NTNU
Kunnskap for en bedre verden

Department of Computer Science

IDATT2502 - Machine learning

In-depth analysis of measuring DCGAN and LSGAN

Authors:

Tomas Beranek
Eirik Elvestad
Carl Johan Gützkow

November, 2023

Abstract

This report aims to analyze the implementation, and image generation, of two different Generative Adversarial Networks (GAN), Deep Convolutional Generative Adversarial Network (DCGAN), and Least Squares Generative Adversarial Network (LSGAN). Through a detailed analysis, using different metrics such as loss, F1 and inception-score, the study provided a deeper understanding of the capabilities and limitations for each model. The resulting images generated by DCGAN ended up being of high-quality, but consistently diverged with longer training. On the other hand, LSGAN took longer to generate images that resembled the original data set, but with a lower probability of mode collapse. The models were tested using the CelebA data set, which offered an overall extensive understanding of the different GAN architectures. Through these tests and evaluations, the findings contributed to valuable perspectives on the generative image models and their applications.

1 Introduction

In the recent years, there has been rapid development in image generation[17][7]. In the context of computer vision, one can leverage the practically unlimited amount of unlabeled images and videos to learn good intermediate representations, which can then be used on a variety of supervised learning tasks such as image classification[14]. Research to find optimal models resulted in the GAN architecture. Consequently, improvements to the structure further developed on its ideas. Image generation models, backed by immense resources[11], generate samples that are hard to distinguish from pictures taken by regular cameras or illustrations from artists[6].

By assessing the implementations of two different Generative Adversarial Networks, namely *DCGAN* and *LSGAN*, this paper will address this thesis problem:

- How does the selection of different GAN architectures, such as DCGAN and LSGAN, influence the overall performance and stability of Generative Adversarial Networks, and what overarching factors contribute to these effects?

2 Related Work and Theory

The field of unsupervised image generation has had many implementations and improvements over the years. This paper further develops on the ideas of previous relevant literature.

2.1 Understanding GAN

Generative Adversarial Networks were first proposed in 2014. It was a way to use unsupervised learning to generate images by training two neural networks, a generator and a discriminator, in a zero-sum game based on game theory[5].

2.1.1 Generator

The generator uses random noise, known as the latent vector z, transforming it into matrices that resemble the set of training images x. Its goal is to generate data that is indistinguishable for the discriminator. In most implementations, the generator has an activation function after the final layer, to normalize the value of the output[1][3][5].

2.1.2 Discriminator

The discriminator acts as a binary classifier. Its purpose is to determine the probability that an image originates from the training data set, or if the image is the output of the generator. The discriminator strives to maximize the probability of making correct classifications. After the final layer, the original GAN paper makes use of an activation function for normalization of the output[5].

2.2 The implementation of DCGAN

A year after the publication of GAN, a paper was written about the concept of using Deep Convolution GAN. DCGAN further developed on the ideas of GAN by increasing model stability and faster training. This was achieved by four main factors:

- Fully connected layers outside of the convolution layers were removed
- LeakyReLu activation function was used instead of ReLu in the discriminator
- Batch normalization after hidden convolution layers
- Replacing spatial pooling functions with strided convolutions

There were still issues with the stability over long periods of training[14].

2.3 Improving and measuring the GAN framework

In the aftermath of several different GAN implementations, a paper demonstrating how to measure GAN performance emerged. One of the proposed metrics was inception score, measuring the quality and diversity among generated images[15]. This is today one of the most common evaluation metrics for GANs[2].

2.4 The implementation of LSGAN

Following the publication of DCGAN, a new paper was written on the usage of least square loss. Compared to DCGAN, Least Squared Generative Adversarial Network (LSGAN) replaces the BCE loss function and sigmoid activation with the mean square loss function. Using MSE, LSGAN tries to prevent the mode collapse problem by making the generator produce more diverse samples with a higher image resolution. Additionally, it removes the need for an activation function, as the network autonomously learns to regress to 0 and 1, using the penalty system of MSE. It penalizes the generator for samples that are far away from the real data distribution[10]. LSGAN therefore provides more stable training[10].

2.5 F1-score

F1-score is a measure of model accuracy on binary classification systems. It is defined as the harmonic mean of the model's precision and recall[18]. See table 1 for the relationship between the values described in the equations below.

	Predicted Negative	Predicted Positive
Actual Negative	True Negative	False Positive
Actual Positive	False Negative	True Positive

Table 1: Relationship between predicted values and the actual value

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

2.6 Inception Score

IS evaluates the quality and diversity between images, using a pre-trained model which tries to label given images. By definition, inception score does not consider real images at all, and so cannot measure how well the generator approximates the real distribution[15]. This leads to the inception score having its own limitations by being primarily used for models trained on ImageNet or similar data sets[15][2].

$$\text{IS} = \exp\left(\mathbb{E}_x \left[D_{KL}\left(P(y|x) \,||\, P(y)\right)\right]\right)$$

3 Methods

Analyzing the performance of DCGAN and LSGAN requires thorough testing. This section describes the process of evaluating the model's training with measurements and how the models were structured. The CelebA data set, consisting of 200 thousand faces[9], was used to benchmark model performance. It was used because of its size and its use in Alec Radfords DCGAN paper[14].

3.1 DCGAN Implementation

The architectural configuration of the generator $G(z)$ and discriminator $D(x)$ was structured as explained, and visualized, in the original DCGAN paper[14]. $G(z)$ was built with several layers of convolution transpose, ReLu and batch normalization. The input noise is distributed through two-dimensional transposed convolution layers, where the output is doubled until the desired resolution is obtained. Finally, the output is transformed through a tanh activation function.

