Engineering Universe for Scientific Research and Management
ISSN (Online): 2319-3069
Vol. XVI Issue XI
November 2024

PixelSake Deep Convolution Generative Adversarial Network

Abhishek Shrivastava[1], Vivek Gautam[2], Naman Patel[3], Sunidhi Sharma[4], Mohit Kadwal[5]
Student, Department Of AI&DS Prestige Institute of Engineering Management and Research, Indore, India[1,2,3,4]
Assistance Professor Department Of AI&DS Prestige Institute of Engineering Management Research, India[5]

Abstract

The popularity of anime as a unique and well-established art form, recognized worldwide has given a great boost to its automatic generation using machine learning. High-quality image synthesis recently became possible using GANs, which constituted an important device in its synthesis. Recent advances enable the generation of anime-inspired images with a close resemblance to reality regarding their style. Our proposed approach, PixelSake, adopts the architectural changes in both generator and discriminator combined to enhance anime-specific features using domain adaptation techniques that better capture the subtle nature of anime aesthetic. PixelSake integrates a multi-scale discriminator with feature-extracting generators for payoffs on anime-specific features, such as line art and color palettes, and exaggerated expressions, which are concentrated less on traditional GANs. The perceptual loss function with features of pre-trained neural networks has been used to improve the quality of images. It further optimizes the model's output to be close to human perception of anime.

Keywords: ***Generative Adversarial Networks (GANs),Deep Learning, Machine Learning, Feature Extraction, Character generation, Neural Networks.***

1. Introduction

Anime is a peculiar and fairly intricate form of visual art of Japanese origin. It has gained worldwide popularity because of its distinct style, including melodramatic facial expressions, stunning backgrounds, and saturated colors. In the age of lightning-fast technological advancements, the need to create pictures in the anime style increased, leading to the development of automatic systems for creating high- quality images resembling real anime. Amongst the promising techniques to generate such images in recent times, one of the most impressive ones has been Generative Adversarial Networks, the family of deep learning frameworks particularly known to produce high-fidelity images in a host of different domains, from photographic images to stylized artworks. [1] However, the process of generating anime images is different from other images synthesis tasks. The traditional GAN models, which are basically developed for realistic photography, fail to capture the stylized nuances of anime, such as precise lines, symbolic facial expressions and culturally embedded design principles. Anime lacks annotated datasets for model training and hard to enforce stylistic consistency during image generation. It calls for specifically arranged new architectures and training techniques that could be designed specifically for GANs, elaborating on the distinguishing characteristics of anime. The author then introduces PixelSake, a state-of-the-art model of GAN, which is engineered to best fit the problem of anime generation. It contains several improvements over regular GAN architectures, including a feature-extracting generator, multiples scale discriminators, and a perceptual loss function concentrating more on aesthetic features of anime. Domain adaptation is combined with specially selected data for better capturing of the style of anime as well as the maximization of quality for an output consistency. We have presented several key contributions that appear to be computer vision and AI-driven art generations Therefore, PixelSake bridges the gap of existing shortfalls in the traditional GAN architectures and represents a novel attempt at producing fully automated anime productions; such applications could be in animation, games, and virtual avatars. Based on experimental results, it can readily be perceived that PixelSake establishes superiority examples for benchmarking both quantitative and subjective quality evaluation metrics, indicating its potentials as an appropriate tool in anime image synthesis. This adversarial framework allows GANs to generate pictures, spectacularly realistic in different domains, hence promoting them as a viable solution in anime image generation. Existing GANs are primarily learned from real photographs and would usually become incapable of reproducing the stylistic characteristics of anime art. This had opened up an opportunity in researching models of GAN especially for anime to serve the unique aesthetic requirements of the latter.[2]

2. Ease of Use

For designing an Anime-Generated Generative Adversarial Network, ease of use plays a very critical role to make the model accessible and effective for various users, including researchers and developers, artists, and animators. Many considerations are involved in making the model easy to use, relating to its interface, training, customizations, or resources. The following are a few of the most important considerations followed in the development of the model:

1. The user interface and APIs: It should be developed with a wonderful user-friendly interface, keeping the ease factor if users who are not technical, like artists or anime creators. It enables the user to generate images by providing APIs and a graphical interface without the deep knowledge of coding or machine learning. High usability would be enhanced by developing it according to integration with popular platforms for image processing including Photoshop and Blender.[3]
2. Pre-trained Models and Transfer Learning: There are pre- trained models permitting users to generate high-quality anime images very rapidly without training a model from scratch. That is a major thing because it reduces both setup time and computation costs. Additionally, through transfer learning, users can fine-tune the model with the smallest computing resources over their special data set that increases flexibility but makes the model accessible.[4]
3. Stylistic Parameters: Customizable style parameters will provide the possibility for selecting key stylistic parameters for enabling users to customize aspects such as color palettes, line thickness, or character features while diversifying the anime style the GAN can produce. Adjustments are often easily allowed through the user interface, making aesthetic details easy to modify with no interference from the codes of the model.[5]
4. Compatibility with existing workflows: The model should be compatible with all the creative workflows; whether it is an animation pipeline or 3D modeling software, making it all the more usable. When it supports standard file formats and offers plugins for popular design software, it can be an excellent tool in the existing setup of an artist or developer.[6]
5. Automated Parameter Tuning: If one is not knowledgeable about the GAN settings, automated parameter-tuning will let quality results be discovered. With preset setups and automated suggestions, users may produce quality images without the fuss of individually fine-tuning a multitude of hyperparameters to make the model user-friendly to those with limited technical know- how.[7]

