Researches Advanced in Generative Adversarial Networks and Their Applications for Image-Generating NFT

Xiaolin Guo

Accademia di Belle Arti di Roma RM 00175, Italy

xiaolin.guo@studenti.abaroma.it

Abstract. A generative adversarial network is a deep learning model, an unsupervised learning method. In computer vision, the generative adversarial network is a research direction with rapid development in recent years; Similarly, the rise of cryptocurrency Non-Fungible Tokens (NFT) in recent years has also attracted much attention to the field of art. As an "irreplaceable currency" NFT provides a more novel and convenient way for content creators and artists to create and increases the continuous income of original creators. At the same time, it has also attracted widespread attention to the financial field. Therefore, this paper is determined to combine the generative adversarial network of the production of NFT and discuss and analyze the autonomous computer generation of artworks. Firstly, this paper starts with the model's structure, the design of the objective function, Block chain technology, and Irreplaceable tokens encrypted using blockchain technology. Then, the image generated by the whole generative adversarial network and transformed into NFT works are described in detail. In addition, this paper briefly discusses the development ethics of human art and machine art and the prospects for its development trend.

Keywords: Image-Generating NFT; Human Art and Machine Art; Autonomous Computer Generation of Artworks.

1. Introduction

Image generation has been widely concerned in deep computer learning, artificial intelligence, and computer vision, which aims to generate high-precision clear target images in massive image datasets. Thanks to convolutional neural networks' powerful feature representation capabilities, image generation algorithms based on deep learning, especially generative models, have gradually become the mainstream framework. Different from the discriminative model, which infers some properties of the image from the original image (such as inferring the name of the number from the digital image), the primary task of the generative model is to generate the corresponding image based on the specific information input. As the most representative generative model, generative adversarial networks (GANs) have been widely used in various image generation tasks in recent years and have continuously made breakthroughs in generation quality and speed.

The quality of image generation is closely related to the complexity of modeling data. Everyday image generation tasks include face image generation, font image generation, animation image generation, etc. The rise of cryptocurrency Non-Fungible Tokens (NFTs) in recent years has also attracted widespread attention in the art world. As an "irreplaceable currency" NFT provides content creators and artists with a more novel and convenient way of creation, increasing the endless benefits for original creators. At the same time, it has also attracted extensive attention in the financial field. To this end, we cannot help but think about whether we can use the automatic image generation algorithm to generate Non-Fungible Tokens images and give them specific artistic properties.

Focusing on the above issue, in this paper, we aim to introduce in detail the transformation of computer-generated images into artworks, which focuses on the development and various usages of GAN, analyzes the structure and training mode of GAN, and analyzes the different processing methods of GAN with various developments. For the development of GAN, as well as the primary operating mode of BigGAN, the basic principle of StyleGAN, the description of StackGAN theory, the application of CycleGAN, the development of new technology Pix2pix, and a brief overview of acGAN. Examples: autonomous image generation, multiple edits of images, high granularity and high

definition, gamut change of image state, text to image, change face age, edit and generate video, etc. Applications are also growing, such as text-to-image generation, font generation, dialogue generation, machine translation, etc. Moreover, it describes in detail how to edit the digital ID of the computer-generated image, create the exclusive smart contract of NFT, and put it in the commercial market for sale. Finally, the ethical issues of machine-generated art will be discussed.