Similarly, but the other way around, is $D(x)$ layered with sets of standard convolution layers for down-sampling. Using standard convolution layers lets the network learn its own pooling function. After each layer, except the last, a batch normalization and Leaky ReLu is applied. The Leaky ReLU contributes to training stability and reduces issues such as the vanishing gradient problem. Another advantage is the mitigation of inactive neurons, which is done by introducing a non-zero slope for negative inputs. After the last layer, is the sigmoid activation function for normalization[14].

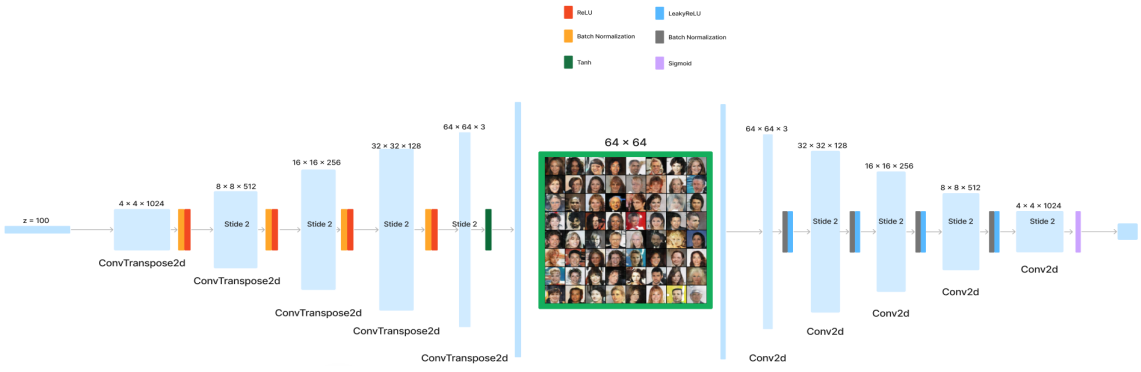

Figure 1: DCGAN Implementation

3.2 Hyperparameters

Several hyper-parameters were chosen based on DCGAN paper[14] and PyTorch documentation[13]. Parameters, such as image size and input/output size, where tuned to optimize the performance of the models. Table 2 shows the default values used when training the neural networks.

batch-size	learning rate	beta1	$D(x)$ input size	$G(z)$ output size	noise size
128	0.0002	0.5	64 & 32	64 & 32	100

Table 2: Default values for some of the hyperparameters

3.3 Measurements

To understand how well the models improved with training, several measurement strategies were implemented. Loss, F1 and Inception score were calculated several times for each epoch.

3.4 LSGAN

The architecture of the LSGAN closely followed the framework in the LSGAN paper[10]. Both the generator and discriminator adopted the same structures as for the DCGAN[13][14] with the exception of the Sigmoid normalization function for the discriminator. This function was replaced with the mean squared error (MSE) loss function as detailed in the research paper.

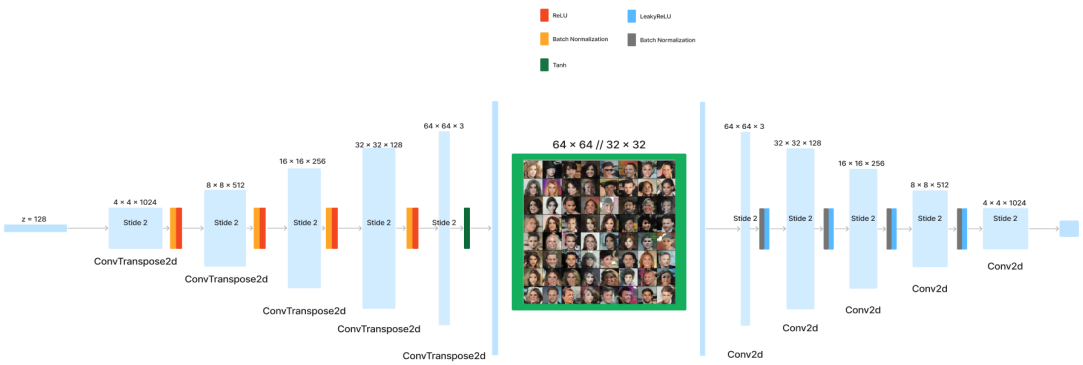

Figure 2: LSGAN Implementation

4 Results

Following the implementation of DCGAN and LSGAN as described in Section 3, the following results were found.

4.1 DCGAN

4.1.1 Loss

Charting the loss of several runs with DCGAN resulted in an interesting pattern. Initially high, the loss slowly decreased before often diverging. Figure 3 illustrates the change in loss between iterations, showcasing variations with an underlying consistency over time.

