**Anime Face Generation using Deep Convolutional Generative Adversarial Network**

Nishita Bajaj

[bajajnishita987@gmail.com](mailto:bajajnishita987@gmail.com)

DY Patil International University

Akurdi, Pune, Maharashtra

School of Computer Science, Engineering and Applications

Abstract — GANs have become a powerful tool that can be used for image synthesis, and is capable of producing high quality and visually consistent images. This paper focuses on applying DCGANs to generate anime faces. Two models were developed: The base model trained for 100 epochs and two improves versions of the base model with 200 and 300 epochs that produce high quality images with better facial details and structure. Research shows extended training effects on quality when generating anime faces - epoch grows more accurate and diverse production. Major challenges - training instability, mode collapses - are addressed, and potential improvement for further research. Conclusion will contribute to applications of AI-generated art, especially in anime character designing, digital illustration and video game development.

Keywords— DCGANs, Generative Adversarial Network, Anime Face Generation, Image Synthesis, Deep Learning

**Introduction**

The Generative Adversarial Network (GANS) has greatly superior the field of synthetic intelligence by using permitting the generation of practical snap shots via damaging education. Between various GAN structure, deep convene generative adversary network (DCGANS) has proved highly effective in synthesizing high satisfactory photos, specifically for based data along with human facials and anime characters.

Anime-style face generation is a complex feature due to range in person designs, which include hairstyles, facial expressions and versions in artistic styles. Traditional image technology strategies battle to capture that complex information, while GANs, especially DCGANs, take gain of deep determination layers to analyse and repeat those characteristics correctly. This study aims to generate anime faces by the use of DCGANs and examine the effect of training duration on the quality of the images.

Three models were evolved:

1. Base model (one hundred generation): Anime establishes a primary information of facial era.
2. Better models (two hundred and three hundred generation): skilled for lengthy periods to produce more state-of-the-art and excessive high-quality outputs.

The key contributions of this work include:

1. Application of DCGANs for anime face generation using convolutional layers to enhance feature extraction.
2. Comparison of training durations to assess improvements in image quality.
3. Discussion of challenges such as mode collapse, training instability, and possible future improvements.

By evaluating the results from each model, these studies indicate how expanded training generates the realism and variety of anime faces. Unlike traditional evaluation techniques, which rely upon matrix inclusive of French inception distance (FID), this study particularly assesses image first-class via visual inspection and qualitative analysis. The findings contribute to the growing field of AI-driven content creation, with ability programs in animation, gaming and digital artwork industries.

**Deep Convolutional Generative Adversarial Network**

1. **Introduction to DCGAN:**

A deep convolutional generative adversarial network (DCGAN) is a better version of traditional GAN architecture that makes use of convolutional layers instead of fully connected layers. This design helps to generate high quality images by learning deep hierarchical features. DCGAN consists of two neural networks, generators and discriminator, who compete against each other in an adverse process.

1. **DCGAN Architecture**

The DCGAN model follows a structured design with specific improvement on traditional GANs:

* Uses convolutional layers instead of fully connected layers for better spatial feature extraction.
* Replaces pooling layers with strided convolutions (in Discriminator) and transposed convolutions (in Generator) for better image quality.
* Makes use of Batch normalization to stabilize training and prevent mode collapse.
* Uses LeakyReLU activation in the Discriminator and ReLU activation in the Generator.

1. **Components of DCGAN:**

* Generator: Generates the fake images
* Discriminator: Differentiates between real and fake images

1. **Training process:**

* Step 1: Generator Training

1. Generator creates fake image from random noise.
2. Discriminator evaluates whether its real or fake.
3. If it is correctly identified by the discriminator, then generator improves itself and creates realistic images.

* Step 2: Discriminator Training:

1. Discriminator makes use of both real images from the dataset and fake images from the generator for its training and learns to differentiates between both of them.

* Step 3: Adversarial Training
  1. Generator tries to fool discriminator by generating fake images that look like real images.
  2. Model is trained until generator becomes capable of producing high-quality images.

**Applications of DCGAN**

DCGAN can be widely used in various areas due to the ability to generate extraordinary images. Some major applications include:

**1.** **Image synthesis and style transfer:**

* Anime and Cartoon Generation: DCGAN is used to generate anime-style faces, characters and creative designs.
* Photo-to-painting conversion: Real-world images can be converted in the form of artwork of unique inventive patterns.
* Style Transfer: Used in innovative industries to apply artistic effects for snap shots.