2. Pipeline of Traditional GAN

Generative Adversarial Networks (GANs) were first proposed in 2014 by Ian J. Goodfellow and colleagues in June 2014 [1]. Generative Adversarial networks are robust neural networks built on the capabilities of unsupervised deep learning. It is a class of learning machine frameworks. It has a wide range of applications in many computer vision fields today (in the form of a zero-sum game) to analyze, capture, and replicate variants in a dataset [2]. In addition, the potential use of GAN makes it a key focus in the future field of deep computer learning. It can generate high-quality images, simulate face changes, improve medical image segmentation, etc. Which has significantly advanced art, medical treatment, and computer vision. A GAN comprises two essential components: a Generator (abbreviated G) and a Discriminator (abbreviated D). Generator: a machine that generates data to "fool" the discriminator as much as possible. The generated data is denoted as G(z). Discriminator: Determines whether the data is accurate or generated by the generator to find fake data generated by the generator as much as possible. The input parameter is x, where x is the data, and the output D (x) is the probability that x is accurate data. If it is 1, it is 100% accurate data; if it is 0, it cannot be actual data. This way, G and D constitute a dynamic confrontation (or game process). As the training (confrontation) progresses, G's data gets closer to the actual data, and the level of D discriminating data gets higher and higher. In an ideal world, G can generate enough data to be "fake"; for D, it is difficult to determine whether the data generated by the generator is accurate or not, so D(G(z)) = 0.5. After training, we get a generative model G that can be used to generate fake data.

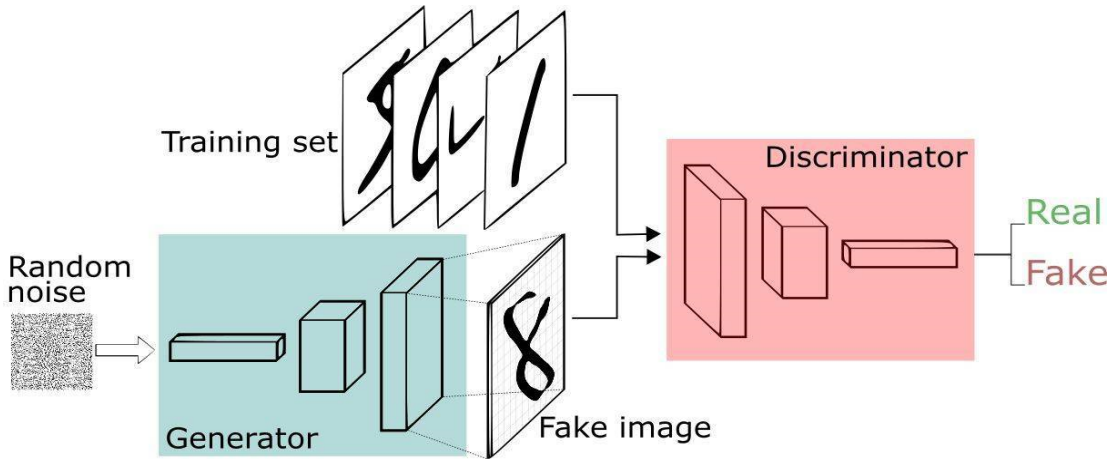

Figure 1. The pipeline of GAN model

As shown in Figure 1, the training of GAN model can be divided two stages: (1) **Phase 1**. Fix "Discriminator D" and train "Generator G." Using a good discriminator, G keeps generating "fake data" and letting D judge. In the first step, G is an initial state, which has not been trained, and is detailed data, so it is easy to be judged as "false" by the discriminator. Then, since the trainer continuously trains G, G's camouflage ability will continuously improve until G can deceive D discriminator through camouflage. At this time, D is in a situation where it cannot distinguish the truth and falsehood of G, and the probability of determining whether the data is false and true is 50%. Therefore, discriminator D becomes unable to answer. (2) **Phase 2**. When we finish the first step, instead of training the generator, we will do the reverse training, completely fix the generator G, and then use G to train the Discriminator D. In this way, through our continuous training, "Discriminator

D" once again improves its discrimination ability and can better match the "generator G" trained in the first step, which can better judge the real and fake images.

Finally, by repeating the first and second steps, the generator and the discriminator improve each other, perform game training in the training, and finally form the "generator G" that achieves the optimal Nash equilibrium and obtains the accurate target picture synthesized after many tests.

3. Representative GAN Methods

In the early days, GAN gave rise to many popular architectures such as DCGAN, StyleGAN, BigGAN, etc. With the development of GAN, the processing of images is no longer limited to improving the clarity of images. The development of StackGAN, Pix2pix, Age-CGAN, and CycleGAN also increases the conversion of images in the text domain and the conversion between different image states, making GAN more applicable in the field of art. In their paper, Radford et al. proposed a model called DCGAN, which addresses training instability along with mode collapse and internal covariate transformation by making some architectural changes. These elements significantly improved the application of GANs in computer vision. It proposes a fundamental change in the framework structure, which improves the model's instability and internal collapse in the training process, which significantly improves the application of Gans in computer vision [3].

