Gans Used to Generate 2d Game Resources

Xintong Liu

School of Computing and Data Science, Xiamen University Malaysia, Sepang, Malaysia

Abstract. Game Assets Generation (GAG) is to create all kinds of resources a game need, including game pictures and level design. Here we focus on the game pictures. With the development of machine learning, the generation of game assets may not be created by artists by Unity Technologies, Unreal Engines or other tools. Different types of Generative Adversarial Network (GANs) can be used generate 2D game resources. In the article, based on wide references, GANs are separated into 2 basic categories, which are direct generation GAN and Indirect generation GAN. And for game assets generation, there are different missions. The basic generation of game assets, texture generation, different angles generation and style transfer. Based on the logic, 4 typical GANs of precedents are introduced and compared in algorithms and effects in detail. Since different GANs are suitable for different missions, the most suitable public datasets are chosen to evaluate the performance of each method. Additionally, problems of the algorithms are discussed, following with some future research directions.

1 Introduction

With the usage of systems to learn from existing data by Machine Learning (ML) to product digital content and the development of GAN [1], picture generation attracts numerous researchers. Along with the largely growth of video game players from 2017 to 2019 [2], the need in the game industry is also increasing. Therefore, Game Assets Generation has become a significant research hotspot. Game Assets Generation (GAG), defined as Procedural content generation (PCG) in the game industry [1], aims to generate a large number of items of different art styles for different games, including characters, things, maps and etc. Researchers have created game resources datasets like Unity Asset Store [3], Tiny Hero [4]. Meanwhile, by constructing GANs like DCGAN, StyleGAN enhances the effect of GAG. If in the future, the researchers manage to create more effective GANs for creating game assets, the cost for game companies will be significantly lower and thus become more lucrative. Without sacrificing game art quality, even with better game assets quality. GANs will be widely used to generate game assets. This article paves the way for it.

In general, the GANs used for GAG are divided into 2 groups. They are direct generation GAN and Indirect generation GAN. For each GAN, because the algorithm usually different, the area of expertise for each GAN is diverse. Thus, the paper divide

DMT2309230@xmu.edu.my

GAG missions into the four categories below, and for each of them there is a typically recommended GAN:

The mission, basic generation of game assets, is about creating art form 0. Hence, the focus of the mission is not generating perfect game assets but providing ideas or fundamentals. With further selecting or flourishing, the final version of the game can be produced. For the basic generation of game assets, DCGAN and StyleGAN can be considered. The key of this mission is to generate reasonable and real pictures.

The definition of texture generation is the creation of surface map. Often applied in generating character skins, environment material and textures or terrain to enrich the visual effects of the game. The essential of the mission is style and quality. The style in one game should be the same and the picture quality should be high. Correspondingly, the GAN which can do well in the task requires the following characteristics. Controllability and Tileability [5]. Controllability: Texture generation requires certain parameter control, such as color, detail density, style, etc. Tileability: Textures often need to be tiled, especially for game backgrounds or floors. The textures generated by GAN must also be seamless so that they can be reused. The best performing model in this regard is StyleGAN.

Different angle variation means generating poses of the character at different angles. In actual production, changing character from one pose to other poses is time demanding and costly. There is a way to build a 3D model to get pictures from different sides. However, a 2D game company rarely do that. Therefore, finding a GAN to satisfy different angles generation can save a lot of time and effort. Pix2pix (based on cGAN) have a good performance in doing this kind of mission. The key in this mission is to reduce the difference of characters between their different poses.

Style transfer defines remain the content of the picture and change the vision style. Its core target is to maintain structural information while change color, texture and other style features. In GAN-related style transfer tasks, StarGAN is a typical multi-domain style transfer model that can convert images between multiple styles without training multiple independent models. According to research, a specified StyleGAN, DualStyleGAN [6], can also do style transfer well.

Around the tasks mentioned above, the article makes analyses and an overview of the Research progress and current status of GAN based 2D Game Assets Generation, with a reasonable arrangement of the text based on creating a game asset from 0 to perfect. Also, with the guide of the 4 missions of GAG. Introduce the representative GAN, the limits and future direction for each specific real game production mission. The visual assessment for Game Assets Generation is also mentioned in the article.