Finally, all of these aspects of usability: pre-trained models, friendly interface, low computational requirements, customizability features, full documentation, and applicability compatible with the workflow of this PixelSake will allow software developers to users such as anime artists to have access to it and to make it efficient for technical and non-technical users. All of these improvements in ease of use make everyone tap into the potential that GANs hold for creating anime images.[8]

3. System Requirements

System requirements are most critical when developing an Anime-Generated Generative Adversarial Network. Software- wise, PixelSake will likely run under Python 3.7 or above in combination with a deep learning framework that might be either PyTorch or TensorFlow since both support GANs. For NVIDIA GPUs, compatible versions of CUDA and cuDNN are required -cuda and cuDNN speed up deep learning computations by "under the hood" acceleration. Additional libraries available include those for handling data, images, and model assessments-supporting operations like NumPy, Pillow, and scikit-learn; add further functionality to the workflow through the manipulation and quality assessment of data. If the general hardware resources are limited in capacity, employing such cloud-based services as Google Colab provides opportunities whereby users could freely train their models without feeling more constrained by local resource utilization. This range of configurations ensures that PixelSake can be accessible and effective for a wide variety of user needs, ranging from simple experimentation to complex, production- level image synthesis.[9]

Table . 1 Software /Hardware -version disk spaces

S. No.	Software/Hardware	Version/Disk Space
1.	CPU	Intel Core i5
2.	GPU	NVIDIA GTX 1060
3.	Additional Libraries	PyTorch, TensorFlow, CUDA, cuDNN, Pillow and other necessary libraries
4.	Operating System	Windows 10, macOS 10.15+
5.	Python Environment	Virtual Environment Tools, Deep Learning Libraries, Python 3.7

Engineering Universe for Scientific Research and Management
ISSN (Online): 2319-3069
Vol. XVI Issue XI
November 2024

6.	Graphical User Interface (GUI)	Jupyter Notebooks, Streamlit, Gradio
7.	Network Connectivity	Stable and highspeed internet connection for image generation
8.	Storage	50-100 GB free space.

System Architecture

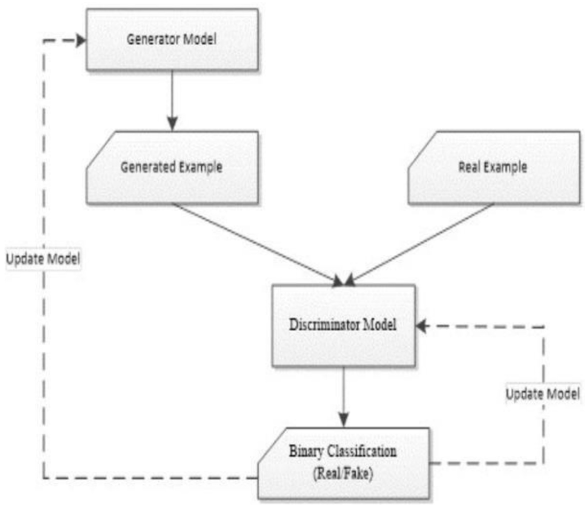
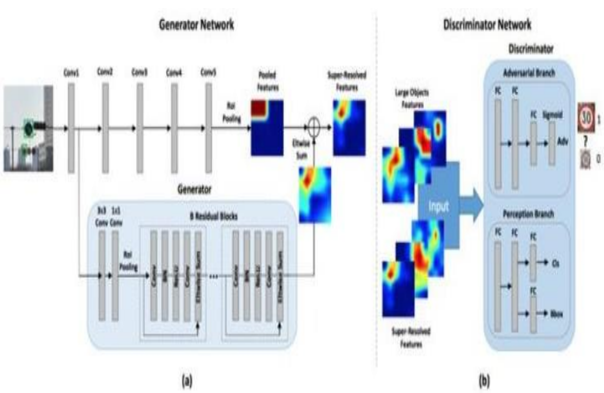


Fig. 1. Architecture Mapping



The style-of-stylized anime image creation architecture for PixelSake was based on the advanced capabilities of GANs. Several interdependent components-it includes the generator, discriminator, and support systems-participate such as processing data and managing training in such a manner that each of them performs a distinct function so that the network can learn to generate anime-style images with those desired stylistic and aesthetic feature.