**2. Data growth:**

* Medical Imaging: DCGANS can produce synthetic medical images (e.g. MRI, CT scan) to strengthen the dataset to train the deep learning model.
* Face Generation: DCGANs can also be used to create training data for face generation model by the creation of synthetic human faces.

**3.** **Gaming and virtual environment:**

* Character production: Practical entertainment for the virtual world produces characters, backgrounds and textures.
* Processive material generation: The guide layout helps in creating newly new-performed items, dynamically.

**4. Fashion and design:**

* Fabrics and Fashion Designs: DCGANS can be used to generate new clothes patterns, styles and textures.
* Interior and Product Designs: AI-based designs assist architects and product designers in innovative ideas.

**5.** **Super-resolution and image growth:**

* Image Upscaling: DCGANS can be used to enhance low-quality images by improving its details and sharpness.
* Restoration of chronic photos: helps restore and paint black and white or damaged photos.

**Methodology (Implementation)**

1. **Dataset**

The model has utilized a custom-curated anime face dataset consists of images that are taken from Kaggle dataset, augmented with extra images sourced from Instagram, Google, and hand-drawn sketches to add variety and beautify generalization for the model. This dataset includes a huge sort of anime faces, differing by means of art style, facial expression, and lights situations.  
  
Preprocessing Steps:

* Resizing: Every image has been resized to 256 × 256 pixels to be able to hold enter dimensions the equal.
* Normalization: Pixel values were normalized to the variety [-1, 1] to enhance stability throughout training.
* Augmentation: Data augmentation strategies, which includes random cropping, flipping, and rotation, have been applied to growth dataset variability and prevent overfitting.
* Filtering: Low-great or beside the point snap shots were manually removed to make sure the dataset contained handiest clear and high-decision anime faces.

1. **Model architecture:**

The Deep Convolutional Generative Adversarial Network (DCGAN) consists of two core components: the Generator and the Discriminator. These models are trained adversarially to improve the quality of generated anime faces.

* 1. **Generator Architecture**

The Generator takes a random noise vector and transforms it into anime face images. The structure consists of:

* Input Layer: A fully connected layer with 8 × 8 × 512 neurons, reshaped into a feature map.
* Reshape Layer: Reshapes the dense output into a spatial function illustration for convolutional processing.
* Transposed Convolution Layers: Up-samples the picture decision step by step:
  + (8×8×512) → (16×16×256)
  + (16×16×256) → (32×32×128)
  + (32×32×128) → (64×64×64)
* Activation Functions:
* ReLU activation within the first layer for characteristic growth.
* LeakyReLU (α=0.2) in intermediate layers to stabilize gradients.
* Tanh activation in the final layer to provide pixel values within the range [-1, 1] for smoother photo era.
* Final Output: A sixty 64×64×3 RGB anime face photo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer Type** | **Output Shape** | **Activation** | **Parameters** |
| Dense | (8,8,512) | ReLU | 12.5 M |
| Reshape | (8,8,512) | ---- | 0 |
| Conv2DTranspose | (16,16,256) | LeakyReLU | 2.1 M |
| Conv2DTranspose | (32,32,128) | LeakyReLU | 820K |
| Conv2DTranspose | (64,64,64) | LeakyReLU | 205K |
| Conv2D (Final layer) | (64,64,3) | Tanh | 3K |

* 1. **Discriminator Architecture**

The discriminator is a binary classifier which is designed to differentiate real anime faces from fake images generated by generator. It takes 64 × 64 × 3 image as input and returns the output as a single probability value (real or fake). The architecture includes:

* Input layer: Accepts 64 × 64 × 3 RGB image.
* Convolutional layers: Remove spatial features using rising filter sizes:
* (64 × 64 × 3) → (32 × 32 × 64)
* (32 × 32 × 64) → (16 × 16 × 128)
* (16 × 16 × 128) → (8 × 8 × 256)
* Pooling layers: Reduces feature maps using max-pooling (2 × 2) to reduce spatial dimensions.
* Flatten layer: The feature converts maps into 1D vector.
* Fully connected layers:
* Dense (512 neurons) with LeakyReLU for feature extraction.
* Dropout to prevent overfitting (0.4).
* Dense (1 neuron) with sigmoid activation for binary classification.

