DOI: http://dx.doi.org/10.26483/ijarcs.v15i6.7155

Volume 15, No. 6, November-December 2024

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Research on Digital Virtual Human Head Avatar Generation Technology with Cartoon Style

Bin Xiao
School of Information Engineering, Guangzhou Vocational College of Technology & Business, Guangzhou 511442, China

Yumeng Peng
School of Information Engineering, Guangzhou Vocational College of Technology & Business, Guangzhou 511442, China

Yiheng Feng
School of Information Engineering, Guangzhou Vocational College of Technology & Business, Guangzhou 511442, China

Abstract: This paper proposes a Generative Adversarial Network (GAN)-based approach for generating anime-style cartoon avatars, addressing the current limitations in cartoon avatar datasets, including their scarcity and copyright restrictions. Due to the limited availability of high-quality, diverse cartoon datasets, particularly for anime-style faces, obtaining large-scale labeled data for training deep learning models remains a significant challenge. To overcome these issues, we train a GAN model on a large collection of anime-style images to generate unique, high-quality cartoon faces. The generator learns to capture the distinctive features of anime art, such as exaggerated expressions, vibrant colors, and stylized facial proportions, while the discriminator ensures the realism and diversity of the generated avatars. Our experimental results demonstrate that the proposed method not only produces visually appealing avatars faithful to the anime aesthetic but also offers a scalable solution to the datasets limitations and copyright concerns. This approach opens up new possibilities for applications in digital art, gaming, and social media, where custom cartoon avatars are increasingly in demand.

Keywords: Generative Adversarial Network; Cartoon Avatars; Artificial Intelligence; Image Processing; Deep Learning

I. INTRODAUCTION

A. *Research Background*

With the rapid development of digital technology, digital avatars have gradually permeated various aspects of daily life, particularly in social media, video games, virtual reality (VR), and augmented reality (AR) fields. These virtual characters not only play a significant role in visual representation but also become crucial elements in the digital world in terms of emotional interaction, personalized expression, and user experience. Among various forms of virtual avatars, cartoon-style digital avatars have gradually gained increasing attention and favor due to their unique visual appeal, simple yet expressive design, and effective emotional communication.

Cartoon-style virtual avatars typically feature exaggerated facial features and vibrant color combinations, which not only enhance emotional interaction with users but also avoid discomfort caused by overly realistic facial expressions (i.e., the "uncanny valley" effect). Therefore, how to utilize advanced technology to generate cartoon-style digital avatars is not only a subject with high market demand but also holds significant technical research value. Although significant progress has been made in the field of image generation, most research so far has focused on generating realistic-style virtual avatars, emphasizing details and realism. In contrast, the generation of cartoon-style virtual avatars places more emphasis on stylized details, emotional expression, and visual appeal, facing unique challenges. These challenges include balancing the diversity and controllability of generated avatars while maintaining a consistent and artistic cartoon style.

Therefore, this paper aims to explore and research methods for generating cartoon-style digital avatars based on artificial intelligence and deep learning technologies. Specifically, it focuses on how to combine traditional avatar generation methods with the unique characteristics of cartoon styles through neural networks and style transfer techniques to achieve efficient, natural, and creative avatar generation. The significance of this research lies not only in providing technical support for the generation of cartoon-style avatars but also in offering new ideas and methods for digital art creation, virtual character design, and personalized customization on social platforms. This paper will focus on analyzing the key technologies for generating cartoon-style avatars and exploring their potential applications in practice, aiming to provide a theoretical foundation and technical support for the future development and application of virtual avatar technology.