2. Methodology

2.1 DCGAN

Aimed at the first mission of the basic generation of game assets, DCGAN (Deep Convolutional Generative Adversarial Networks) is used. The overall process consists of three main stages: input processing, model training, and output generation. Preprocessing steps includes resizing all images to a fixed size pixel imager, removing duplicates, standardizing backgrounds and Data augmentation. The dataset is split into training (75%), validation (5%), and testing (20%). The model is composed of a Generator and a Discriminator. The structure is in Figure 1:

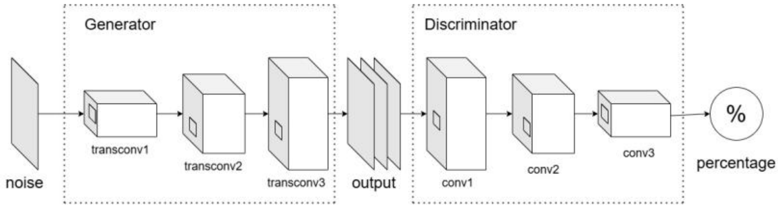

Figure 1 the structure of DCGAN

The Generator transforms noise into images using 3 transposed convolution layers. Outputs a specific pixel image. The Discriminator is a standard convolutional neural network (CNN). It takes in both real and generated images, then classifies them as real or fake. In the training process, the model uses Minmax loss function. In addition, Adaptive Discriminator Augmentation is applied to improve stability. The model is trained by mini-batches including 16-32 images each. After training is the output generation procedure, the system takes new random noise as input into the generator and outputs novel and real basic game assets. Here are the Pros of DCGAN, compared to traditional GANs, DCGAN uses convolutions and batch normalization, preventing mode collapse and improving training stability. CNN is utilized to enhance image quality and sharpness. While the DCGAN has some cons. DCGAN is lack of the capability to produce high-resolution images: DCGAN is primarily used for low-resolution images, for example, 64×64. Struggling with high-resolution image generation. Additionally, the Mode collapse issue: In some datasets, generated samples are probably extremely similar, leading to a lack of diversity.

2.2 StyleGAN

DCGAN can only generate the basic image. Nowadays, a game cannot success with high quality textures on the game assets. Therefore, StyleGAN is needed for the second mission, texture generation. The structure of StyleGAN is like Figure 2:

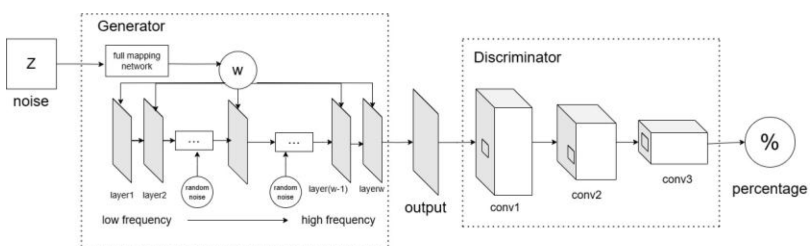

Figure 2 the structure of StyleGAN

The input for StyleGAN is a random noise z. z is transferred into an intermediate latent space w via the Mapping Network. The multiple dimensions of w are used to control the generation process of different layers. The generator, Synthesis Network [7], starts generating image from the low frequency space. In low frequency space, the rough shape is created. High frequency details, including contours, colors, lights, are added into the image controlled by w layer by layer. Eventually, a high-resolution image is generated as output. And the discriminator used in the training procedure is a CNN, same to DCGAN. Additionally, StyleGAN include Adaptive Instance Normalization (AdaIN) to control the styles of different layers, hence, the characteristics of different scales can be adjusted independently. Random noise is also introduced into the system, makes the generated image more natural. There are enhanced versions of StyleGAN, such as StyleGAN2, StyleGAN3 [7]. They may have better performance but basic logic is similar. Here are the advantages of StyleGAN, StyleGAN has strong style control capabilities (style mixing, style transfer). It also has the ability to gradually generate high-resolution images to improve quality. The potential spatial decoupling of StyleGAN is better, which improves editability. The disadvantages are also listed, Drop-like artifacts affect image quality (corrected in StyleGAN2). Progressive growth leads to loss of position invariance, affecting detail control (StyleGAN2 adopts a new architecture and improves it). For StyleGAN, the high-resolution details are insufficiently utilized, and the generated images still have a little blur (StyleGAN2 improves by increasing network capacity).

2.3 Pix2pix

After creating different basic characters by DCGAN, and flourished by StyleGAN, a refined game character is created. However, in the game, game companies need the character to change his (or her) actions or poses. The companies do not need to build it again. Pix2pix can help to change characters poses or actions, which is a good solution to the third mission, different angle variation. The procedure is still data input, model training and output generation. Figure 3 is the basic structure of the Pix2pix system.