BigGAN was first proposed by a Google intern and two researchers from Google's DeepMind division [4]. However, this is the first time a GAN has generated high-quality, high-definition images. The images generated by BigGan were greatly improved in background processing and person details, making the photos closer to the actual scene. This was due to the Large Scale described in the Large-Scale GAN Training for High Fidelity Natural Image Synthesis, which adopts a Large Batch in the Training. It has reached 2048, a considerable value, and has become more prominent in the convolution channel. In addition, the parameters of the network have become more and more, and the parameters of the whole network have reached nearly 1.6 billion in the 2048 Batch, which is why BigGAN is called BigGAN. As shown in the left image, the noise vector z is split into many small pieces in the first step, and the second step generates connections between them and the condition label C and sends them to each layer of the generative network. In this way, for each residual block of the generated network, the structure of the correct graph can be further expanded. Next, we can see that the condition label C under the residual block and the small block of noise vector z are fed into the BatchNorm layer after concat operation, where this embedding is the shared embedding, linearly projected to the bias and weight of each layer.

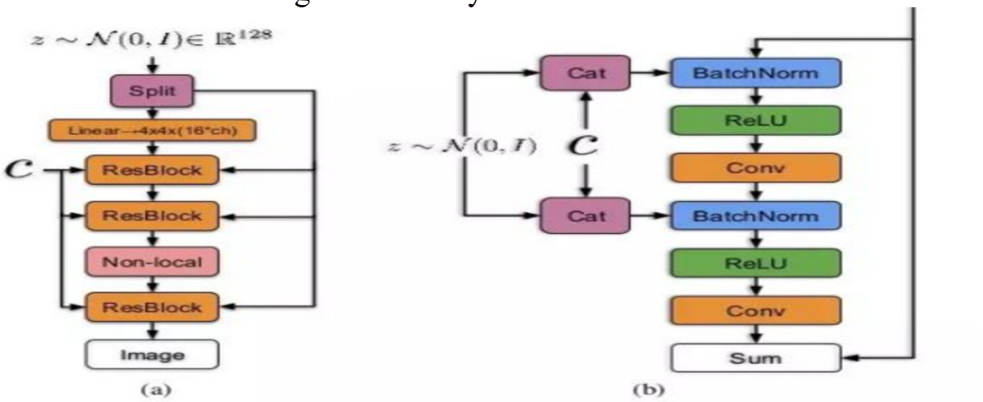

Figure 2. (a) A typical architectural layout for BigGAN's G; details are in the following tables. (b) A Residual Block (ResBlock up) in BigGAN's G.

The enhancement of the discriminator model has led to many improvements in the structural framework of GAN. These improvements allow the discriminator model to become better, and the discriminator will then assist the program in generating more realistic and sharper images. However, generators have been largely neglected and remain an area worth exploring. For example, the sources

of randomness used in synthetic image generation still need to be clarified, including the amount of randomness in the sampling points and the structure of the latent space. The most striking example of this limited interpretation of generators is that there is usually very little control over the generated images. In general, few tools are available to continually adjust the properties of the generated image, such as style. Many advanced features are covered here, such as background and foreground, as well as more precise detail instructions, such as the synthesized object's characteristics or the theme's necessary styling. To do this, we must separate features and properties in our target image and load these unique properties into the generator's model.

StyleGAN is an expanded form of gradually learning growth-type GAN, a rapid and effective method for training generator models [5]. It can cooperate with small to large discriminators in the training process and then synthesize high-precision and fine-grained images by increasing the extension of generator models. The model's growth during the continuous training process is an essential feature of StyleGan. In addition, StyleGan will change the overall architecture and structure of the generator. The StyleGAN generator goes from a point in the latent space as a separate input to a single independent image combined from two new random sources: a mapping network with independent features and a noise system. The output of the mapping network is a vector that will have a uniform style specification for each point in the new layer definition generator model through an adaptive power normalization model. If you want to change the state and details of the generated image, you can use the vectors mentioned above.

