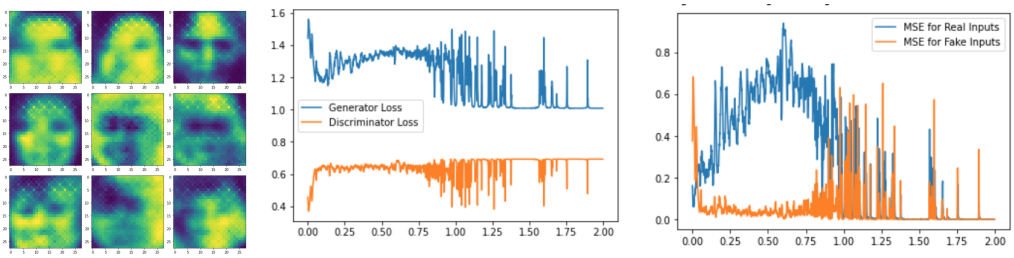
(a) Output, 1 Epochs (b) Non-Saturating Loss (c) Mean Squared Error

Figure 10: Model #2:Batch normalisation causes increased loss of the discriminator function which stabilises after 2 Epochs



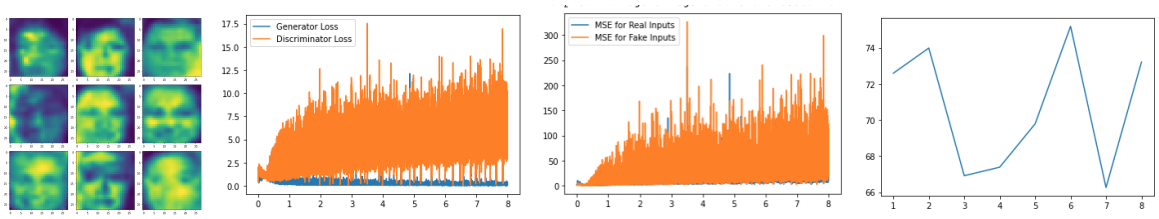
(a) Output, 1 Epochs (b) Non-Saturating Loss (c) Mean Squared Error

Figure 11: Model #3:Bounding the output of the discriminator a sigmoid function improved performance too much



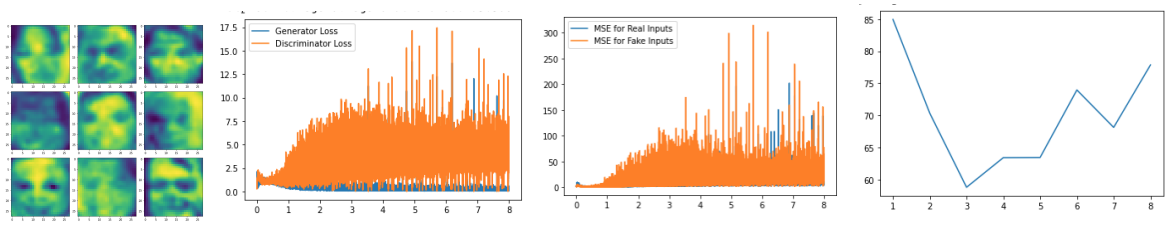
(a) Output, 2 Epochs (b) Non-Saturating Loss (c) Mean Squared Error

Figure 12: Model #4: Adding Dropout delayed the excessive performance of the discriminator.



(a) Output, 8 Epochs (b) Non-Saturating Loss (c) Mean Squared Error (d) Frechet Inception Distance

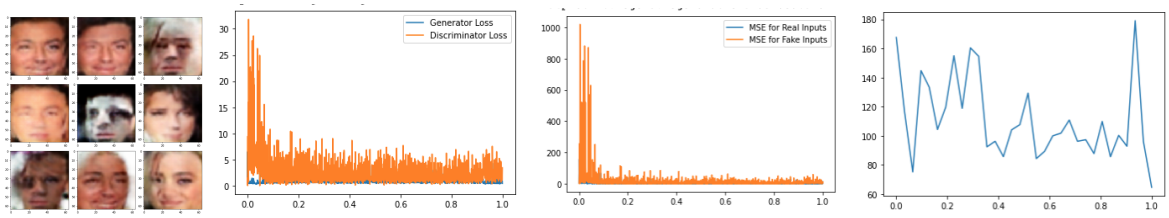
Figure 13: Model #5:Returning to a linear activation function increased variation likely caused the output to be unstable



(a) Output, 8 Epochs (b) Non-Saturating Loss (c) Mean Squared Error (d) Frechet Inception Distance

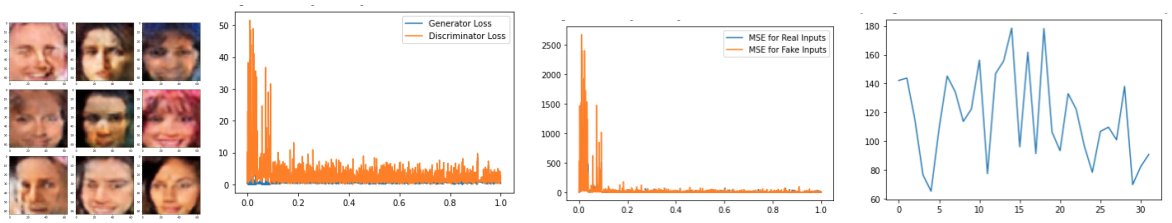
Figure 14: Model #6 Increasing the number of parameters and depth of the discriminator improved stability

7.3.3 CelebA $(64 \times 64 \times 3)$



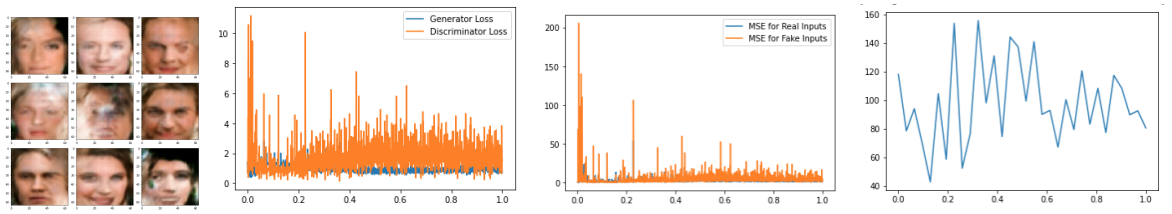
(a) Output, 1 Epoch (b) Non-Saturating Loss (c) Mean Squared Error (d) Mean Squared Error

Figure 15: Model #A Label Smoothing = 0.1, Notice significant mode collapse in output



(a) Output, 1 Epoch (b) Non-Saturating Loss (c) Mean Squared Error (d) Frechet Inception Distance

Figure 16: Model #B Label Smoothing = 0



(a) Output, 1 Epoch (b) Non-Saturating Loss (c) Mean Squared Error (d) Frechet Inception Distance

Figure 17: Model #C with Label Smoothing = 0.1 and no Batch Normalisation