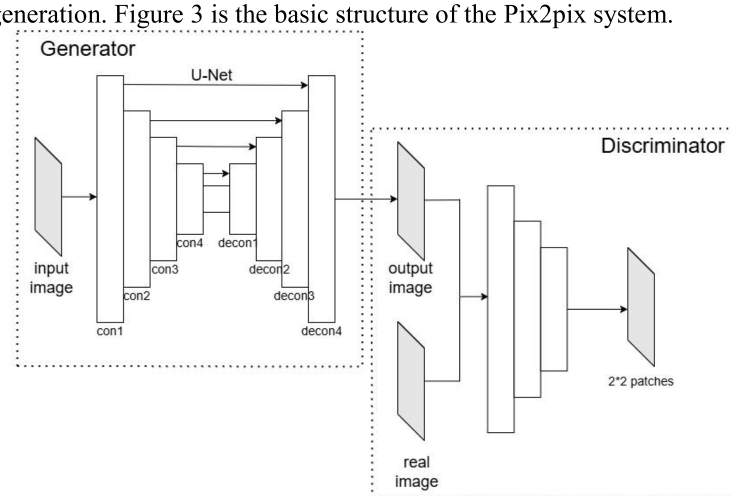

Figure 3: the structure of Pix2pix

The system gets an input image with tag (based on cGAN). By adopting generator based on U-Net, the system processes the input image by encoding and decoding (convolution and deconvolution). In addition, Skip Connections are used to maintain special information, ensuring the integrity of details of the input. The Pix2pix system then produce the output, image of the input character that match the target pose (e.g., transition from frontal to side view). In the training part, the output is put into the discriminator with real images to enhance the performance of the system with the help of Adversarial Loss. The system also adopts L1 loss to ensure that the generated image is close to the real target image and reduce blurs. The system uses PatchGAN as discriminator, its main function is to distinguish local areas rather than to classify the entire picture globally. Therefore, improving generation effect. The merits of Pix2pix system include high-precision image-to-image conversion and the retention of details by using U-Net Structure. While the demerits of Pix2pix are here, it relies on paired data, which is hard to prepare. Low generalization ability to unseen poses. High computing cost with low efficiency because for each pose, a model is needed to train.

2.4 StarGAN

As for the Pix2pix, the most crucial problem is the lack of efficiency of style transfer. Actually, the character poses variation belongs to style transfer, the fourth mission. The method StarGAN can change that. Here is the simple Schematic diagram.

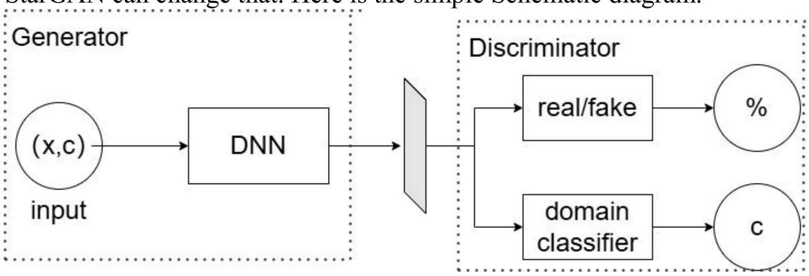

Figure 4 simple Schematic diagram of StarGAN

As in the Figure 4, the input x is tagged with a style c. Then, it is put into a DNN. DNN processes it. Features are extracted through convolution (down sampling) to reduce resolution. The style is transformed through Residual Blocks (ResNet Blocks) to learn how to modify specific properties of the image. Finally, deconvolution (up sampling) is used to gradually restore the original image size to generate a high-quality image. After that, a new image with targeted style is created. When training, the discriminator returns the percentage of real. It also returns the style to see whether the image is in the targeted style. With the help of U-Net and 3 Losses including Adversarial Loss, Domain classifier Loss and Reconstruction Loss [8,9]. The StarGAN manage to realize style transfer. The advantage of StarGAN includes the following. Firstly, high efficiency. Multi-domain translation is realized in one model. The generated images are more controllable by using domain labels. It can also be trained on multiple datasets to improve the generalization ability. However, StarGAN has the following disadvantages, poor generalization to unseen domains and lower quality for complex Features than GANs such as StyleGAN.

3 Experiment

The evaluation of the effects of image generation is a little subjective. Therefore, these conditions are adopted to evaluate the algorithms respectfully for each corresponding task.