On the other hand, we can use the various points of the model to introduce noise and add some unexpected changes. When there is noise in the entire mapping, we can make the model interpret the style changes in various ways. This high degree of combination allows the image to present the clarity and precision we want.

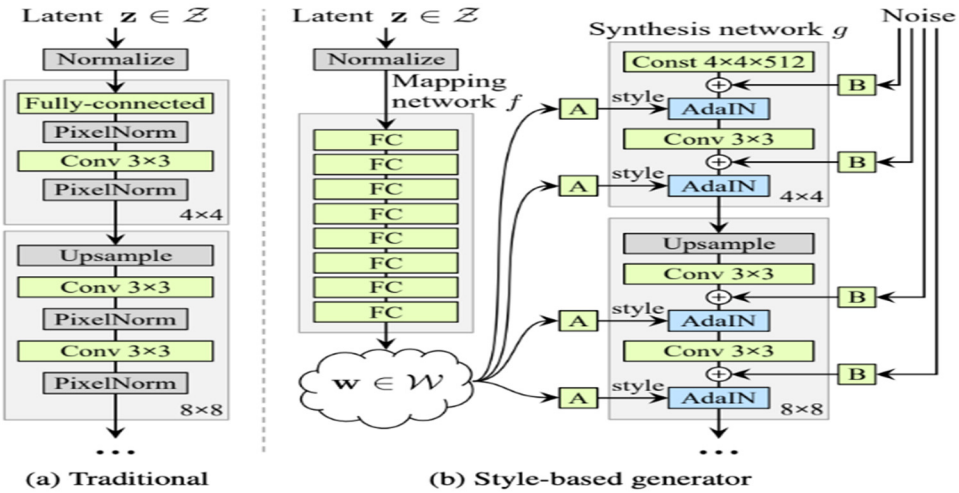

Figure 3. Comparison of generators between Traditional GAN and StyleGAN

As Figure 3 shows, typically, a traditional generator only needs to provide the necessary code for the input layer. In the first step, the code will be mapped into a hidden space EW located in the middle. Each convolutional layer will control the generator using Adaptive Instance Normalization (AdaIN). Gaussian noise was added after each convolution before computing the nonlinearity. Here "A" represents the learned affine transformation, and "B" applies the learned per-channel scaling factor to the noisy input. The mapping network F consists of 8 layers, and the synthesis network G consists of 18 layers, each with two resolution layers (4^2-1024^2). Similar to Karras et al, a separate 1*1 convolution is used to convert the output of the last layer to RGB. Our generator has 262M trainable parameters. In contrast, the conventional generator has only 23.1m. The StyleGAN generator and discriminator models are trained using the growing GAN training method. This represents both models starting with small images, in this case, a 4×4 rendering. The model cooperates through

simulations until it becomes more stable, and the generator and discriminator begin the next run. They grow to twice the height and width (and four times the area), for example, 8×8. By adding a new block to each small model, we can get a larger image, and the color is slowly erased during the training process. Erase color. After repeated training on the model, open another until our model becomes more and more stable. After that, the trainer will constantly become larger the size of the image in the process of repeatedly, until the size of the image reaches our target size in our budget (for example, 1024 x 1024), and then we can get more precise, more detailed and directional feature pictures.