B. *Related work*

The generation of virtual avatars can be mainly classified into two categories: traditional image processing techniques and deep learning-based techniques. Traditional methods often rely on manual design and rule-driven style transfer algorithms [1]. However, these methods are often limited in terms of detail handling and style consistency, and they offer poor flexibility and controllability during the generation process. In recent years, the development of deep learning, particularly Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs), has led to significant progress in the generation of cartoon-style avatars. For example, Generative Adversarial Networks (GANs) have been widely used to generate high-quality, style-consistent cartoon avatars [2]. Through adversarial training, GANs can maintain a high level

of visual consistency and creativity during generation. However, generating cartoon-style avatars still faces several challenges. First, handling the differences between cartoon and realistic styles, particularly in balancing between maintaining personal features and exaggerated artistic expressions, is a significant challenge [3]. Second, achieving diversity in the generated avatars, to avoid overly monotonous results that lack individuality, is also a key issue [4]. Furthermore, generating cartoon-style avatars requires special attention to facial expressions and emotional conveyance, as these features directly affect the interactive experience of virtual humans.

Deep learning, especially Generative Adversarial Networks (GANs), has shown tremendous potential in the generation of cartoon-style avatars. GANs, through the adversarial training of two neural networks (the generator and discriminator), can generate high-quality images and achieve ideal style transfer results. Reference [5] proposed a GAN-based method for generating cartoon-style faces, which effectively captures multi-level image features using multi-scale convolutional networks, enabling the transformation from real faces to cartoon avatars. This method enhances style consistency during the generation process by employing Conditional Generative Adversarial Networks (cGANs). Another important method in generating cartoon-style avatars is style transfer technology. Style transfer involves applying the style of one image to another, generating an image with a specific visual style. Reference [6] explored how deep convolutional neural networks can be used for style transfer, successfully applying this technique to generate cartoon-style images. This method allows the preservation of image content while imparting a unique cartoon-like visual style.

The research on cartoon-style avatar generation relies heavily on large-scale datasets. Common datasets used for training cartoon avatar generation models include CelebA [7] and CartoonSet [8]. These datasets provide a large number of clearly labeled and diverse avatar images, offering rich data support for training deep learning models. Additionally, the evaluation of generated results typically involves criteria such as image visual quality, style consistency, and the accuracy of emotional expression. The Frechet Inception Distance (FID) is a commonly used metric for assessing the quality of generated images [9], while establishing specific evaluation standards for cartoon-style generation remains an ongoing research topic.

II. THE GAN-BASED AVATAR GENERATION METHOD

The GAN model was first proposed by Professor Ian J. Goodfellow and his team in 2014 in their paper "Generative Adversarial Nets" [2]. As a form of unsupervised learning, GAN consists of two independent neural networks: a generator (G) and a discriminator (D). The generator (G) is responsible for generating fake samples, while the discriminator (D) is used to distinguish whether the samples produced by G are real data or fake data. The result of each judgment is used as input for backpropagation to both G and D. If D makes the correct judgment, the parameters of G need to be adjusted to make the generated fake data more realistic; if D makes an incorrect judgment, the parameters of D need to be adjusted to prevent similar errors in future judgments. Training continues until both networks reach a balanced and harmonious state. The ultimate goal of the GAN model is to obtain a high-quality automatic generator and a strong classifier with good judgment capabilities.

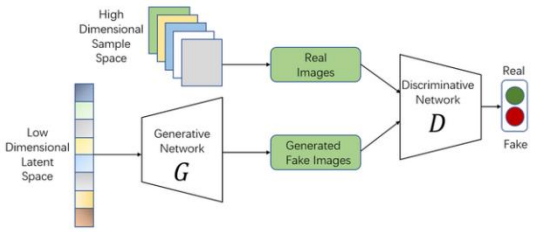


Figure 1. Schematic Diagram of GAN Model Structure

The main task of the generator is to produce "fake" images that resemble real images, starting from random noise. In this GAN model, the generator receives a random noise vector of shape (100, 1, 1) as input. Through a series of transposed convolution (ConvTranspose2d) operations, the generator gradually enlarges the spatial dimensions of the noise vector. Activation functions (such as ReLU) and batch normalization (BatchNorm2d) are applied to ensure the quality and stability of the output image. Ultimately, the generator transforms the noise vector into a 64x64 RGB image, normalizing the pixel values to the range of [-1, 1] using the Tanh activation function.

