UNIVERSITÀ DEGLI STUDI DI PADOVA

DIGITAL FORENSICS

GAN IMPLEMENTATION PROJECT REPORT

INSTRUCTORS

SIMONE MILANI

MASSIMO BOLOGNARI

STUDENTS

ADNAN KEREM AKSOY - 2081392

ÖMER CEM TABAR - 2081169

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Abstract: The application of machine learning, deep learning and artificial intelligence techniques in the social world is thriving. Interest in the data changed its course from discriminative approach to generative approach lately for the purpose of creating a synthetic environment. Development of generative models, especially Generative Adversarial Networks (GANs), enlightened the path for the creation of synthetic data, data enhancement and advancing the decision making systems. Data augmentation techniques are significant for the creation of sufficient and large synthetic data to be used in most of the applications in order to avoid possible issues in the real-world settings such as connection loss in the data retrieval, malicious and irregular use of data. In this project, we applied the concept of Deep Convolutional Generative Adversarial Network (DCGAN) and Adversarial AutoEncoder (AAE) to perform a data augmentation on the Anime Faces dataset obtained from kaggle for the purpose of generating real-like synthetic data. We also validated both models' performance in terms of three different metrics: convergence of the modals according to losses, Inception Score (IS) and Frechet Inception Distance (FID) score. The results that are obtained from these metrics demonstrate how synthetic data is created through well structured DCGAN and AAE in comparison with the real dataset.

1. INTRODUCTION

By the emergence of Artificial Intelligence, generative models have become one of the most prominent topics starting from the 1950s until now. After the arrival of the deep learning methods, significant improvements were constructed on the generative models in order to avoid and overcome the issues that faced up such as long durations of training phases due to lack of parallelism, space and time complexity of the approaches and low quality on the generated outputs. Some of the relatively recent models that you are going to see in this report are Deep Convolutional Generative Adversarial Networks (DCGANs) and Adversarial AutoEncoders (AAEs) in order to overcome the drawbacks that are stated above. GANs are highly considered when producing the best quality synthetic images on top of it Deep Convolutional GANs are a mostly used approach for detailed and high quality image generation. That is why the combination GAN notion with Variational AutoEncoders (VAEs) architecture under the terminology of AAEs is also considered promising even if VAEs are not considered consistent. That is why during the course of the report and in the analysis part both performances of the models are compared in order to understand whether the quality of the generative process is sufficient for our purpose of generating fake anime character faces and what further developments can be applied for further research.

Within this perspective we proposed the optimal implementations of both models in which their performances are analyzed with different metrics in terms of convergence of the generator and discriminator, and quality of the image generation in comparison with the original anime faces data. Due to promising results obtained, implemented models are sufficiently capable of generating fake anime character faces of the relatively same quality as the original data which makes them also promising models for other image generations for other projects.

Given the historical background and importance of the recent models that overcomes the issues of the real-world settings, such as small dataset, malicious or irregular data pollution and underdeveloped decision making systems, more models and system can be developed in order to discriminate the authentic and synthetic from each other and provide a better solution for filling the data shortage with high quality data generated from the original ones.

The remainder of the report is structured as follows: Section 2 reports the implementation of DCGAN and AAE models by giving the details about the structure of models, iterative process for the model training and optimal solutions that are implemented for the models. Section 3 demonstrates generated anime faces from original models, shows the iterative quality increase in the outputs and convergence between the generator and discriminator models. Section 4 contains the analysis on the quality of the output images by considering Inception Score (IS) and Frechet Inception Distance (FID) score. Finally, Section 5 concludes the report.

2. IMPLEMENTATION DETAILS

In both networks we have used the anime faces dataset from Kaggle (https://www.kaggle.com/datasets/soumikrakshit/anime-faces). We have read the images and appended them to a list as numpy arrays. And turned the numpy array list to a dataloader.

DCGAN Implementation and Training

For the implementation of DCGAN, multiple activation functions have been used. First of all, the LeakyReLu Activation function has been tried to use on both the Generator's and Discriminators input layers and hidden layers. At the last layer of the Generator, softmax is used, and on both the Generator and Discriminator learning rate is adjusted as 10^(-3). With these parameters and activation functions, our network trained to some point, but the results were not what we expected.