When StackGANs were proposed, people applied them to the conversion from text to images, which significantly promoted the development of GAN in the field of vision machines [6]. StackGAN is composed of a set of generators and discriminators. When we input text as a target into StackGAN, we can get the described target image through calculation. The latent space will provide an input cut point to the generator model, which will compute and output the target image. At this point, the discriminator trains the model, which uses deep learning to distinguish between authentic images in the data and fake images simulated by the generator model. The two models play a two-way game while reaching Nash equilibrium during training. Finally, CycleGAN [7] is applied to transform different images. The typical approach is to combine pairs of model data and train them to convert between images. You can have an X with many data images (e.g., black and white images) and a dataset with target images that help transform the original image to produce the target output Y (e.g., color images). However, these datasets that can generate images of multiple scenes are expensive and challenging, for example, various images under specific target conditions. However, many copyrighted images cannot be included in the dataset in many specific cases. Therefore, this requires a training technology system to complete the conversion of image states or tones when the established target image is unavailable. Specifically, two utterly unrelated image datasets can be used from which target-related features can be extracted for image translation. For example, many spring photos with background images independent of content and a large number of autumn pictures with background images independent of content are taken at the first location. A specific photo is converted into a group with the target photo. This is known as the unpaired image-to-image translation problem.

For the function of transforming one image into another image, Pix2Pix [8] also has excellent results and wide applications. Pix2pix, the new generation in the GAN family, does the job perfectly, converting colorless images to colored images, spring images to fall images, sketches to oil paintings, etc. Pix2Pix is a way to transform conditional images into frame images using conditional GAN objective and reconstruction loss, for example, converting a red photo to a green photo. If you want to complete the transformation between images has been the concern of others. In general, you need to do it for the target task usually requires a specific model for a given dataset or transformation task. Pix2Pix GAN has been a popular image transformation method in recent years. It is an upgraded version of the primary conditional generation network that, given an image in the input dataset, generates the desired target image. Under these conditions, Pix2Pix GAN changes the loss function so that the generated image is both reasonable in terms of the content of the target domain and a reasonable transformation of the input image. For example, the conditional GAN target X, output image Y, and random noise vector Z of the observed image are as equation (1):

$$\mathrm{LGAN}\ (\mathrm{G}, \mathrm{D})\ = \mathrm{Ey}[\log \mathrm{D}(\mathrm{y})] + \ \mathrm{Ex}, \mathrm{z}[\log(1 - \mathrm{D}(\mathrm{G}(\mathrm{x}, \mathrm{z}))] \tag{1}$$

(The role of G is to minimize the reduction of this objective, and the role of D is to maximize the expansion of this objective, that is, G*= arg minG maxD LcGAN (G, D)). The end goal is as equation (2):

$$G * = \ arg\ min\ G\ max\ D\ LcGAN\ (G, D)\ + \ \lambda LL1(G) \tag{2}$$

The images in the dataset are used to input the target image to the generator model, and then the image is translated. Furthermore, the discriminator will judge the actual image and the input image and give the discriminator's judgment of the picture's authenticity. Finally, the discriminator is fooled

by the model generated by the trainer to minimize the loss and noise of the generated image and obtain the most straightforward and close to the established target picture. Pix2Pix GAN will be rehearsed on image datasets before and after image output. This general architecture allows the Pix2Pix model to be trained for various image-to-image translation tasks.

Age-can (Age Conditional Generative Adversarial Networks) is mainly used for facial Age adjustment and face recognition. In addition to its application in art, medical and other fields, it can also be used to find missing people. Every year, there are a large number of missing people of different ages around the world. If Age-GAN can be applied, it can significantly improve people's safety levels. Grigory et al. [9] proposed Conditional GAN-based facial Aging in their paper "Face Aging with Conditional Generative Adversarial Networks".

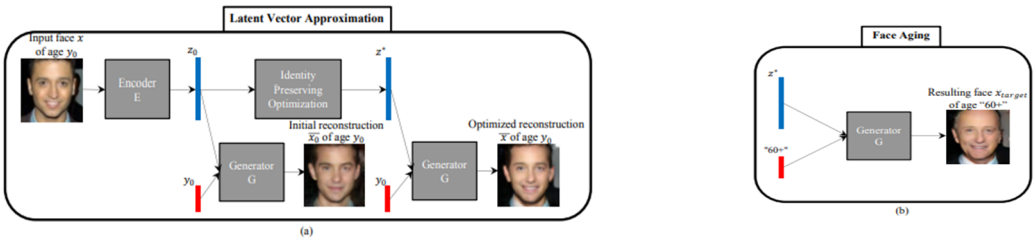

Figure 4. From the figure shown above, we can understand the specific way to change the age of the face. (a) Reconstruct the input image using the hidden vector; (b) Cut into generator G and enter the target age condition to perform facial age change.