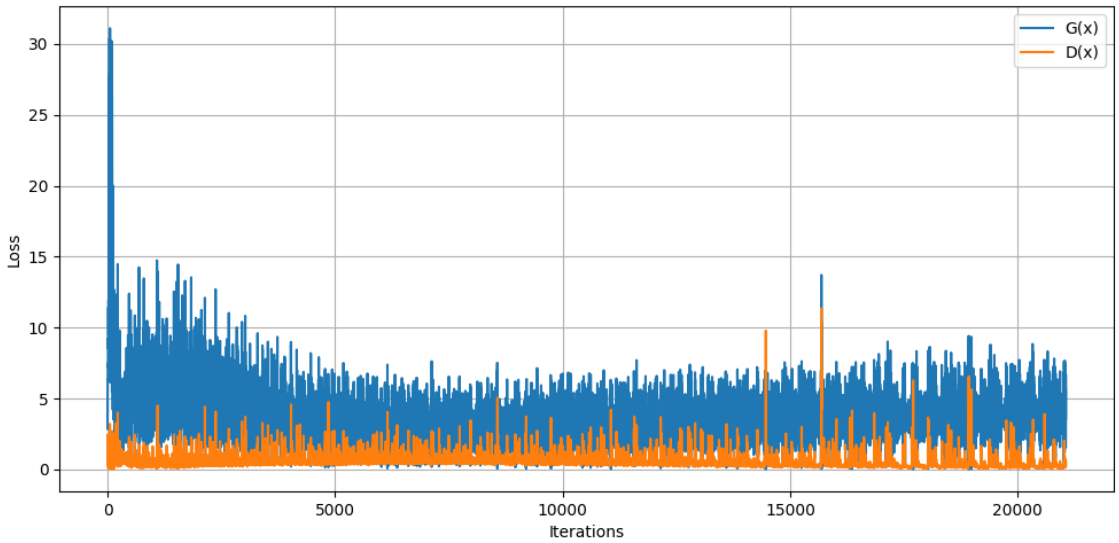

Figure 3: Chart of loss from epoch 0 to epoch 30

4.1.2 F1 Score

The F1 score, plotted in Figure 4, exhibits a steady decrease before it increases on average. It follows the loss graph charted above it. Figure 5 charts the f1 score with recall and precision as well. Notice that the calculation of F1 from the precision and recall does not always comply with the F1 score on the chart. This is a result of an average rolling window. In the figure, recall is consistently higher than precision.

4.1.3 Inception Score

In Figure 6, the mean and standard deviation of the inception score of the generated images are plotted over 70 epochs. Except for the start, there is not much of an increase or decrease.

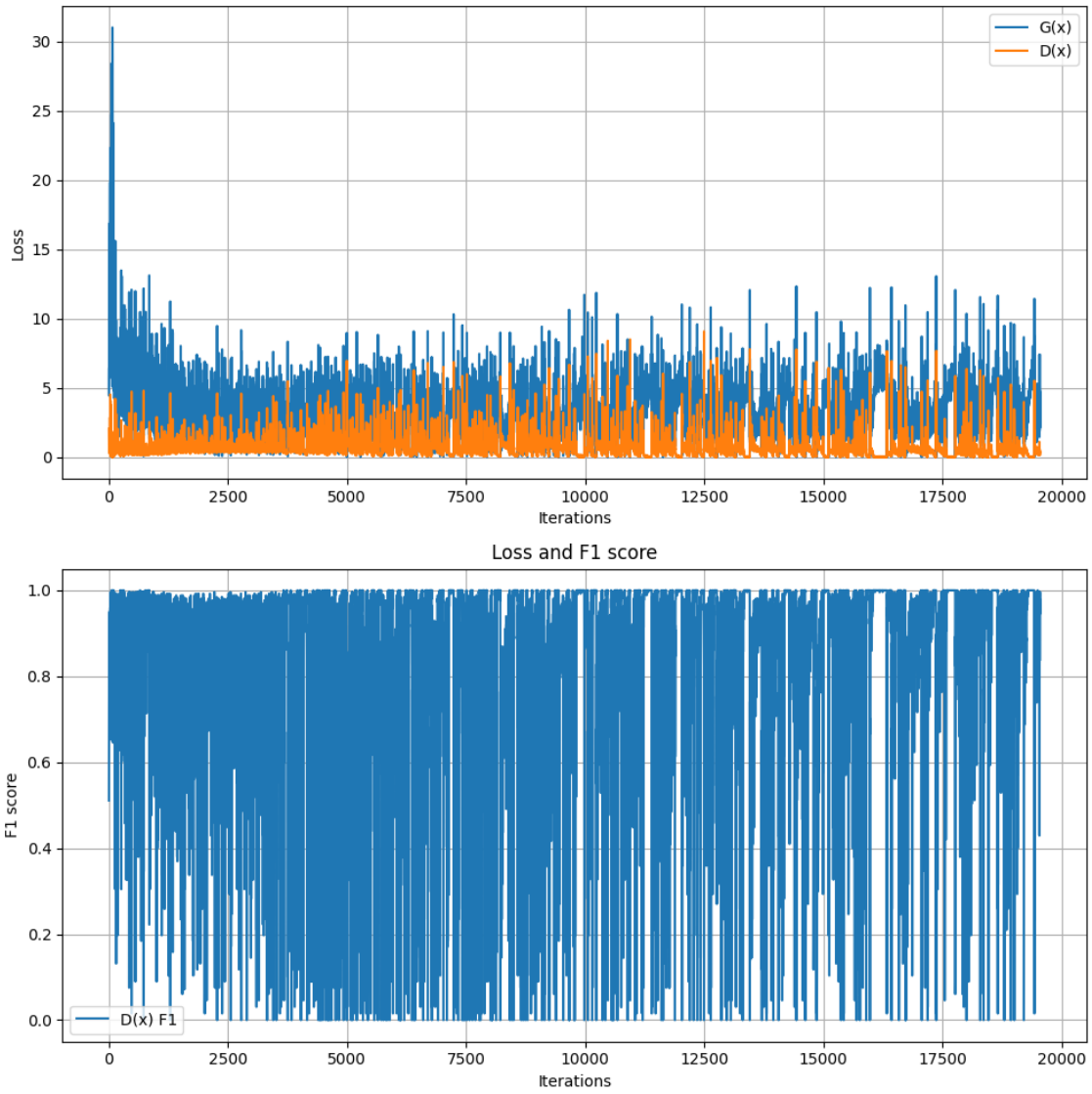

Figure 4: F1 score plotted over 30 epochs

4.1.4 Image Quality

Figure 7 visualizes three stages in the training process, with images generated at epochs 1, 25, and 58. Notably, mode collapse often occurred between 50 and 70 epochs. Additionally, Figure 8 compares sample images from the data set with generated images after epoch 25.