The task of the discriminator is to distinguish whether an input image comes from the real dataset or is a fake image generated by the generator. The discriminator takes a 64x64 RGB image as input, which passes through multiple convolutional layers (Conv2d) for feature extraction. The LeakyReLU activation function is used to enhance the model's nonlinear capabilities. Each convolutional layer extracts different levels of features from the image, helping the discriminator determine its authenticity. In the end, the discriminator outputs a single scalar value, which is passed through a Sigmoid activation function to convert it into a probability between 0 and 1, indicating the likelihood that the input image is real. The process is illustrated in the figure below:

Layer Type	Input Size	Operation (Convolution/Transpose)	Output Size
ConvTranspose2d	(100, 1, 1)	4x4 kernel, stride 1, padding 0	(512, 4, 4)
BatchNorm2d	(512, 4, 4)	-	(512, 4, 4)
ReLU	(512, 4, 4)	-	(512, 4, 4)
ConvTranspose2d	(512, 4, 4)	4x4 kernel, stride 2, padding 1	(256, 8, 8)
BatchNorm2d	(256, 8, 8)	-	(256, 8, 8)
ReLU	(256, 8, 8)	-	(256, 8, 8)
ConvTranspose2d	(256, 8, 8)	4x4 kernel, stride 2, padding 1	(128, 16, 16)
BatchNorm2d	(128, 16, 16)	-	(128, 16, 16)
ReLU	(128, 16, 16)	-	(128, 16, 16)
ConvTranspose2d	(128, 16, 16)	4x4 kernel, stride 2, padding 1	(64, 32, 32)
BatchNorm2d	(64, 32, 32)	-	(64, 32, 32)
ReLU	(64, 32, 32)	-	(64, 32, 32)
ConvTranspose2d	(64, 32, 32)	4x4 kernel, stride 2, padding 1	(3, 64, 64)
Tanh	(3, 64, 64)	-	(3, 64, 64)

Figure 2. Generator

Layer Type	Input Size	Operation (Convolution)	Output Size
Conv2d	(3, 64, 64)	4x4 kernel, stride 2, padding 1	(64, 32, 32)
LeakyReLU	(64, 32, 32)	-	(64, 32, 32)
Conv2d	(64, 32, 32)	4x4 kernel, stride 2, padding 1	(128, 16, 16)
BatchNorm2d	(128, 16, 16)	-	(128, 16, 16)
LeakyReLU	(128, 16, 16)	-	(128, 16, 16)
Conv2d	(128, 16, 16)	4x4 kernel, stride 2, padding 1	(256, 8, 8)
BatchNorm2d	(256, 8, 8)	-	(256, 8, 8)
LeakyReLU	(256, 8, 8)	-	(256, 8, 8)
Conv2d	(256, 8, 8)	4x4 kernel, stride 2, padding 1	(512, 4, 4)
BatchNorm2d	(512, 4, 4)	-	(512, 4, 4)
LeakyReLU	(512, 4, 4)	-	(512, 4, 4)
Conv2d	(512, 4, 4)	4x4 kernel, stride 1, padding 0	(1, 1, 1)
Sigmoid	(1, 1, 1)	-	(1)

Figure 3. Discriminator

III. EXPERIMENTAL ANALYSIS

A. *Experiments and Data Description*

The simulation analysis conducted in this study utilizes the Windows 10 operating system as the experimental platform. The computer employed for this purpose is outfitted with an Intel(R) Core i7-1185G7 processor, an NVIDIA GPU p100 graphics card and 16 GB of RAM. The applied simulation tool is PyCharm Community 2023, which uses the Python language to complete simulation experiments and analyses based on the PyTorch deep learning framework. Cartoon Set is a collection of random, 2D cartoon avatar images[10]. The cartoons vary in 10 artwork categories, 4 color categories, and 4 proportion categories, with a total of 1013 possible combinations. We provide sets of 10k and 100k randomly chosen cartoons and labeled attributes.