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer Type** | **Output Shape** | **Activation** | **Parameters** |
| Conv2D | (64, 64, 64) | LeakyReLU | 1.8K |
| MaxPooling2D | (32, 32, 64) | — | 0 |
| Conv2D | (32, 32, 128) | LeakyReLU | 73K |
| MaxPooling2D | (16, 16, 128) | — | 0 |
| Conv2D | (16, 16, 256) | LeakyReLU | 295K |
| MaxPooling2D | (8, 8, 256) | — | 0 |
| Flatten | -16384 | — | 0 |
| Dense | -512 | LeakyReLU | 8.4M |
| Dropout (0.4) | -512 | — | 0 |
| Dense (Final) | -1 | Sigmoid | 513 |

* 1. **DCGAN Training Framework:**

The DCGAN consists of a generator and discriminator trained in an adversarial manner. The training procedure follows:

* Adversarial Training:
  + The generator takes random noise as input and generates fake images.
  + The discriminator classifies real and generated images.
  + The generator is updated based on the discriminator’s feedback.
  + Loss Functions: Binary cross-entropy is used for both networks.
  + Optimizers: Adam optimizers with different learning rates (Generator: 0.0003, Discriminator: 0.0001)
  + Beta value set to 0.7 for stability.
* Label Smoothing: Real labels are perturbed slightly to improve generalization and prevent overconfidence in the discriminator.

**Result and Discussion**

**Observations Across Epochs (100, 200, 300)**

* **100 Epochs:**
* The generated images have recognizable facial structures but lack fine details.
* Some artifacts or noise may still be visible.
* Mode collapse might be partially present (similar faces appearing frequently).
* **200 Epochs:**
* Significant improvement in facial details and diversity.
* Better contrast and sharper edges compared to 100 epochs.
* Mode collapse is reduced, leading to more varied facial expressions.
* **300 Epochs:**
* Images appear more refined and high quality.
* Some overfitting may occur, leading to slightly repetitive structures.
* Fine details (like eyes and hair textures) are sharper than in previous models.

**Discussion**

1. **Impact of Training Duration:**

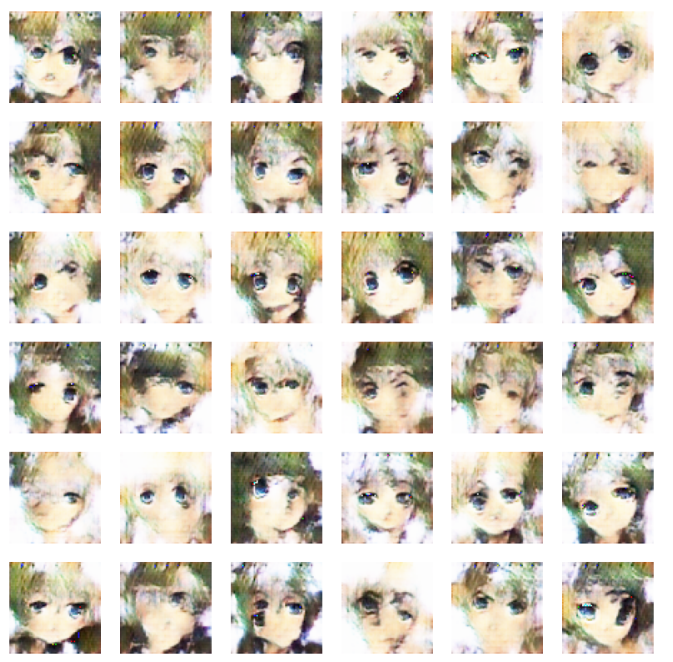
* Increasing epochs improves image quality but at the risk of overfitting.
* 200 epochs seem to offer the best balance between quality and variety.

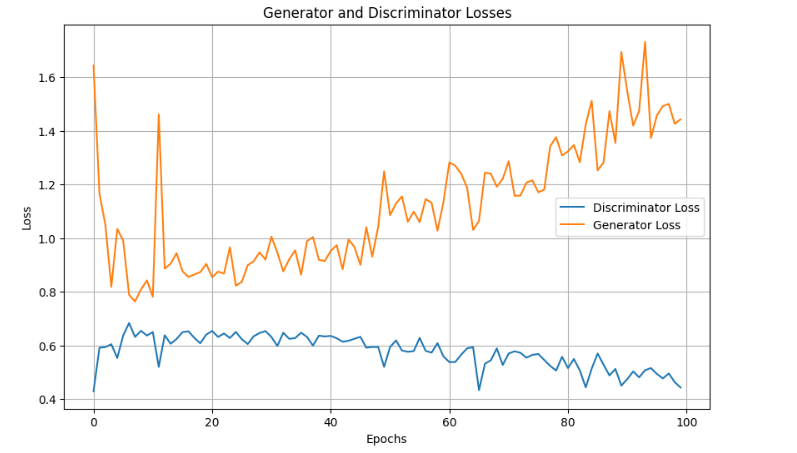
1. **Hyperparameter Effect:**

* **Learning Rate:** Lower lr\_d (0.0001) helped avoid a too-powerful discriminator.
* **Beta (0.7):** Improved stability, reducing mode collapse.
* **Latent Dimension (300):** Allowed richer feature representations.