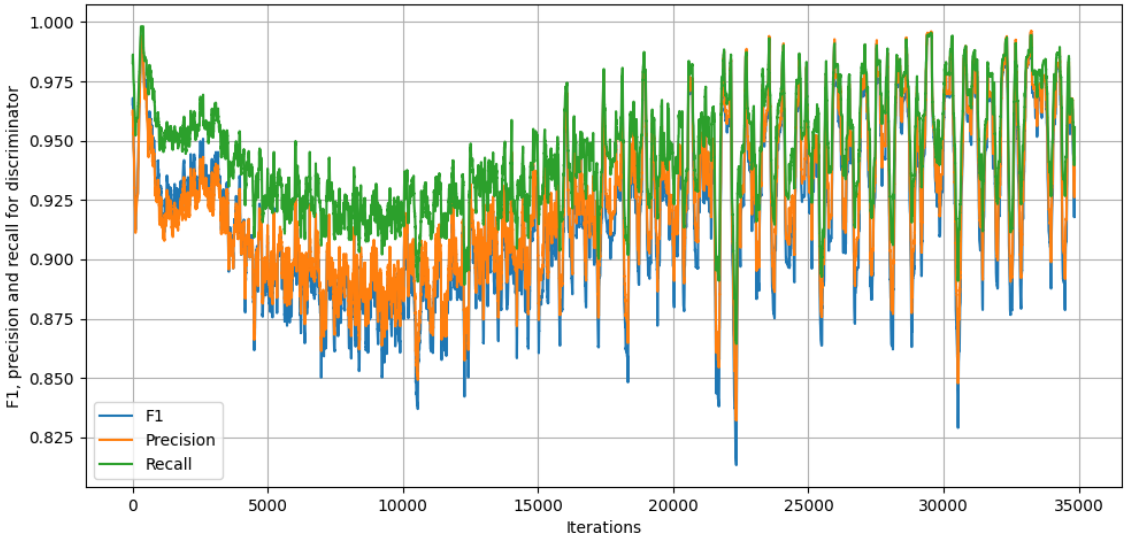

Figure 5: F1 score plotted with recall and precision for a run with DCGAN over 50 epochs

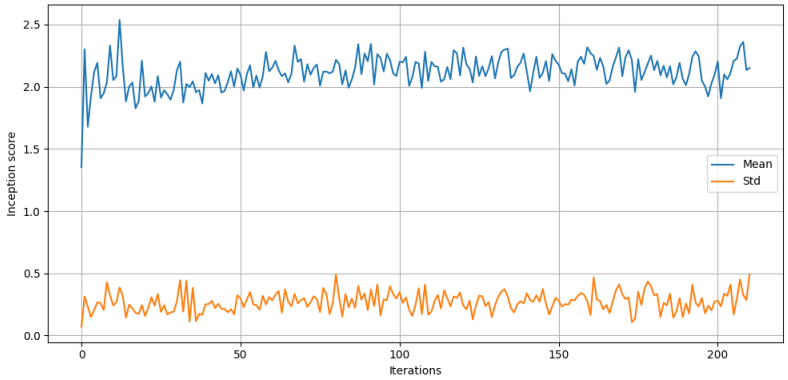

Figure 6: Inception score plotted for DCGAN over 70 epochs

4.2 LSGAN

4.2.1 Image Quality

Training LSGAN did not significantly improve overall results. The LSGAN on the CelebA data set produced high-resolution images with more details but deviated from the original data set. The images suffered from decreased diversity. Convergence rates varied on 32x32 and 64x64 feature map sizes, as depicted in figures 9 and 10.