The network kept training with more epochs, but the results of hair or faces were still incomplete. With these parameters, the learning rate is adjusted to a lower level in order to make results more detailed. But the results didn't change much. So, to improve the Generator's efficiency, Generator must not update its weights if the score is negative. As a result, the input and hidden layer's activation function changed to ReLu on the Generator, and the end layer activation function of the Generator changed to Tanh.

The activation function on the Discriminator has not been changed, but the negative slope on LeakyRelu adjusted inside the Discriminator. Also, to retrain a new model, both of the network's learning rates adjusted to 10^(-3) again. At the end of the training network with these activation functions and these parameters, the results were looking more promising.

DCGAN trained with checkpoint logic. DCGAN trained until the images started to not change much (Vanishing Gradient). When this occurs, the network starts to not learn properly. At this point, the pre-trained model saves each time this occurs. Parameters have been re-adjusted, and the pretrained model loaded again on each step. With each step, the learning rate was lowered to make the network create more detailed images, and the Discriminator's negative slope was lowered to make the Discriminator learn better.

A better Discriminator means better results because the generator will adjust itself according to the Discriminators scoring. Therefore, with each step, the Discriminator keeps learning better, so the Generator can learn from the Discriminator's scores and learn better. We have trained our network in 6 checkpoints, with a total of 400 epochs. 64 to 512 Batch sizes are used in the networks, the best fitting one was 64 as a batch size. Therefore mainly 64 batches were used during the training phases.

Adversarial AutoEncoder Implementation and Training

Adversarial Autoencoders Encoder and Decoder networks were created as Convolutional Networks. The Discriminator network was created as a linear network. The LeakyReLu activation function is used at the input and hidden layers of all networks. The sigmoid activation function is used as the end layer on the Discriminator network, Tanh activation function is used at the end layer of the Decoder.

Even though there were not many neurons inside each network, the training process used too much memory probably because of the end activation functions of the Discriminator and Decoder, and the batch size. The result of training the AAE network showed us the Encoder is learning much quicker than the Discriminator in the first batch. Even if we have 168 batches, the first batch is enough for the Discriminator to converge. At the end of the first epoch, the result of the decoder network is promising, and detailed, on the other hand the Discriminator network's loss converged around the first batch and keeps converging around a 100 loss score.

Adjustments in learning rates and adjustments in negative slopes of each LeakyRelu activation function have been tried to create a balance between the Encoder and Discriminator network losses. To see the balance between the Encoder and Discriminator losses, batch size has been lowered, but at that point, our hardware was getting out of memory. When the network starts to train, the network uses around a total of 19 to 20 GB of memory(8 to 10GB from RAM, 9 to 10GB from Graphics Card).

The convergence of the Discriminator loss is seen, but it was not very detailed. After the end of 10 epochs with AAE training, the zoomed plot of the Discriminator loss shows us the convergence clearly. With the convergence that is seen clearly, and the Decoder network's result that is also seen promising and detailed, the training has been completed.

You can access the pretrain models provided in this google drive link: https://drive.google.com/drive/folders/1rlCIl5CZm_MolS-Uc797QIA7uw3UUVsH?usp=drive_link

3. OBTAINED RESULTS

In this section you can observe the output generated images from both models by making the comparison with raw original image data. Within each batches that are shown below, the corresponding original image data and generated version of data with noise accordingly can be observed clearly. Even though the human eye decision making is considered as biased, the results seem promising.