B. *Model Training*

In this experiment, a larger-scale data sample is used. Based on the grid search idea, the hyperparameters of the model are optimized. Each training batch consists of 64 samples, and the model is trained for 300 epochs. Both the generator and discriminator are assigned a fixed learning rate of 0.0002, employing an ensemble-based optimizer. The dimensionality of the input noise is set to 100. The performance of the model's training is assessed using the training errors from both the generator and discriminator, with binary cross-entropy chosen as the loss function. The loss function is defined as follows:

$$\mathrm{BCELoss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \cdot \log p(y_i = 1) + (1 - y_i) \cdot \log(1 - p(y_i = 1))] \quad (1)$$

In this context, y_i signifies the binary target label (either 0 or 1) for the i-th instance, while $p(y_i = 1)$ denotes the model's predicted probability for the i-th instance, reflecting the likelihood that the model classifies the i-th instance with a label of 1.

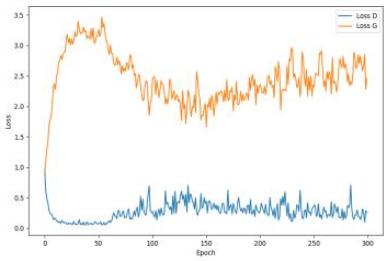


Figure 4. Training errors of the generator and discriminator

The model's error curves converged and stabilized after about 50 epochs, indicating rapid achievement of balance in adversarial training. The generator's error consistently hovered around 5, while the discriminator's error remained stable at about 1, implying a well-balanced adversarial relationship. We observed two significant spikes in the generator's error, reaching values of about 10 and 16. These likely represent critical moments where the generator faced challenges in fooling the discriminator, possibly due to sudden improvements in the discriminator's performance. Despite these fluctuations, the overall training process remained stable, highlighting the robustness of the GAN architecture.

C. *Performance Metrics*

We have chosen a combination of SSIM, PSNR as our evaluation metrics. SSIM and PSNR are traditional image quality metrics that provide insights into the structural similarity and signal fidelity of generated images compared to real ones.

SSIM (Structural Similarity Index Measure) functions as a metric to assess the resemblance between two images. It hinges on three fundamental comparison criteria between samples x and y, namely luminance, contrast, and structural integrity.

$$\mathrm{SSIM}(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \quad (2)$$

and $\alpha > 0, \beta > 0, \gamma > 0$ represent the relative importance of each measure. The functions include luminance comparison, contrast comparison, and structure comparison.

PSNR (Peak Signal to Noise Ratio) serves as a measure for assessing the quality of an image post-compression in comparison to its original form. An elevated PSNR value signifies reduced distortion following compression.

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{(2^n-1)^2}{\mathrm{MSE}}\right) \quad (3)$$

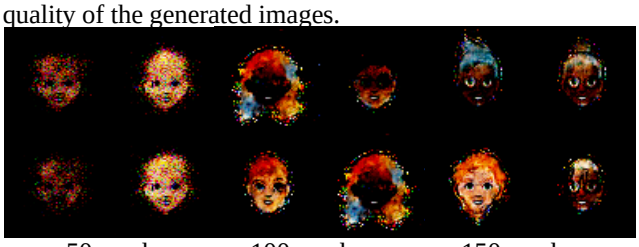
In this context, n denotes the number of bits per pixel, expressed in decibels (dB). A larger numerical value implies lower distortion. MSE stands for Mean Square Error, representing the average of the squares of errors between the current image X and the reference image Y.

$$\mathrm{MSE} = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}(X(i,j) - Y(i,j))^2 \quad (4)$$

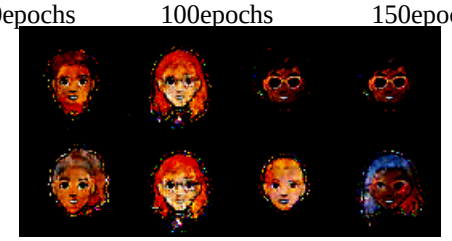
where H denotes the height of the image, and W signifies its width, respectively.

