Face Generation Model Based on DCGAN

Yu Dong, Zhou Li

School of Computer Science and Software Engineering, Jincheng College of Sichuan University,
Chengdu 611731, Sichuan, China

Abstract: ***After breakthroughs in computer hardware technology, deep learning has undoubtedly become the biggest winner in the field of learning. Various deep neural networks have achieved remarkable progress in computer vision, speech processing, and natural language processing. By combining CNN with the traditional GAN, DCGAN has made significant advances in unsupervised learning. In this paper, we train the original DCGAN using Python on the TensorFlow deep-learning framework and apply commonly used network-optimization techniques in deep learning, ultimately generating face images that share the same characteristics as the training samples.***

Keywords: Convolutional Neural Network; Generative Adversarial Network; Deep Convolutional Generative Adversarial Network; Face Generation.

1. INTRODUCTION

With the rise of deep learning, GAN has been widely and increasingly maturely applied in image processing and text generation. In recent years, DC-GAN has demonstrated remarkable effectiveness in super-resolution, image inpainting, and image generation, showing that DCGAN is an important deep neural network for tackling unsupervised problems in computer vision. This paper trains and optimizes the network model on a set of face data, ultimately obtaining images generated from noise through the model. Ding and Wu (2024) systematically reviewed self-supervised learning for ECG and PPG signals [1]. Complementing this, Restrepo et al. (2024) developed multimodal embedding alignment for healthcare in low-resource settings [2]. Natural language processing innovations include Yang et al.'s (2025) GAN-based text summarization combining transductive and reinforcement learning [3]. Industrial applications feature Xie and Chen's (2025) multi-agent system Maestro for manufacturing optimization [4]. System reliability research includes Zhu's (2025) RAID framework for anomaly detection in ad systems [5], Zhang's (2025) CrossPlatformStack for high-availability service deployment [6], and Hu's (2025) AdPercept for visual attention modeling in 3D ads [7]. Computer vision advances include Peng et al.'s (2024) domain-adaptive human pose estimation through representation aggregation-segregation [8]. Anomaly detection is furthered by Zhang et al.'s (2025) ML techniques for biomechanical big data [9], while healthcare AI features Wang's (2025) transformer-GNN hybrid RAGNet for arthritis risk prediction [10]. Enterprise solutions encompass Qi's (2025) generative AI framework AUBIQ for automated business intelligence [11], Fang's (2025) microservice-driven low-code platform for SME digital transformation [12], and Lin's (2025) product management approach to AI governance [13]. Foundational data techniques include Chen's (2023) data mining applications [14]. Computer vision research continues with Wang, Li, and Li's (2024) YOLOv8-based road car detection [15], while causal AI is advanced by Wang's (2025) targeted learning method for MNAR recommendation data [16].

2. A BRIEF DISCUSSION OF GAN AND DCGAN

2.1 GAN

The core idea of the Generative Adversarial Network is to learn the feature distribution underlying a large set of training samples and then, based on this distribution model, use specified noise data to generate new samples that conform to the learned distribution, thereby expanding our sample database. GAN consists of two sub-networks: a generative model and a discriminative model. These two models compete against each other, ultimately training a powerful generator [1].

2.2 DCGAN

Deep Convolutional Generative Adversarial Networks, abbreviated as DCGAN, is a new deep network proposed on the basis of the GAN introduced by Ian Goodfellow in 2014, by incorporating a convolutional architecture to address the instability of GAN training [2]. Before this, CNNs had achieved great success in supervised learning tasks such as image classification and object detection, yet made little progress in unsupervised learning. Therefore, the authors of DCGAN combined CNNs with GANs to tackle unsupervised learning problems, and experiments yielded excellent results. This paper will reproduce the entire workflow of the DCGAN network by implementing a face-image-generation model, optimize the original architecture, and demonstrate its advantages in handling unsupervised learning tasks. Network-optimization techniques obtained from deep-learning research will also be applied to the DCGAN network, aiming to improve its learning speed and feature-extraction capability in several aspects.