After acGAN training, its first step is to ensure that the optimized identity finds a hidden vector z_star with the best data. Then, with the support of this vector, we will use this data to recombine and generate a brand-new image with a face, which we will define as X_bar, which will be very close to its original image x, and we will assume the age label is: y_0. After doing this, we will use acGAN to feed this hidden vector z_star and y_target with the target age label into the generator so that it can generate a face image that is consistent with the target age. Compared to the traditional GAN, the architecture of acGAN is very similar, and the following function (3) can represent its training process:

$$\min \theta G \max \theta D \, v(\theta G, \theta D) = Ex, y \sim pdata \, [\log D(x, y)] + Ez \sim pz(z), ye \sim py \, [\log (1 - D(G(z, ye), ye))] \quad (3)$$

Where theta_G and theta_D are G and D parameters, y is the additional label for training set x (condition for x). In this project, y is a six-dimensional one-hot vector for six age categories. There is a generator and a discriminator for the workflow of GAN in GAN. The GAN can take an arbitrary distribution as an input generator, generate fake data samples (be it images, audio, etc.) to fit the actual data distribution, and try to fool the discriminator. On the other hand, the discriminator needs to continuously calculate the probability of generating an actual image and make a judgment on the authenticity of the sample image. Both the generator and discriminator are neural networks, each updating the network reverse based on the results returned. They compete in the training phase to play a zero-sum game. After repeating the above steps repeatedly, the functions of the generator and discriminator will be optimized. Updating the network in reverse based on the results returned, respectively. The dynamic changes eventually reach the Nash equilibrium.

4. Experiments

The title of BigGAN not only indicates the vastness of the network but also implies that this article will give people a great impression. Of course, such a Blindly increasing the batch size and network parameters do not achieve significant improvement. In SAGAN, the batch size is 256. As you can see, if you want to achieve better and faster capacity improvement, you can increase the batch size. The experimental results are shown in Table 1. Here, batch stands for Batch size, and Param shows the sum of parameters. Ch represents the channel multiplier for the number of units in each layer.

Shared is using Shared embeddings. Skip-Z is using skip connections from latent layers to multiple layers, and Ortho is orthogonally regularized, indicating whether the setting stabilizes to 106 iterations or collapses at a given iteration. Except for lines 1-4, all results are computed with eight random initializations.

As shown in Table 1, batch size increases by 8 resulting in a 46% improvement in the generated IS. This may result from covering more patterns per batch, increasing the number of channels by 50% each time will provide better gradients for two different networks, roughly twice the number of parameters in both models. This would further increase IS to 21%. In addition, BigGAN improves the embedding of the prior distribution Z. Unlike GANs, which will use z as input to embed directly into the generative network, BigGAN will use the noise vector z to send to multiple layers of G instead of just the single initial layer. Conditional generation of BigGAN is achieved by splitting Z into a block by resolution and concatenating each block to the conditioning vector C, which provides a performance boost of ~ 4% and speeds up training by 18%.

Table 1. Frechet Inception Distance (FID, lower) and Inception Score (IS, higher is better) for ablations of our proposed modifications

Batch	Ch	Param(M)	Shared	Skip-z	Ortho.	Itr×10³	FID	IS
256	64	81.5	SA-GAN Baseline			1000	18.65	52.52
512	64	81.5	F	F	F	1000	15.30	58.77(1.18/-1.18)
1024	64	81.5	F	F	F	1000	14.88	63.03(1.42/-1.42)
2048	64	81.5	F	F	F	732	12.39	76.85(3.83-3.83)
2048	96	173.5	F	F	F	295(18/-18)	9.54(0.62/-0.62)	92.98(4.27/-4.27)
2048	96	160.6	T	F	F	185(11/-11)	9.18(0.13/-0.13)	94.94(1.32/-1.32)
2048	96	158.3	T	T	F	152(7/-7)	8.73(0.45/-0.45)	98.76(2.84/-2.84)
2048	96	158.3	T	T	T	165(13/-13)	8.51(0.32/-0.32)	99.31(2.1/-2.1)
2048	64	71.3	T	T	T	371(7/-7)	10.48(0.1/-0.1)	86.90(0.61/-0.61)

Taking the classic image-to-image translation task as an example, we further report the visualization results of different GAN-based algorithms, which can be found in Figure 5.