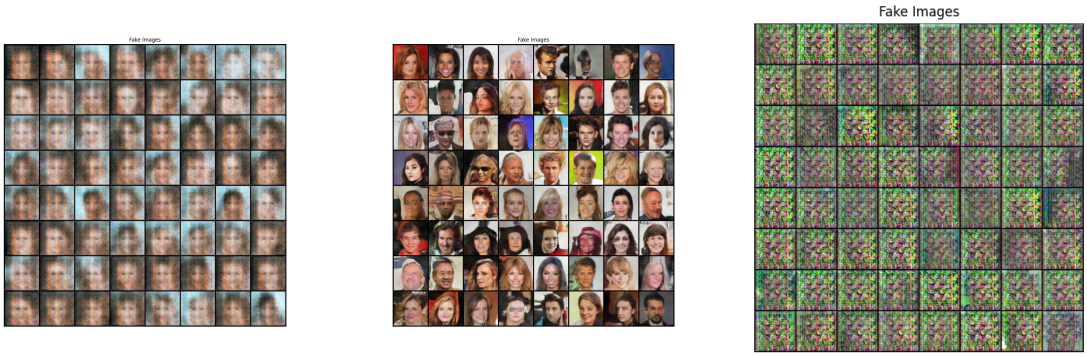

Figure 7: Three images in the process training at epochs 1, 25, and 58

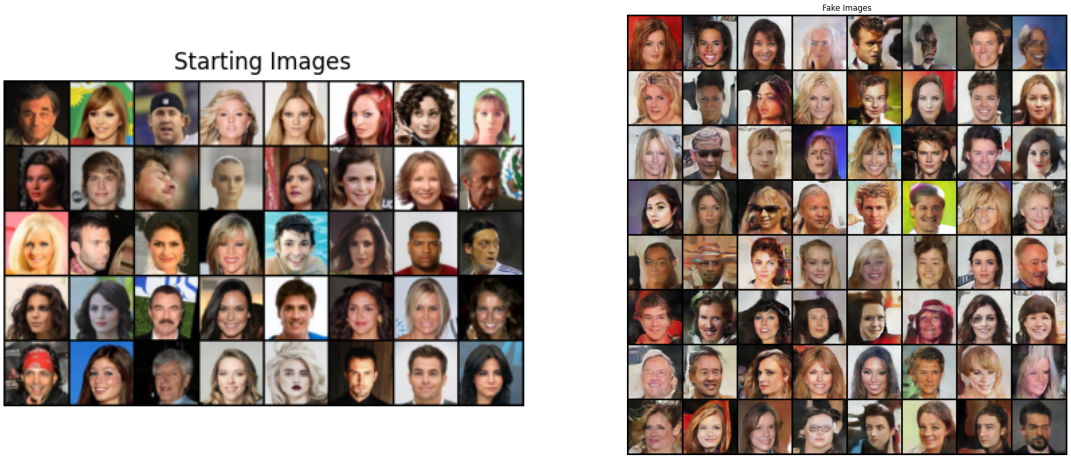

Figure 8: Comparison of images from the data set on the left with generated images from dcgan after 25 epochs on the right

4.2.2 Loss

The biggest observed difference between LSGAN and DCGAN is that the mode collapse appeared less frequently for LSGAN. Figure 12 charts the loss of LSGAN which starts with a high loss, but falls rapidly to a stable value. Both LSGAN and DCGAN has a reduction in loss for both neural networks at the start. Both the generator and discriminator of LSGAN has a prolonged decline. In contrast, the combined loss of DCGAN increases after several epochs as visualized in figure 3.

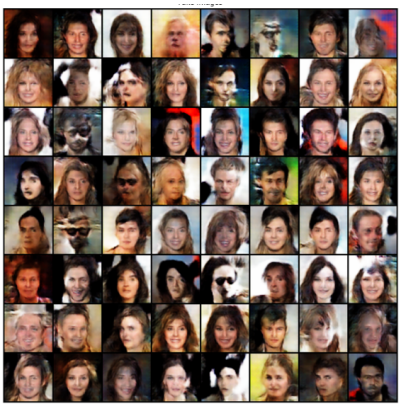

Figure 9: Comparison LSGAN (left) and DCGAN (right) on 32x32 and epoch 8

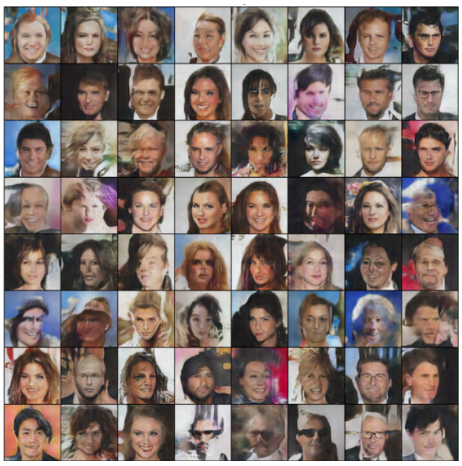

Figure 10: Comparison LSGAN (left) and DCGAN (right) on 64x64 and epoch 15

4.2.3 F1 Score

The F1-score plotted in Figure 12 presents oscillating values (high and low), similar to the DCGAN's f1-score result. In slight contrast to DCGAN, the overall score increases until it drops, before another steady incline. The increment is steeper than for DCGAN.

4.2.4 Inception Score

Figure 13 has plotted the inception score with standard deviation. The results are nearly identical to the inception scores measured for DCGAN showcased in Figure 6. DCGAN's scores are slightly higher, which could indicate better diversity or quality.