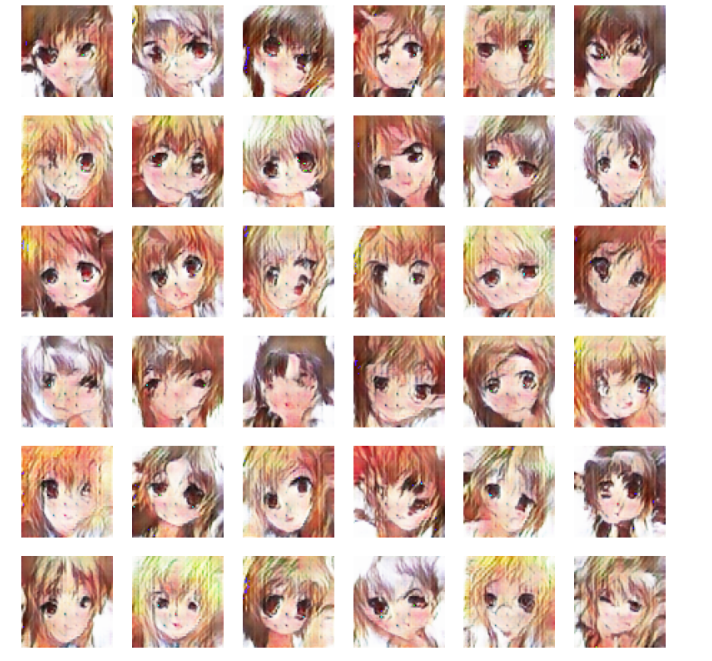
1. **Potential Improvements:**

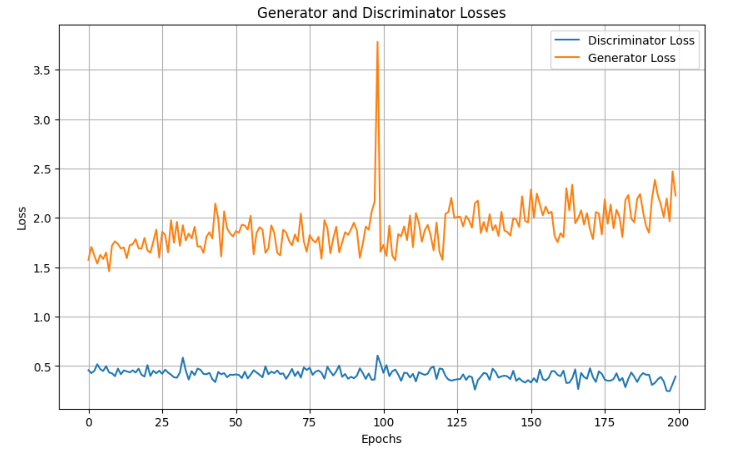
* Adding Fréchet Inception Distance (FID) for better evaluation.
* Using adaptive learning rates to prevent overfitting in later epochs.
* Experimenting with different architectures like Progressive Growing GANs for higher resolution.