D. *Model Validation*

The table displays the evaluation of the generated image quality using two metrics: SSIM and PSNR at different training epochs. From the data, it can be observed that the quality of the generated images fluctuates during the early stages of training. At 150 epochs, the SSIM and PSNR reached their highest values, 0.6344 and 14.2750, respectively, indicating that the generated images were of the best quality at this point. However, as the number of training epochs increased, particularly at 200 and 250 epochs, both SSIM and PSNR values showed a slight decline, suggesting a degradation in image quality. This could be due to instability or overfitting during the training process, preventing further improvement in the quality of the generated images.



50epochs 100epochs 150epochs



200epochs 250epochs

Figure 5. Generated Cartoon Avatars

From the data above, it can be observed that the quality of the generated images fluctuates throughout the early and later stages of training. Notably, at 150 epochs, both SSIM and PSNR reached their peak values, indicating that at this point, the generator was able to produce cartoon-style images that

were closest to real images. However, as training continued, despite some fluctuations, the SSIM and PSNR values slightly declined at 200 and 250 epochs. This could be due to overfitting in the later stages of training, or the adversarial training between the generator and discriminator causing an unstable learning process. The decline in the later stages of training may also reflect a reduction in the diversity of images generated by the generator, with no significant improvements in quality.

Table I. Performance Statistics of Generated Cartoon Avatars

Metric	***50 epochs***	***100 epochs***	***150 epochs***	***200 epochs***	***250 epochs***
SSIM	0.5756	0.5605	0.6344	0.6200	0.5541
PSNR	12.8518	12.5694	14.2750	13.4414	13.0245

IV. CONCLUSION

This paper presents a GAN-based method for generating cartoon-style digital avatars, aiming to address the issues of data scarcity and copyright limitations in current cartoon avatar datasets. Experimental results show that the proposed method can generate visually appealing avatars that align with anime aesthetics. It also provides an extensible solution to datasets limitations and copyright concerns. This method opens up new possibilities for custom cartoon avatar applications in fields such as digital art, gaming, and social media, meeting the growing demand for personalization.

Funding

This research is supported by the Guangdong Province Higher Education Institution Special Innovation Project(2022KTSCX301), the Key Project of Guangzhou Key Laboratory of Intelligent Interactive Technology and the Application Laboratory Open Research Topics(2022SYS04) and the High-Level Talent Start-up Project of Guangzhou Vocational College of Technology & Business.

References

[1] Hertzmann, A., et al. (2001). Image analogies. ACM SIGGRAPH 2001 Papers, 327-340.

[2] Goodfellow, I., et al. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 27.

[3] Zhu, J., et al. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. IEEE International Conference on Computer Vision, 2223-2232.

[4] Liu L, Ouyang W, Wang X, et al. Deep learning for generic object detection: A survey[J]. International journal of computer vision, 2020, 128: 261-318.

[5] Sankalpa D, Ramesh J, Zualkernan I. Using generative adversarial networks for conditional creation of Anime posters[C]//2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT). IEEE, 2022: 197-203.

[6] Gupta V, Sadana R, Moudgil S. Image style transfer using convolutional neural networks based on transfer learning[J]. International journal of computational systems engineering, 2019, 5(1): 53-60.

[7] CelebA Liu Z, Luo P, Wang X, et al. Large-scale celebfaces attributes (celeba) dataset[J]. Retrieved August, 2018, 15(2018): 11.

[8] Zheng Y, Zhao Y, Ren M, et al. Cartoon face recognition: A benchmark dataset[C]//Proceedings of the 28th ACM international conference on multimedia. 2020: 2264-2272.

[9] Heusel M, Ramsauer H, Unterthiner T, et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium[J]. Advances in neural information processing systems, 2017, 30.

[10] https://www.kaggle.com/datasets/brendanartley/cartoon-faces-googles-cartoon-set