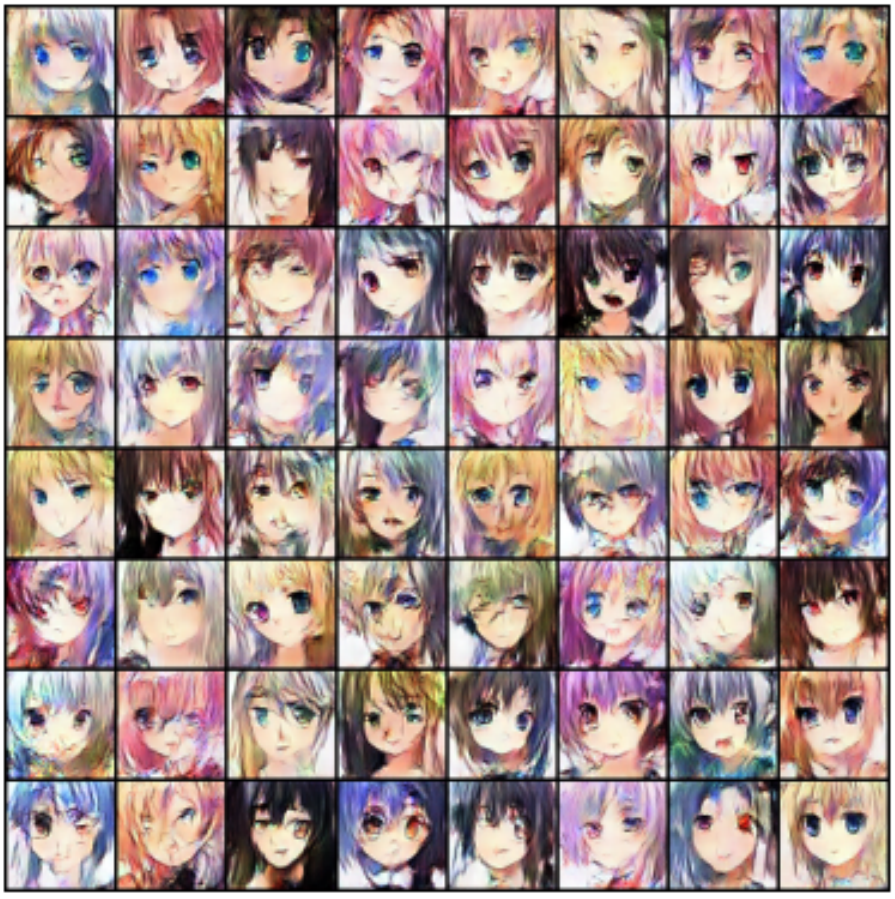

Figure.1 DCGAN 400 Epoch Generated Results

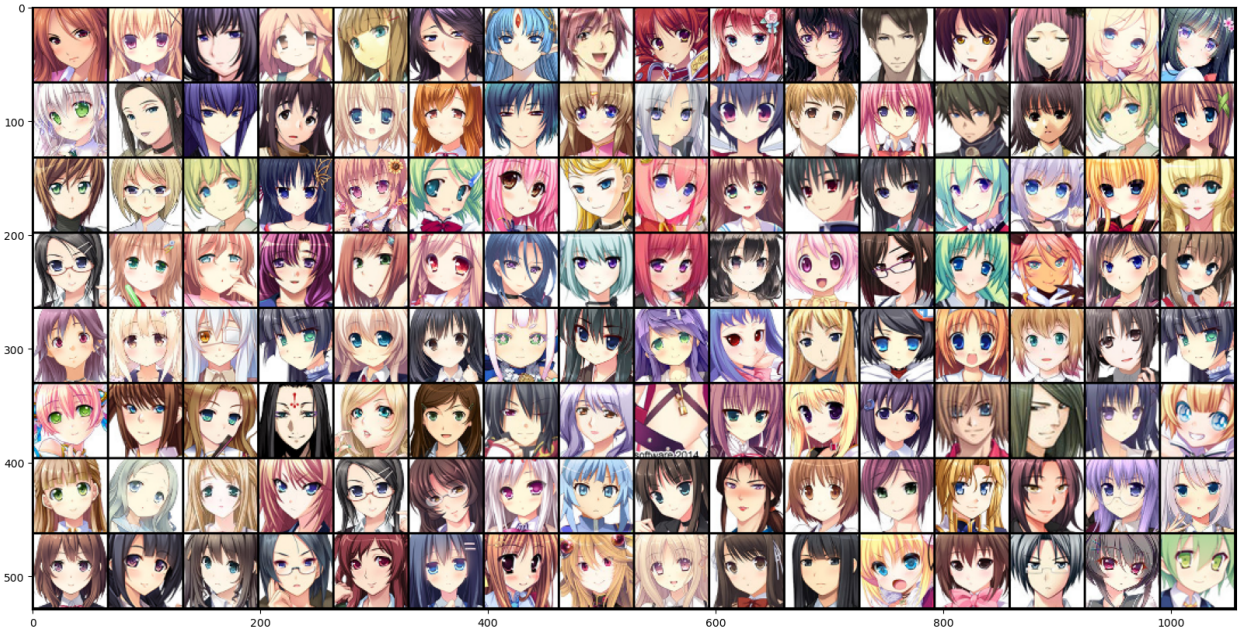

Figure.2 AAE Raw Images Batch Size128

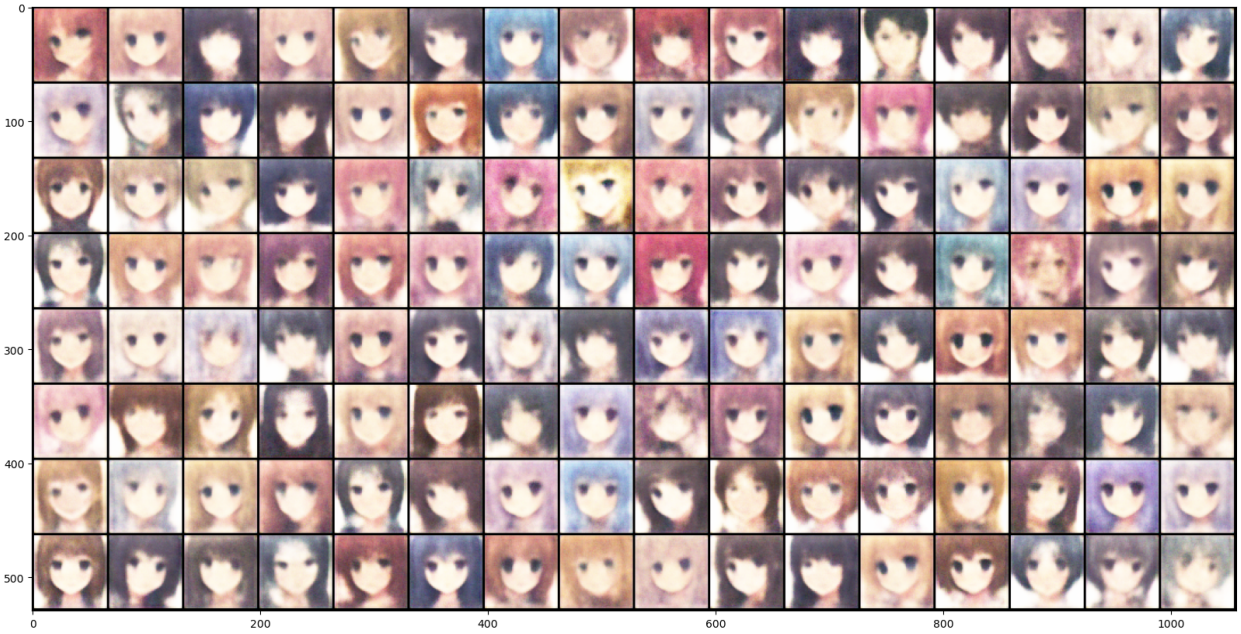