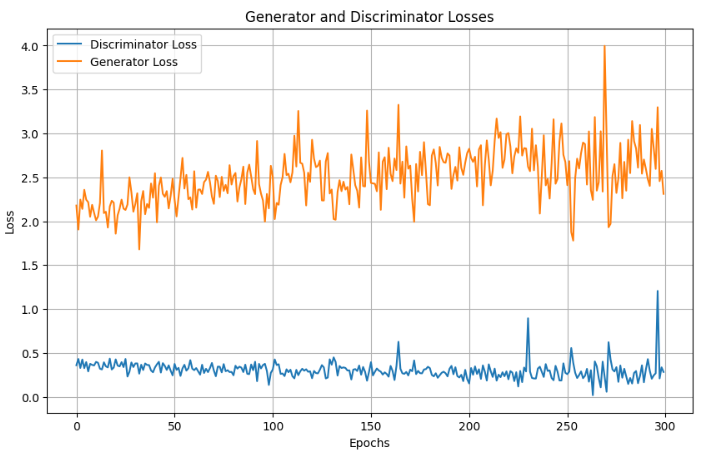


**Figure 1: 100 epochs output and loss graph**

****

****

**Figure 2: 200 epochs output and loss graph**

**** ****

**Figure 3: 300 epochs output and loss graph**

**Conclusion**

This study detected the generation of anime facial using the DCGAN model which are trained for 100, 200 and 300 epochs. The results suggest that the increase in the number of ages improves the quality of the image, but excessive training can lead to overfitting. Trained for 200 epochs, maintaining stable generator-discriminatory training, achieved the best balance between image sharpness, diversity and realism, reducing the collapse of mode while maintaining the collapse of the mode.  
  
key-findings:

* The 100-epoch model produces recognized but less wide images with some artifacts.
* The 200-epoch model produces high quality, varied faces with better realism.
* The 300-epoch model refines further image quality, but shows signs of overfitting and repeating patterns.

Future work can be detected by adaptive learning rates, progressively growing architecture and evaluation matrix to further enhance the performance of the model.

**References**

**[1]** A. Al-qerem, Y. S. Alsalman, and K. Mansour, “Image Generation Using Different Models of Generative Adversarial Network,” in *2019 International Arab Conference on Information Technology (ACIT)*, Al Ain, UAE, 2019, pp. 241–245.

**[2]** M. Li, R. Xi, B. Chen, M. Hou, D. Liu, and L. Guo, “Generate Desired Images from Trained Generative Adversarial Networks,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, 2019, pp. 1–8.

**[3]** Y.-J. Cao, L.-L. Jia, Y.-X. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X.-X. Li, and H.-H. Dai, “Recent Advances of Generative Adversarial Networks in Computer Vision,” *IEEE Access*, vol. 7, pp. 14985–15006, 2018, doi: 10.1109/ACCESS.2018.2886814.

**[4]** S. Wang, D. Bau, and J. Zhu, “Sketch Your Own GAN,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 14030–14040.

**[5]** W. Zhang, “Sketch-to-Color Image with GANs,” in *Proceedings of the 2nd International Conference on Information Technology and Computer Application (ITCA)*, Guangzhou, China, 2020, pp. 322–325.

[6] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” in *4th International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, 2016.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Nets,” in *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, Montreal, Canada, 2014.

[8] Y. Jin, J. Zhang, M. Li, Y. Tian, H. Zhu, and Z. Fang, “Towards the Automatic Anime Characters Creation with Generative Adversarial Networks,” *arXiv preprint arXiv:1708.05509*, 2017. [Online]. Available: <https://arxiv.org/abs/1708.05509>.

[9] B. Li, Y. Zhu, Y. Wang, C.-W. Lin, B. Ghanem, and L. Shen, “AniGAN: Style-Guided Generative Adversarial Networks for Unsupervised Anime Face Generation,” *arXiv preprint arXiv:2102.12593*, 2021. [Online]. Available: <https://arxiv.org/abs/2102.12593>.

[10] M. J. Chong and D. Forsyth, “GANs N' Roses: Stable, Controllable, Diverse Image-to-Image Translation (Works for Videos Too!),” *arXiv preprint arXiv:2106.06561*, 2021. [Online]. Available: <https://arxiv.org/abs/2106.06561>.

[11] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 4681–4690.

[12] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 4401–4410.

[13] H. Kim, M. Park, and S. Yoon, “Anime Face Synthesis Using Deep Convolutional GANs,” *arXiv preprint arXiv:2103.00115*, 2021. [Online]. Available: <https://arxiv.org/abs/2103.00115>.

[14] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, “Least Squares Generative Adversarial Networks,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 2794–2802.

[15] X. Huang and S. Belongie, “Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 1501–1510.

[16] K. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, Australia, 2017, pp. 214–223.

[17] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training Generative Adversarial Networks with Limited Data,” in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, 2020, pp. 12104–12114.

[18] A. Brock, J. Donahue, and K. Simonyan, “Large Scale GAN Training for High Fidelity Natural Image Synthesis,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, 2019.

[19] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, “Deep Generative Adversarial Networks for No-Reference Super-Resolution,” in *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4516–4529, Sep. 2019, doi: 10.1109/TIP.2019.2905125.

[20] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 586–595.