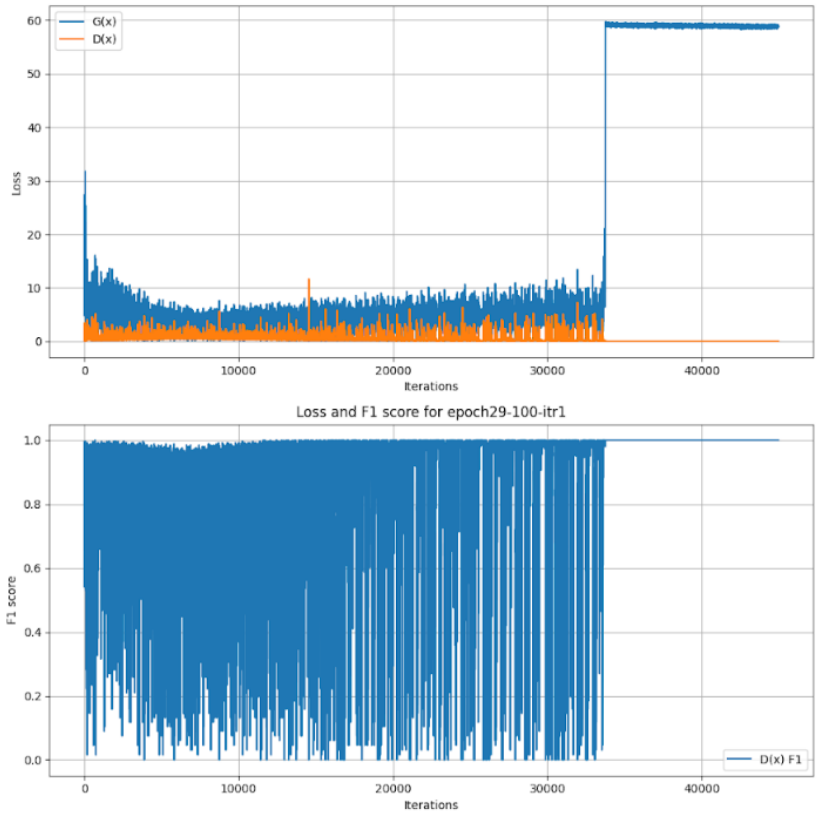

Figure 11: DCGAN mode collapse

5 Discussion

The similarly structured models DCGAN and LSGAN produced images with multiple similar qualities. Prominent differences were mostly apparent after many epochs. The use of MSE instead BCE for calculating loss, shaped a stable model less susceptible to mode collapse. Reviewing the images with human eyes demonstrated the expected result. Images from DCGAN's generator created images more diverse and akin to the original data set. "The disadvantage of LSGAN is that excessive penalties for outliers lead to reduced sample diversity"[4]. The metrics might explain this behavior.

5.1 Loss

While there is a difference in loss over time, it alone cannot be used to justify the difference in image quality. The spectrum of loss values is innate to the model structure and cannot be directly compared[12]. Instead, a comparison can be done by observing how LSGAN has a continuous reduction in loss while DCGAN has an immediate decline before gradually increasing. This could be the result of DCGAN's discriminator learning faster than its generator, while LSGAN kept

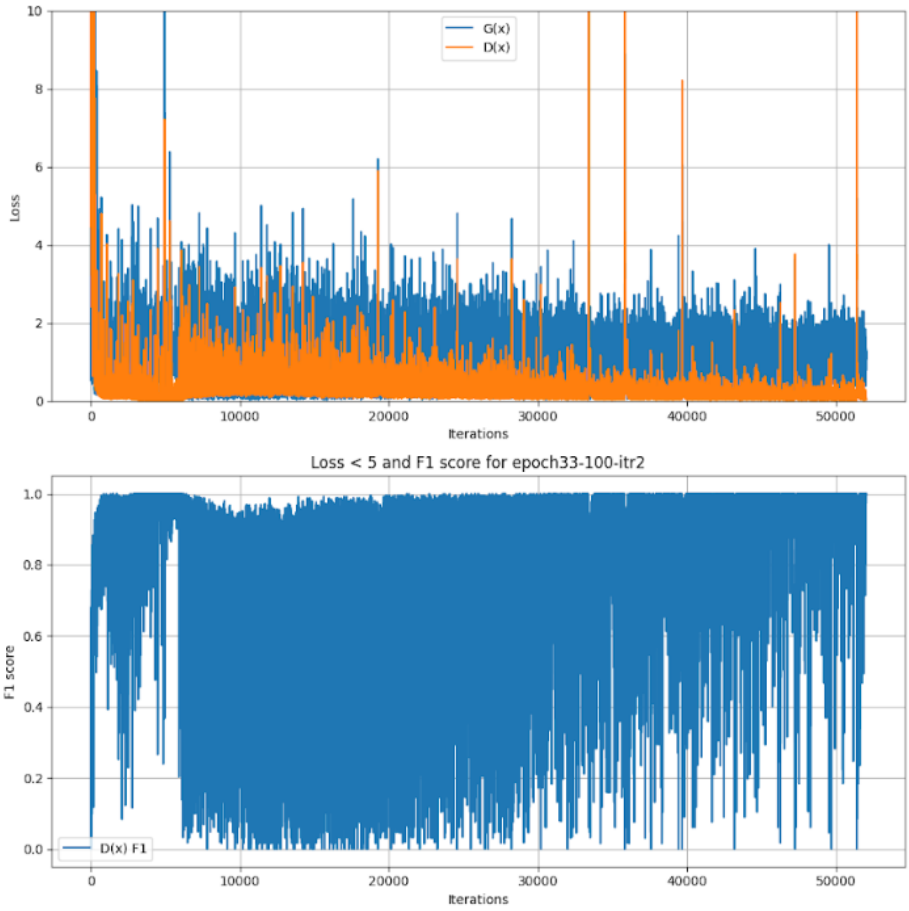

Figure 12: LSGAN f1-score and loss

equilibrium between them for longer. As a result, LSGAN would be more stable over time which aligns with results of less frequent mode collapse.

5.2 Inception

The similarity in inception scores could provide a deeper understanding of the models' similarities. The goal of the inception score was not to measure the performance of the models individually, but rather to compare the two. The resulting score was not expected to reach a score of models trained on ImageNet[2]. However, the scores are so similar they do not illustrate any significant differences in diversity or image distinction.