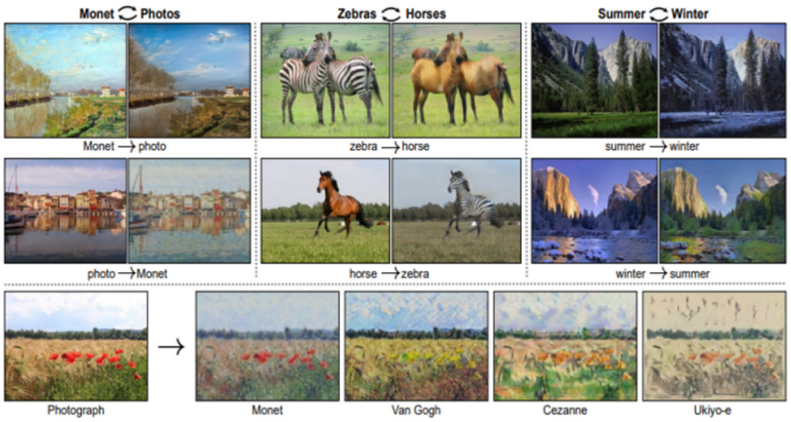

Figure 5. cycleGAN image-to-image conversion. Monet paintings and landscapes from Flickr on the left, horses and zebras from ImageNet in the center, and summer and winter landscapes from Flickr on the right

We need a comprehensive paired dataset to train an image-to-image translation. For example, an input image X contains several datasets (e.g., a spring sketch) and a dataset image with the desired target orientation that can be used as the output image Y (e.g., a fall oil painting). However, too high requirement of pair of training data set, the problem in the above explanation, for example, the world-famous paintings and no copyright images, will be an enormous challenge, the problematic solution is usually only data chart, used in the general characteristics of the two groups, language translation, and complete the picture has no connection between the translation and conversion. For the image-to-image translation task.

5. Application for NFT

Non-fungible tokens, or NFTS, are blockchain records associated with a specific digital or physical asset. In the blockchain language, the full name of a virtual Token is Crypto Token, often referred to as Token or Token. According to different functions and properties, coins can be further divided into three types, which are payment tokens (mainly used for payment, with functions similar to legal currency, common payment tokens include bitcoin (BTC), Ether (ETH), Binance (BNB), etc.), functional tokens (users who own this token can enjoy services or special rights brought by the token, common functional tokens include ADA and XRP) and asset tokens (the role of asset tokens itself is to be saved as a kind of digital asset, just like gold and antiques, etc., which are collected because of their value. The NFT introduced in this paper can be regarded as an asset token). NFTS is one of the three types of tokens. Its characteristics are irreplaceable, indivisible, and unique, a total of three characteristics. NFTS cannot be broken into several pieces for trading like bitcoin, and their irreplaceable nature makes each NFT unique.

Table 2. Difference between Homogenized and Non-homogeneous tokens

Homogenized tokens	Non-homogeneous tokens
Substitutability	Not substitutable
divisible	not divisible
consistency	uniqueness
payment and transaction functions	anti-counterfeiting function and can be used as a digital collectible with circulation

When NFT is art, it has a broader and novel art form, not restricted to the form of pictures, can be GIF or image, and has also developed into a programmable art form. Moreover, NFT has an open and effective art value evaluation platform worldwide. Usually, the pricing of art is controlled by the private interest of buyers and the transaction price of the free market. However, the difference is that NFT has a broader audience, which can form a discussion on new materials and art, and has more collection value than the times. Furthermore, this easy way allows anyone to create their NFT with almost no coding skills required. This has dramatically increased the spread of NFTS and has made itself a good advertising effect, which has been well promoted in the art world, NFT works generally refer to digital files, such as audio, images, and videos., which will protect the rights and interests of NFTS, but the legal rights and interests of NFTS need to be improved due to the short development of NFTS so far. However, although the current law does not have clear regulations on the ownership of blockchain and NFT, nor does it affect their ownership of digital files, intellectual property rights are protected.