Figure.3 AAE Reconstructed Images Batch Size 128

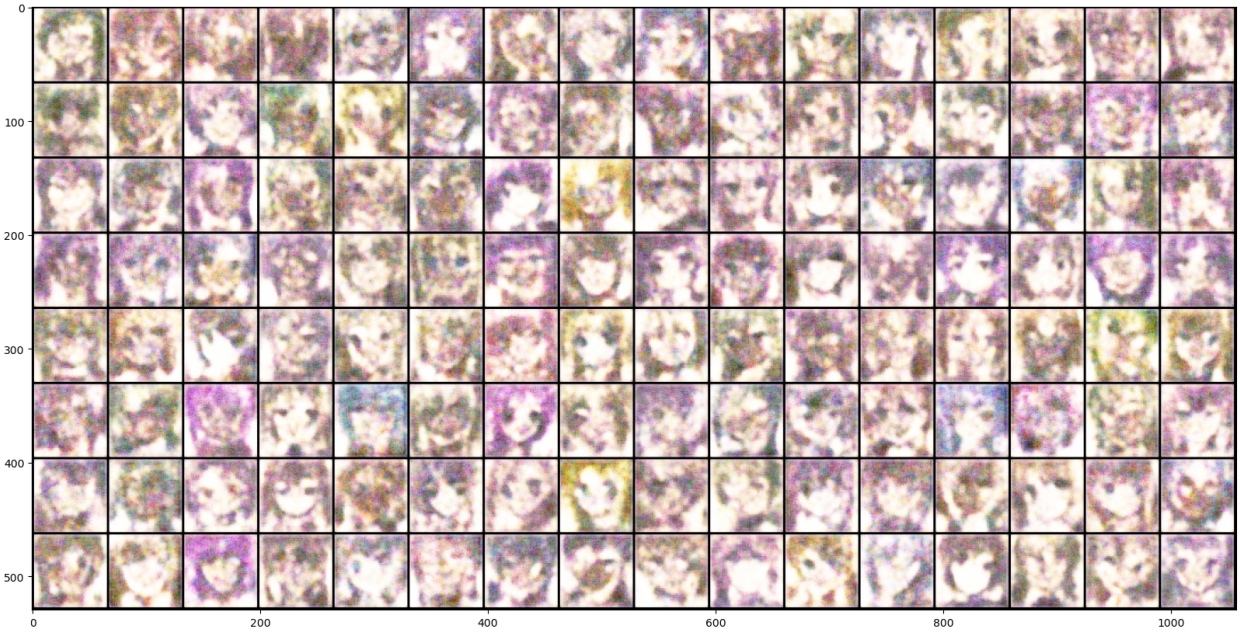

Figure.4 AAE Noise Reconstructed Images Batch Size 128

Convergence of the models that stated above can be observed from the figures below:

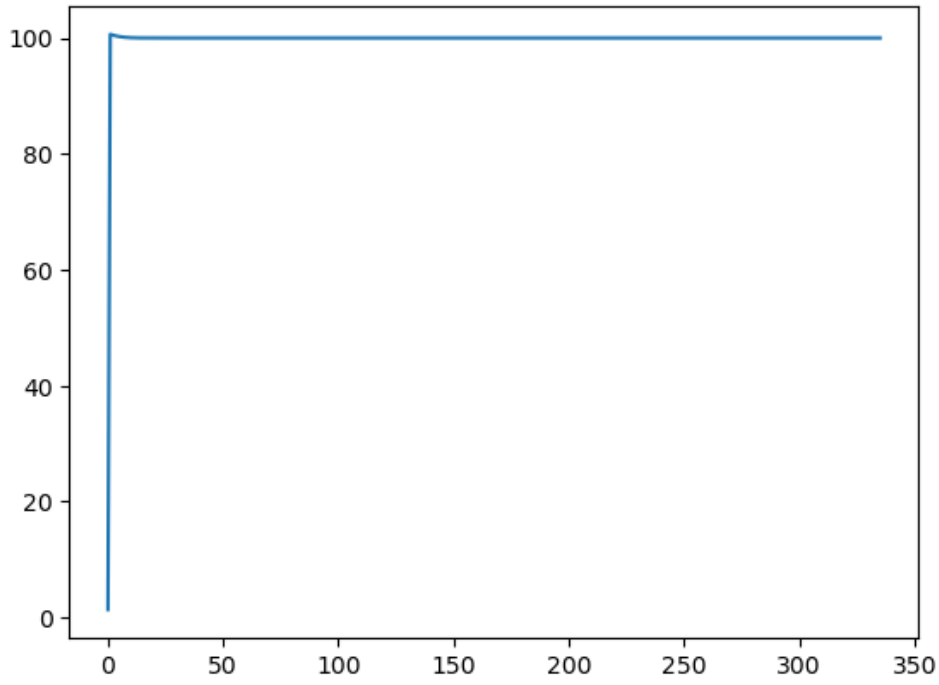

Figure.5 AAE Discriminator Loss Convergence -1

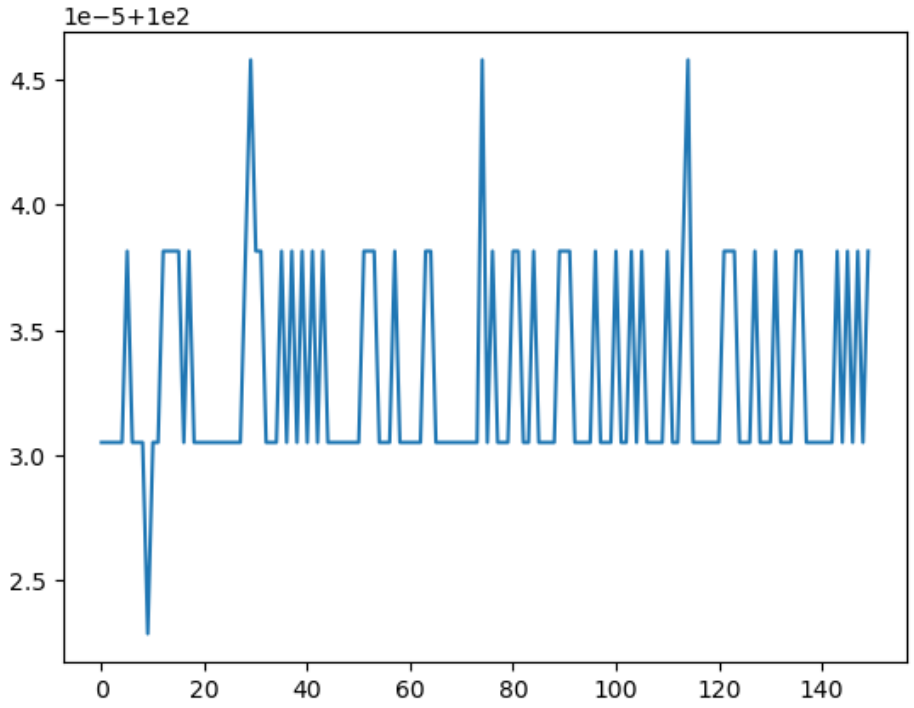

Figure.6 AAE Discriminator Loss Zoomed Convergence -1

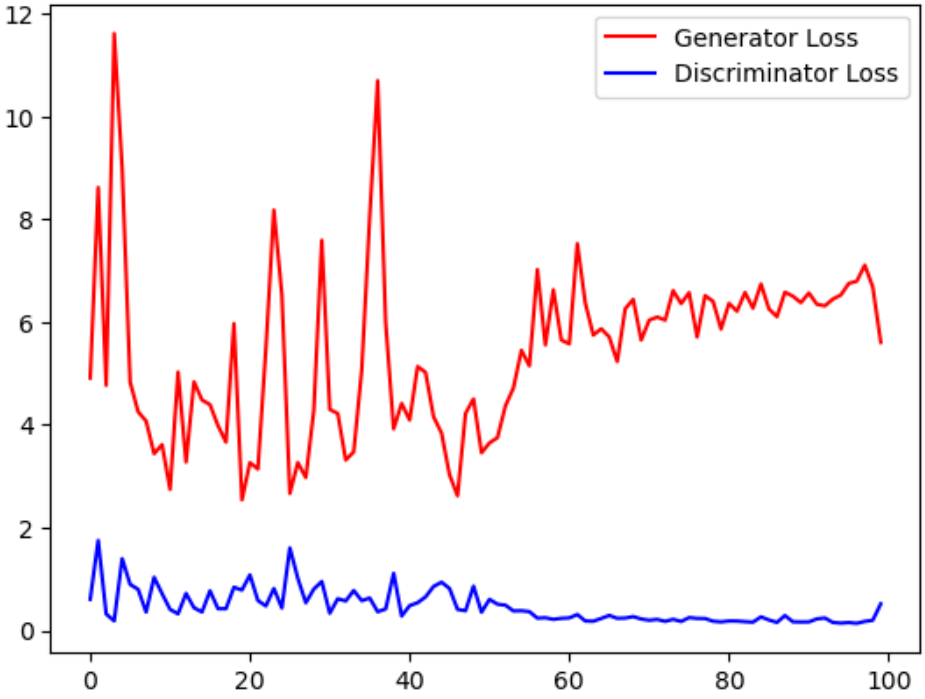

Figure.7 DCGAN Generator Discriminator Loss Convergence

4. ANALYSIS

Even if the generated images are sufficiently observed in high quality and reflect the anime faces originated from our dataset, human discrimination and decision making on the quality of generated version of images is considered biased. That is why evaluation methods and metrics are suggested and widely used in most of the GAN projects in order to satisfy the need for verification of the quality on an objective basis. For this purpose Inception Score (IS) and Frechet Inception Distance (FID) score is calculated for both model performances. In below parts you can follow how the scores obtained and observed iteratively for both models performances.

The Inception Score (IS) is an objective and mathematical metric for evaluating the quality of the generated images that are outputted from our models. Likely, Frechet Inception Distance (FID) score, is also an objective and mathematical metric that calculates the diversity, by considering distance, of the feature vectors calculated from the original and generated images. Both approaches are using a pretrained Inception Model for image classification among 1000 possible classes in order to compare which ensures the subjective human eye bias.