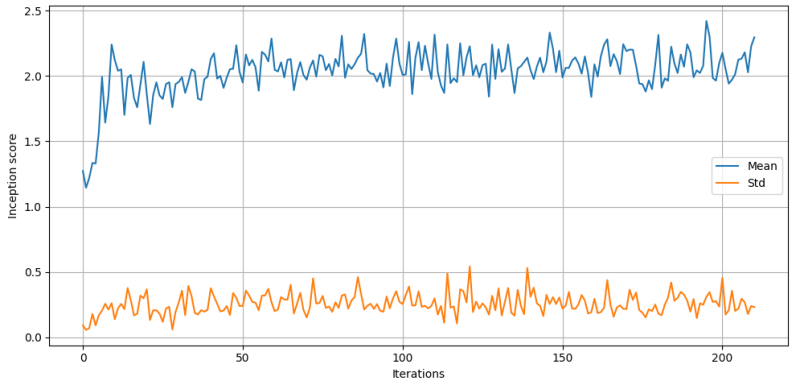

Figure 13: Inception score plotted for LSGAN every 250th iteration over 70 epochs

5.3 F1

Comparing F1 scores for the two models reveals how the image quality improved over time. DCGAN was early on able to produce images that deceived the discriminator. In contrast, the time to produce higher quality images was greater for LSGAN, which can be seen on its F1 score shifting to a lower average after a considerable amount of iterations. As the generators for the two models were equal, the F1 score was seen as fit to display differences in the two discriminators.

Although plotted precision and recall follows each other, there is a notable difference that should be expected for the discriminator. A lower precision would correspond to a higher value of false positives. The discriminator is therefore shown to have a tendency to more often predict that an image is from the real data set. As the loss continuously decreases, this strategy appears to be beneficial for the discriminator.

5.4 Longer training

Would training for longer periods improve the image quality? The result of mode collapse for DCGAN in addition with its slow but steady increase in loss (for the generator), suggests that longer training doesn't solve any issues past a certain point. For LSGAN however, training over long periods of time could help generate images more akin to the data set. The observed trajectory of the LSGAN loss graph suggests a convergence trend towards a diminished value, in contrast to its initial loss values. This implies that the LSGAN has learned its activation function through the utilization of the MSE loss function.

6 Conclusion and Further Work

In conclusion, the analysis of DCGAN and LSGAN provided valuable insights into their respective performances in image generation. Both GANs exhibited different strengths and limitations based on their differences in architecture and training model.

The result affirm the success of the DCGAN implementation, as evidenced by generated images resembling the original data set. Building upon the fundamentals of DCGAN, the LSGAN further

demonstrated advancements and higher image resolution. Even though the generated images ended up having better resolution, new limitations were also introduced. The process of the implementation and exploring additional applications has been highly educational and captivating.

Replacing the sigmoid activation function, and the BCE loss function, with MSE loss improved the overall training stability. However, this commitment comes with the trade-off of requiring longer training to produce images similar to the original data set.

To further expand on our work, more data sets would be considered. Benchmarking with ImageNET or Cifar100 would yield more nuanced results in inception score[2]. An example of this can be seen in appendix B. Another task would be to examine if the models could successfully overcome mode collapse through extended training period with a higher number of epochs. Furthermore, more metrics could be used to spot other differences in quality and performance. This includes Fréchet Inception Distane (FID), sliced Wasserstein distance (SWD), and Multi-Scale Structural Similarity (MS-SSIM)[7].

Acknowledgments:
We would like to thank Nicolai Sivesind for his guidance during the project. Finally we extend a special thanks to our professors Ole Christian Eidheim, Donn Morrison and Jonathan Jørgensen for their guidance throughout the subject.

References

[1] *Activation functions in Neural Networks*. [Online; accessed 21. Nov. 2023]. Feb. 17, 2023. URL: https://www.geeksforgeeks.org/activation-functions-neural-networks (visited on 11/21/2023).

[2] Shane Barratt and Rishi Sharma. "A Note on the Inception Score". In: *arXiv* (Jan. 2018). DOI: 10.48550/arXiv.1801.01973. eprint: 1801.01973.

[3] Zack Brodtman. "The Importance and Reasoning behind Activation Functions". In: *Medium* (Jan. 4, 2022). URL: https://towardsdatascience.com/the-importance-and-reasoning-behind-activation-functions-4dc00e74db41 (visited on 11/21/2023).

[4] Christine Dewi et al. "Various Generative Adversarial Networks Model for Synthetic Prohibitory Sign Image Generation". In: *Appl. Sci.* 11.7 (Mar. 24, 2021), p. 2913. ISSN: 2076-3417. DOI: 10.3390/app11072913. (Visited on 11/20/2023).

[5] Ian J. Goodfellow et al. "Generative Adversarial Networks". In: *arXiv* (June 2014). DOI: 10.48550/arXiv.1406.2661. eprint: 1406.2661.

[6] Tero Karras et al. "Analyzing and Improving the Image Quality of StyleGAN". In: *arXiv* (Dec. 2019). DOI: 10.48550/arXiv.1912.04958. eprint: 1912.04958.

[7] Tero Karras et al. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *arXiv* (Oct. 2017). DOI: 10.48550/arXiv.1710.10196. eprint: 1710.10196.

[8] Uttam Kumar. "Generative Adversarial Networks (GANs) or Transfer Learning". In: *Medium* (Feb. 2023). ISSN: 4696-3419. URL: https://kr-uttam.medium.com/generative-adversarial-networks-gans-or-transfer-learning-b469634e1f9d.

[9] Ziwei Liu et al. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

[10] Xudong Mao et al. "Least Squares Generative Adversarial Networks". In: *arXiv* (Nov. 2016). DOI: 10.48550/arXiv.1611.04076. eprint: 1611.04076.

[11] Cade Metz. "OpenAI in Talks for Deal That Would Value Company at $ 80 Billion". In: *N.Y. Times* (Oct. 2023). ISSN: 0362-4331. URL: https://www.nytimes.com/2023/10/20/technology/openai-artifical-intelligence-value.html.