To turn the file into an NFT [10] artwork, the first step to transforming the images generated by us into NFT is establishing unique identifiers. Unique identifiers can be defined for both virtual and physical objects. Virtual objects themselves are digital, and digital fingerprints can be directly generated. A digital fingerprint can be generated using descriptive information (such as the manufacturer, date of manufacture, scanned documents, photographs, etc.) as digitized content.

Moreover, we will create the Smart Contract. Smart Contracts can be developed based on different public chains. It is open to more than one public chain. The implementation of an intelligent contract

scheme is different for different public chains, which will be responsible for tracking the owner of the NFT in order to display the owned NFTS and subsequent sales (financial transactions take place on the website of the NFT marketplace, such as Opensea [11]) embedding a link with blockchain hosting in the token, put the work on the NFT market so that we have a series of works of art that have been deeply thought and done by the computer.

So convenient produce works of art, of course, and cancel the art dealer, wider circulation of works of art, let more people involved in the creation, let NFT between the financial sector not only popular, also let it received a lot of attention in the field of art, but at the same time, also received a lot of questions about NFT, for many existing work transform painting style, or will be painting photography and video processing of NFT, when a creator has been questioned for a non-original, independent and computer generated art power is defined as the art of buying and selling, whether to impact of human civilization and art, actually, for that matter, we always keep an open attitude, in the NFT early yuan "new life" of the universe, there are still many questions we need to improve, I as a scholar very support the combination of art and technology, science and technology should be as part of the art of civilization, will not only promote the development of its, can let more people have more art theory of reflection.

6. Conclusion

Generated against the network as a focus of current famous computer vision, throughout the development history of neural network technology, from the deep learning of computers to the development of artificial intelligence, game theory ideas have played a significant role in this field. There is also a wide range of development space in the field of neural networks in the future—for example, a lot of game development and medical equipment system upgrades. An excellent example of the development of combining game theory with deep computer learning is the GAN introduced in this paper, which takes machine learning and artificial intelligence to the next level through Nash equilibrium in game theory, which has significant development in the field of image and vision. It has developed many kinds of edit image processors, not only in improving the sharpness of the image has many ascensions. It can generate specific images from one sentence of data, adjust the picture's details, change the picture's style, easily change the age of the face, and has breakthrough applications in the medical and art fields. NFT as an overall development of the decentralized network, NFT has unlimited possibilities, and potential NFT has opened up a new way for blockchain technology. By replicating physical assets in the digital world, NFTS has the potential to become an essential part of the blockchain ecosystem and the broader macroeconomy. In the field of NFT, digital art is developing rapidly. When GAN technology is combined with NFT, the identities of humans and machines are about to be exchanged. When machines create independently, humans are only additional tools. To a certain extent, this violates the ethical issues of humans, machines, and art creation, but this is the trend of development in the era of science and technology.

References

[1] Goodfellow I, Pouget-Abadie J, Mirza M, et al. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.

[2] Bacharach, M. (1989). Zero-sum games. In Game theory (pp. 253-257). Palgrave Macmillan, London.

[3] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[4] Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

[5] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4401-4410).

[6] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 5907-5915).

[7] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).

[8] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).

[9] Antipov, G., Baccouche, M., & Dugelay, J. L. (2017, September). Face aging with conditional generative adversarial networks. In 2017 IEEE international conference on image processing (ICIP) (pp. 2089-2093). IEEE.

[10] Kugler, L. (2021). Non-fungible tokens and the future of art. Communications of the ACM, 64(9), 19-20.

[11] Jeong, S. Y. E. (2022). Value of NFTs in the Digital Art sector and Its Market Research (Doctoral dissertation, Sotheby's Institute of Art-New York).