Quantitative indicators:

As with many ML problems, the progress of image generation methods hinges on good evaluation metrics. One of the most popular methods is the Frechet Inception Distance (FID). FID estimates the distance between a distribution of Inception-v3 features of real images and those of images generated by the algorithm. Because the number of researches on GAG are limited, therefore, GANs can only show the effect of the GAN on specific task. Instead of comparing with each other. And for style transfer problems we choose Classification Error as our standard. Classification Error is usually defined as the probability or proportion of incorrect classification in a classification task. [10] It can evaluate the effects of style transfer.

Qualitative evaluations:

Observer study results including precision and recall. Visual comparison including images generated by GAN compared to target images.

For DCGAN, which is suitable for basic generation of game assets, here is an experiment which is about generating fantasy and sci-fi style game signs, also adopting Adaptive Discriminator Augmentation to improve the training part of the GAN. In this study, the scientists use FID to evaluate the effects of generation, the best model's FID is about 22.2. Additionally, there is an observer study to further evaluate the performance of the DCGAN. After mixing up the generated images and the real images, the writers let 50 observers to distinguish those generated by AI from those drawn by artists. And the result is that 68.7% of the pictures generated by AI are evaluated as real pictures (or drawn by hand). While 69.9% of the real pictures are recognized as real pictures [11], according to Figure 5. The difference is slight. In conclusion, the performance of using DCGAN to generate basic game assets is outstanding.

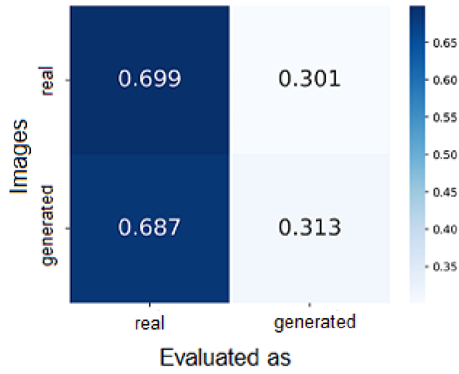

Figure 5 DCGAN user test [11]

For StyleGAN in texture generation, there is an experiment on generating wood texture. Which is similar to texture generating in GAG. The FID of the result is 13 [11]. This low FID means that wood texture generated by StyleGAN is very similar to real wood texture. It is almost impossible for human to distinguish which one is the real wood. Therefore, we can say StyleGAN has a good performance on texture generation.

For Pix2pix in Different angles variation mission, an experiment of using CGAN based Pix2pix to realize image-to-image translation, generating sprites in different angles, catch my attention. The best performance dataset is Tiny Hero, with a FID of 0.092, which is almost the same as the real characters in the specific pose [9]. This shows the excellent effects of using pix2pix to generate pixel art assets. Moreover, there is also a visual analysis in this experiment. As in Figure 6, on the left is the best performed dataset. We can see the generated character in another post is similar to the Target.

Figure 6 visual analysis of Tiny Hero and test examples [4]

According to the reference, there is an experiment on the RaFD dataset, StarGAN generated face images with different expressions including being sad, happy, angry and etc. It compares the classification error with CycleGAN and IcGAN doing the same task, the result shows that StarGAN generates the most realistic expressions (with the lowest error rate of 2.12%), outperforming CycleGAN and IcGAN.

Table 1 Different models and their classification error compared with StarGAN

model	Classification error
StarGAN	2.12%
CycleGAN	5.99%
IcGAN	8.07%

This shows that the expression images generated by StarGAN are more consistent with the target category and have higher quality. Therefore, StarGAN performs good in style transfer mission.

Table 2 Experiment results for different GANs

GAN (direct/indirect)	Mission type	FID (best model)	Real mission
DCGAN(direct)	Basic generation of game assets	22.2	Generating game signs
StyleGAN(direct)	Texture generation	13	Generating wood texture
Pix2pix(indirect)	Different angles variation	0.092	Pixel character pose transfer
StarGAN(indirect)	Style transfer	/ (classification error is 2.12%)	Expression transfer

The data of the 4 experiments verifies that the specific GAN performs good in the relative mission. In Basic generation of game assets, DCGAN performs well, but is not be as good as other models in detail control and style consistency. StyleGAN is suitable for texture generation and the generated effect is usually of higher quality. Pix2Pix (cGAN), which relies on paired data, suitable for different angles variation mission, and can better maintain structural consistency. StarGAN has outstanding multi-domain conversion capabilities and is suitable for style transfer tasks.