3. MODEL OBJECTIVES

In real life, we expect computers to accomplish tasks according to our intentions—for instance, if we want an image of a flower, the computer can quickly generate one. This is the fundamental goal of generative networks. This thesis will train a DCGAN model to produce face images that do not exist in reality but are generated based on the features learned from the training data.

4. DATA PREPARATION

4.1 Data Collection

Part of the model in this paper employs convolutional networks, which have strong feature-extraction capabilities for image data. Therefore, a large number of face images were collected from online face databases to train the convolutional network.

4.2 Data Processing

After collection, the data are preprocessed. First, OpenCV and a Haar-feature-based cascade algorithm are used to detect faces in the images; the face regions are cropped and saved, while images without faces are discarded. Next, all images are resized to 64×64×3 (3 denotes the number of channels). Finally, the image data are stored in a directory structure.

4.3 Data Loading

The network in this paper is built on the TensorFlow framework, so training data must be fed into the network in tensor format. To balance training speed and feature-learning ability, OpenCV is used to read all images, which are then input to the network in batches.

5. NETWORK ARCHITECTURE INTERPRETATION

DCGAN consists of two functional modules: a generator module and a discriminator module. The generator takes input noise $Z(x)$ and uses it to produce images $G(x)$. The discriminator's role is to examine input images x and G(X) and decide whether each image is real or fake, i.e., generated by the generator. For example, if a real face image from the training set is fed to the discriminator, it should return the result that the image is real. Conversely, if a fake image produced by the generator is fed to the discriminator, it should return the result that the image is fake. When both the generator and discriminator have been trained to a certain level, the discriminator's output hovers around 0.5, indicating that the generator has become sufficiently powerful: the generated images are highly consistent with real images in terms of feature distribution, so the discriminator can no longer tell whether the image is real.

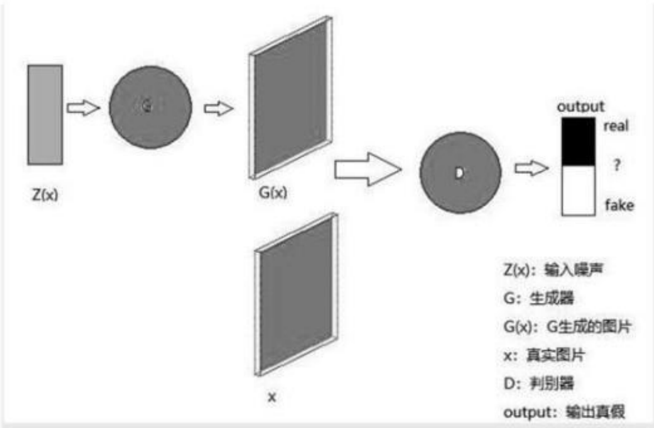

Figure 1: DCGAN network architecture

5.1 Generator

In this paper, the generator's input is random noise $Z(x)$ drawn from a uniform distribution over [-1,1]; the noise dimension is assumed to be [None, 100]. Instead of the convolution used in CNNs, the model employs transposed convolutions to progressively transform the noise into feature maps, ultimately yielding a color image of size 64*64*3. To ensure overall network stability, the architecture uses an output layer and transposed convolutions (also called fractionally-strided convolutions) to convert noise into images; after each convolution the image size doubles and the number of channels halves [3].