Keras library is used for the pretrained Inception Model v3, since the default implementation of the inception model expects an image in size 299*299*3 as an input, we created our model by removing the top and specifying the input shape as 128*128*3. Then image batches are inputted to the model for the predicted probabilities and other required calculations used in the formulas that are below:

$$\mathrm{IS}(G) = \exp\left(\mathbb{E}_{\mathbf{x}\sim p_g} D_{KL}(p(y|\mathbf{x}) \parallel p(y)) \right),$$

Img.1 : Inception Score(IS) formula

$$FID = \|\mu_r - \mu_g\|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2})$$

Img.2: Frechet Inception Distance (FID) formula

For the Inception Score (IS), the resultant score for the generated image came as 1.0 which states that all the images that are in the batch are classified as the same class and with the standard deviation results as 0.0 we made sure that there was no diversity in the classification of the image batch. Different from the regular IS calculation approach, after the research that we have done, we applied the IS calculation formula by cutting the batch in parts and calculating the

average scores and standard deviations which also gives the same expected results as before. You can observe some of the score results that we were obtained from DCGAN and AAE below:

```
2/2 [==============================] - 3s 398ms/step
Average Inception Score 1.0
Standard Deviation of the Inception Score 0.0
```

Figure.8 DCGAN Inception Score Result

```
2/2 [==============================] - 3s 398ms/step
Average Inception Score 1.0
Standard Deviation of the Inception Score 0.0
```

Figure.9 AAE Inception Score Result

For the Frechet Inception Distance (FID), the resultant scores are observed iteratively in parallel with the training of the models. Since the training process of the generator is in an iterative manner, scores are noted from the last 3 batches of last epochs of the training phase in order to investigate the progress of the image generation. When we calculated the scores from the formula by making comparisons between the original and generated images we observed that our score started to decrease while models became more trained. From this observation we can state that by training the models more we can obtain more results that are close to 0.0 which states that the considered set of images are almost the same.You can observe some of the score results that we were obtained from DCGAN and AAE below:

```
2/2 [==============================] - 3s 348ms/step
2/2 [==============================] - 1s 376ms/step
FID (same): -0.001
2/2 [==============================] - 1s 433ms/step
2/2 [==============================] - 1s 346ms/step
FID (different): 2639.041
```

Figure.10 DCGAN 100 Epoch Result FID Score

```
2/2 [==============================] - 2s 345ms/step
2/2 [==============================] - 1s 345ms/step
FID (same): -0.001
2/2 [==============================] - 1s 345ms/step
2/2 [==============================] - 1s 345ms/step
FID (different): 2240.828
```

Figure.11 DCGAN 400 Epoch Result FID Score

```
4/4 [==============================] - 3s 376ms/step
4/4 [==============================] - 2s 411ms/step
FID (same): -0.001
4/4 [==============================] - 2s 402ms/step
4/4 [==============================] - 2s 397ms/step
FID (different): 2988.197
```

Figure.12 AAE Ten Batch FID Score

```
4/4 [==============================] - 3s 427ms/step
4/4 [==============================] - 2s 516ms/step
FID (same): -0.001
4/4 [==============================] - 2s 410ms/step
4/4 [==============================] - 2s 471ms/step
FID (different): 2692.370
```

Figure.13 AAE One Epoch FID Score

5. CONCLUSION

In this report, we examined the implementation techniques and training of Deep Convolutional Generative Adversarial Networks (DCGANs) and Adversarial AutoEncoders (AAEs) modals on Anime Faces data.

Moreover, the evaluation of the generated synthetic dataset with DCGAN and AAE is performed by the calculation of Inception Score (IS) and Frechet Inception Distance (FID) score compared with the original dataset. The results that are obtained show how newly generated data is aligned with the original data during the iterative training process. Classification of the synthetic images obtained from the pretrained Inception Modal v3 states that we managed to generate good quality images that are classified within the same class.

Further work may be applied in order to generate more high quality images due to computational and time complexity during the training steps of both models. Current versions of the modals are optimized in accordance with the optimal hyperparameters and optimal activation functions. However due to computational power shortage (in terms of CPU, GPU and RAM usage) of the daily computers, it is observed that the training phase of the modals took more time than expected. As a result, the solution can be stated as usage of computer systems that have more computational power. Another future work is to extend the scarcity of the generative modals with enlarged and diverse dataset for the generation of various different fake data. In conclusion, due to many complexity issues that are faced during the project, research for the more optimized and better choices of generative models will be maintained.