4 Discussions

Game market is expanding fast. The Generation of Game Assets with the help of GANs may significantly improve efficiency. The article will discuss problems to be solved and future research directions.

4.1 Computational cost and training efficiency

GANs have high computational cost, especially high-resolution generative models such as StyleGAN. It has high demand for computing resources and are not suitable for small game teams or real-time generation scenarios. Therefore, methods to reduce the computational resource requirements of GANs to make them suitable for game development pipelines can be a future research direction.

4.2 Computational cost and training efficiency

It is said that the world is run out of data. Though internet has an enormous number of data on it, data is still not enough. But all GANs need a lot of data to train. Sometimes it is hard to find a wanted dataset. Pix2pix relies on paired data, however, in the task of game asset generation, it is very difficult to obtain enough paired training data. While StarGAN requires style labels to learn style transfer and is prone to overfitting when there is insufficient training data. Our mainstream way now is to label the datasets artificially. The cost is too much. The future research direction is if it is possible to combine Few-Shot Learning or Self-Supervised Learning to train GAN with limited data.

4.3 Text driven GAN combined with LLM (Large Language Model)

Nowadays, most GANs take images as input. The meaning contained by an image is far less than the meaning contained in a paragraph of texts. Also, in the game design documents, the needed picture or game assets are mostly described by texts in detail. There are a few AIs or GANs can generate photos based on texts, for example, StyleGAN with CLIP. However, the effect is not that good. The current text-based game resource generation is not very effective, mainly due to the lack of sufficient prior knowledge. (Including text-image semantic understanding, game style definition, generalization ability) GAN can be connected with LLM. LLM can provide a deeper understanding of semantics and improve the accuracy of GAM image generation. Texts input as details are called prompt. Users can modify the prompt through conversation, while the LLM intelligently adjusts the text. The LLM can help optimize the prompt to better suit the current image generation model. Therefore, connecting GAN with LLM can be a future research direction.

5 Conclusions

Focusing on the task of using GAN to generate 2D game assets, this article connects the four most influential tasks in GAG and their representative methods in a sequence from creating game resources from scratch to a perfect game resource. Additionally, the basic procedure, the advantages and disadvantages of each GAN are discussed in the article. For each method in each mission, one referenced experiment is discussed. Problems like computational cost and data issue are mentioned to be solved, and text driven GAN combined with Large Language Model (LLM) can be a future research direction.

References

1. Fukaya, K., Daylamani-Zad, D., Agius, H.: 'Intelligent generation of graphical game assets: A conceptual framework and systematic review of the state of the art', arXiv preprint arXiv:2311.10129, 2023.
2. Palma-Ruiz, J. M., Torres-Toukoumidis, A., González-Moreno, S. E., et al: 'An overview of the gaming industry across nations: using analytics with power BI to forecast and identify key influencers'. Heliyon, 2022, 8(2).

3. 'Unity Technologies. Unity Asset Store', https://assetstore.unity.com, accessed December 20, 2020.
4. Coutinho, F., Chaimowicz, L.: 'Generating pixel art character sprites using GANs', //2022 21st Brazilian Symposium on Computer Games and Digital Entertainment (SBGames). IEEE, 2022: 1-6.
5. Rodriguez-Pardo, C., Casas, D., Garces, E., et al: 'TexTile: A differentiable metric for texture tileability', //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 4439-4449.
6. Yang, S., Jiang, L., Liu, Z., et al: 'Pastiche master: Exemplar-based high-resolution portrait style transfer', //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 7693-7702.
7. Karras, T., Laine, S., Aittala, M., et al: 'Analyzing and improving the image quality of stylegan, //Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 8110-8119.
8. Choi, Y., Choi, M., Kim, M., et al: 'Stargan: Unified generative adversarial networks for multi-domain image-to-image translation', //Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8789-8797.
9. Fukunaga, K., Flick, T. E.: 'Classification error for a very large number of classes', IEEE transactions on pattern analysis and machine intelligence, 1984 (6): 779-788.
10. Karp, R., Swiderska-Chadaj, Z.: 'Automatic generation of graphical game assets using GAN', //Proceedings of the 2021 7th International Conference on Computer Technology Applications. 2021: 7-12.
11. Zhan, W., Yang, Z., Xu, H., et al: 'Exploring the potential of StyleGAN for modeling high-quality and diverse digital wood textures: Towards advancements in the wood industry', Industrial Crops and Products, 2024, 209: 117880.