[12] Mathew Mithra Noel et al. "Alternate Loss Functions for Classification and Robust Regression Can Improve the Accuracy of Artificial Neural Networks". In: *arXiv* (Mar. 2023). DOI: 10.48550/arXiv.2303.09935. eprint: 2303.09935.

[13] PyTorch. *DCGAN Tutorial — PyTorch Tutorials 2.1.0+cu121 documentation*. [Online; accessed 13. Nov. 2023]. Nov. 11, 2023. URL: https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html (visited on 11/13/2023).

[14] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *arXiv* (Nov. 2015). DOI: 10.48550/arXiv.1511.06434. eprint: 1511.06434.

[15] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. "How good is my GAN?" In: *arXiv* (July 2018). DOI: 10.48550/arXiv.1807.09499. eprint: 1807.09499.

[16] *Transfer Learning for Computer Vision Tutorial — PyTorch Tutorials 2.1.1+cu121 documentation*. [Online; accessed 20. Nov. 2023]. Nov. 16, 2023. URL: https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html (visited on 11/20/2023).

[17] Rocio Vargas, Amir Mosavi, and Ramon Ruiz. “DEEP LEARNING: A REVIEW”. In: *Advances in Intelligent Systems and Computing* 5.2 (June 2017). ISSN: 2194-5357. URL: https://www.researchgate.net/publication/318447392_DEEP_LEARNING_A_REVIEW.

[18] Thomas Wood. “F-Score”. In: *DeepAI* (July 2020). URL: https://deepai.org/machine-learning-glossary-and-terms/f-score.

Appendix A

Transfer learning is a machine learning technique that allows a model trained on one task to be reused for another related task, with little or no additional training[8]. Using transfer learning proves invaluable, as it helps the model reduce the amount of labeled data required(for training), and leads to a performance enhancement of the deep learning models.

The first attempt was to train a DCGAN model on CelebA data set. The weights were then saved and the model retrained on an animated face data set to see the potential of transfer learning[16] (figure 14).

Figure 14: Transfer learning between celeba and animated faces

The model, initially trained on the CelebA data set, extended its generative capabilities to the animation data set. As can be seen in figure 14 the model in its first epoch, suggesting an accelerated convergence, surpasses the model only trained on animated faces.

The observed outcome may also be influenced by the choice of data set. The DCGAN tends to perform better with images that contains more realistic photos, rather than animations. This discrepancy may stem from realistic photos naturally having more noise, which can mask imperfections in generated results. Another possibility behind the result could be that realistic images are less flat and therefore have more depth, resulting in more diverse generated outputs.

Appendix B

Several other data sets were tested on the two models. Figure 15 is an example of this. An interesting take away from the figure is that the inception score has a notable increase over time, even though the model was only trained for a short period. CIFAR10 shares similarities with the ImageNET data set and therefore the calculated inception score is higher.

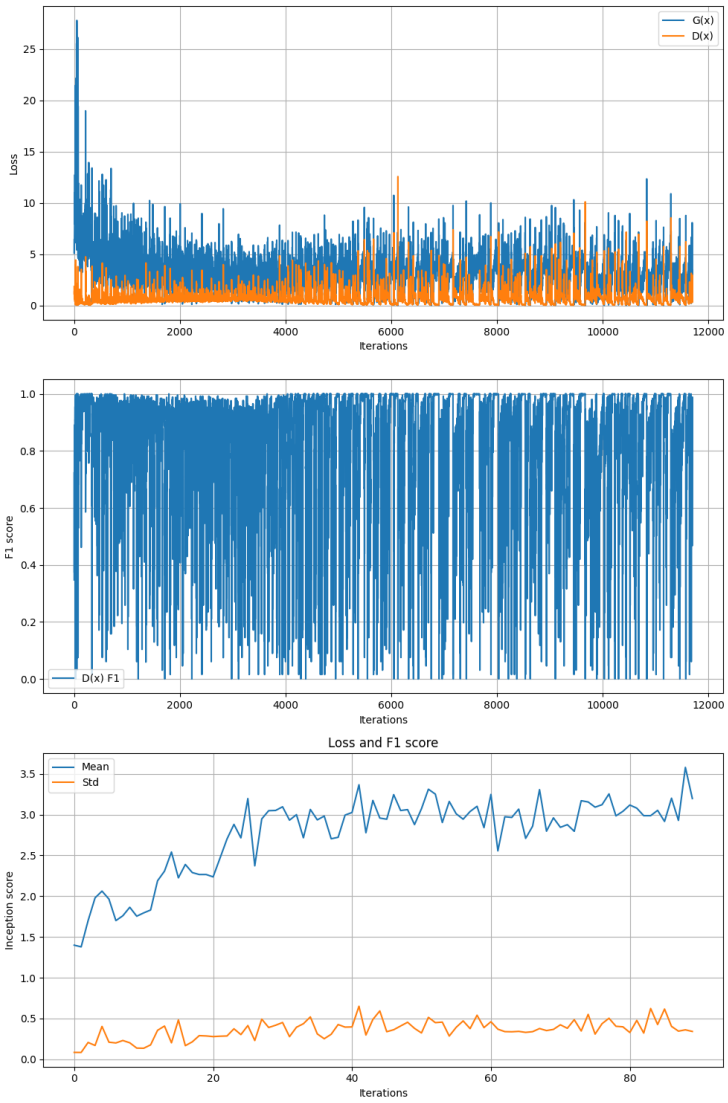

Figure 15: DCGAN trained on CIFAR10. Displays a clear increase in inception score