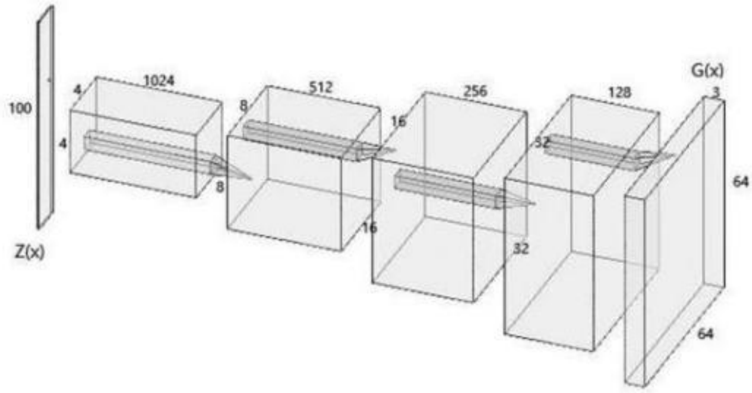

Figure 2: Generator G architecture

5.2 Discriminator

The discriminator classifies input images and outputs the classification result (1 for real data, 0 for images produced by the generator). This part can therefore adopt traditional CNNs such as LeNet, AlexNet, VGGNet, etc. [4]. However, this paper makes two modifications: all pooling layers are replaced by strided convolutions, and, except for the output layer, Leaky ReLU is used as the activation function throughout.

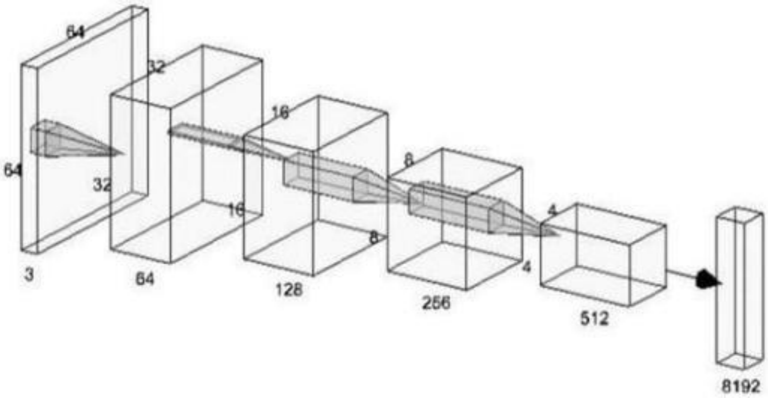

Figure 3: Discriminator D architecture

6. NETWORK MODEL OPTIMIZATION

After extensive literature review and parameter tuning, the following methods are adopted to optimize the network, enhancing model stability and the quality of generated images.

a) All discriminator pooling layers are replaced by strided convolutions.

b) Apply batch normalization to both the generator and the discriminator; this trick can prevent training collapse caused by poor parameter initialization. However, applying batch normalization to every layer can destabilize the model, so the output layer of the generator and the input layer of the discriminator are excluded from batch normalization.

c) Remove all fully-connected layers from the network, because the original CNN contains most of its parameters in these layers. Retaining fully-connected layers can improve model stability but drastically slows convergence, which is highly undesirable.

d) In the generator, use the ReLU function for nonlinear mapping in every layer except the output layer, which uses the Tanh activation function.

e) Use the LeakyReLU activation function in every layer of the discriminator.

7. HYPERPARAMETERS

Table 1: Hyperparameter Table

参数名	参数值
噪声维度	[64,100]
批次大小	64
学习率	0.0003
迭代次数	100

8. AUXILIARY FUNCTIONS

a) Batch training data loader: this paper adopts mini-batch training, splitting the data into batches whose size we choose ourselves; selecting an appropriate batch size maintains both training speed and stability.

b) Generated image display function: to monitor optimization progress, this function is implemented to feed a batch of noise into the generator every fixed number of iterations and inspect the resulting image quality. If the displayed images become progressively clearer, the network is steadily improving—an exciting sign.

c) This paper uses the cross-entropy function as the loss function.

d) The network is trained with gradient descent and optimized using the AdamOptimizer.

9. IMAGE GENERATION RESULTS

Figure 4: Real human face images

Epoch 0/100 Discriminator Loss:1.2113 Generator Loss:1.5655 Generator Loss:4.4816 Generator Loss:5.8984

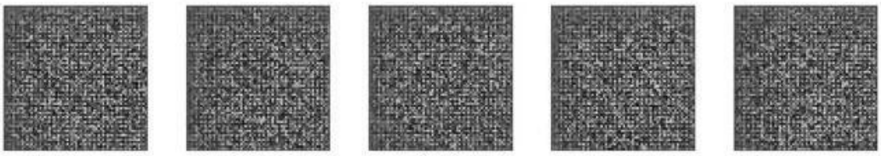

Figure 5: Human faces generated after the 1st iteration

Epoch 20/100 Discriminator Loss:0.4854 Gen-

Figure 6: Faces generated at the 20th iteration

Epoch 60/100 Discriminator Loss:0.3544 Gen-

Figure 7: Faces generated at the 60th iteration

10. CONCLUSION

By combining supervised and unsupervised learning, DCGAN breaks away from the traditional mindset of building deep neural networks, shifting from the previous linear stacking approach to a bidirectional adversarial network structure that integrates convolution and transposed convolution. This enables a generation paradigm that moves from samples to features and then to new samples, offering a refreshing perspective in deep neural network research and greatly stimulating our imagination. Of course, there is no free lunch; DCGAN also has its own shortcomings. For instance, we can observe from the loss values generated during the experiment that the iterative process of the model is unstable—the loss does not decrease monotonically but fluctuates within a large range. This indicates poor interpretability of the model: it does not update its weights according to the gradient-descent optimization we set. We can only tinker with the model's hyperparameters in hopes of obtaining a more stable performance [5].

REFERENCES

[1] Ding, Cheng, and Chenwei Wu. "Self-Supervised Learning for Biomedical Signal Processing: A Systematic Review on ECG and PPG Signals." medRxiv (2024): 2024-09.

[2] Restrepo, David, et al. "Multimodal Deep Learning for Low-Resource Settings: A Vector Embedding Alignment Approach for Healthcare Applications." medRxiv (2024): 2024-06.
[3] Yang, Jing, et al. "A generative adversarial network-based extractive text summarization using transductive and reinforcement learning." IEEE Access (2025).
[4] Xie, Minhui, and Shujian Chen. "Maestro: Multi-Agent Enhanced System for Task Recognition and Optimization in Manufacturing Lines." Authorea Preprints (2025).
[5] Zhu, Bingxin. "RAID: Reliability Automation through Intelligent Detection in Large-Scale Ad Systems." (2025).
[6] Zhang, Yuhan. "CrossPlatformStack: Enabling High Availability and Safe Deployment for Products Across Meta Services." (2025).
[7] Hu, Xiao. "AdPercept: Visual Saliency and Attention Modeling in Ad 3D Design." (2025).
[8] Peng, Qucheng, et al. "Exploiting Aggregation and Segregation of Representations for Domain Adaptive Human Pose Estimation." arXiv preprint arXiv:2412.20538 (2024).
[9] Zhang, Shengyuan, et al. "Research on machine learning-based anomaly detection techniques in biomechanical big data environments." Molecular & Cellular Biomechanics 22.3 (2025): 669-669.
[10] Wang, Y. (2025). RAGNet: Transformer-GNN-Enhanced Cox–Logistic Hybrid Model for Rheumatoid Arthritis Risk Prediction.
[11] Qi, R. (2025). AUBIQ: A Generative AI-Powered Framework for Automating Business Intelligence Requirements in Resource-Constrained Enterprises. Frontiers in Business and Finance, 2(01), 66-86.
[12] Fang, Z. (2025). Microservice-Driven Modular Low-Code Platform for Accelerating SME Digital Transformation.
[13] Lin, Tingting. "ENTERPRISE AI GOVERNANCE FRAMEWORKS: A PRODUCT MANAGEMENT APPROACH TO BALANCING INNOVATION AND RISK."
[14] Chen, Rensi. "The application of data mining in data analysis." International Conference on Mathematics, Modeling, and Computer Science (MMCS2022). Vol. 12625. SPIE, 2023.
[15] Wang, Hao, Zhengyu Li, and Jianwei Li. "Road car image target detection and recognition based on YOLOv8 deep learning algorithm." unpublished. Available from: http://dx. doi. org/10.54254/2755-2721/69/20241489 (2024).
[16] Wang, Hao. "Joint Training of Propensity Model and Prediction Model via Targeted Learning for Recommendation on Data Missing Not at Random." AAAI 2025 Workshop on Artificial Intelligence with Causal Techniques. 2025.