4. Methodology

Such a structured methodology for the development of PixelSake consists of data collection, model design, training, evaluation, and fine-tuning. The steps involved will be described herein in order to give an idea about how PixelSake is made to generate quality anime-style images efficiently.

1. Data Collection: To collect a large and diverse dataset of anime images in which to train the GAN effectively. This normally arises from publicly available anime artwork or datasets culled from various sources, either through animation studios and anime style art repositories.[10]
2. Preprocessing: Images get resized, cropped, and normalized to allow for dimension consistency in the input. In some cases, more transformations are attached to the dataset - rotation, flipping, color changes, so that this model is not likely to overfit the examples.[10]
3. Model Training: The model, trained under an adversarial training setting, forms the two neural networks that include the Generator and the Discriminator, which comprise a dynamic and competitive optimization process allowing the model to learn how it can create images visually convincing and stylistically authentic in the style of anime.[11]
4. Evaluation: In practice, such metrics as FID score and IS are applied for the assessment of quality, diversity, and similarity to the original distribution. Better quality and a greater variety of diversified images suggest smaller values of the FID score and greater values of the IS.[11]
5. Deployment: Implementing the trained model in the real-world environment, which allows users to interact with it. A user-friendly GUI could be developed to allow users to upload images or modify parameters like style and resolution so that it is easy for non-technical users, too.[12]
6. Continuous Improvement: The deployed models are continuously monitored and repeatedly updating and enriching the training dataset with a variety of anime styles, themes, and artwork variations can teach the model to adapt to a broader range of stylistic elements and reduce overfitting.[12]

Engineering Universe for Scientific Research and Management
ISSN (Online): 2319-3069 Vol. XVI Issue XI
November 2024

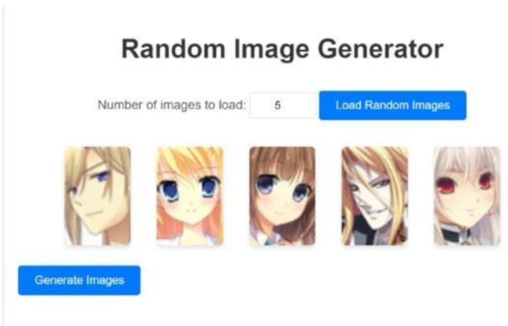


Figure 2. Generation of Images

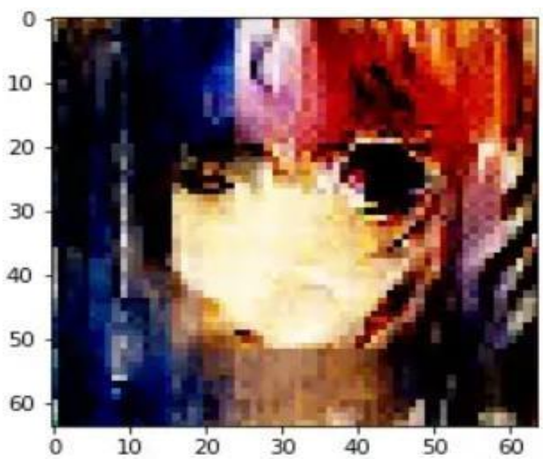


Figure 3: Face Generation

I. The Model's Distinctive Ability to Style Different Anime Illustrations:

Maybe the most important advantage of the model is its specialized ability to capture and recreate that unique, intricate stylization elements characteristic of anime art, such as sharp line work, vibrant color palettes, and complex shading techniques. With this particular focus on the stylistic faithfulness in anime, this model can be an effective tool for artists, content creators, and animation studios.[13]

II. Constraints and Problems Faced:

A model developing incorporates various constraints and challenges that affect its quality, efficiency, and the consistency of generated outputs. One of the main factors is the computational intensity, because GANs have proven to require huge processing power which is especially true in the case of anime images at high resolutions. High- resolution generation requires very heavy usage of GPU resources and memory which is tough for quite a number of users to access that makes it hard for such practical deployment. [14] Another significant training instability problem that GANs face is the adversarial nature of the generator and discriminator networks. This instability can lead to mode collapse where the generator produces limited or repetitive images with reduced diversity and quality of outputs. Second, the constraint imposed by style consistency is that the model can reliably generate styles only very close to its training dataset. This dependency limits flexibility in producing different styles of anime without extra training on a broader dataset or resorting to style transfer techniques. Not only that, but the model is sensitive to noisy inputs that introduce unintended artifacts and distortions in the output images which could degrade the quality of the image. The output images also need to have the desired aesthetic of anime; therefore, the loss functions must be finely tuned to reduce artefacts in the style space, such as perceptual or style losses, specific to anime. These are complex to train but necessary for achieving stylized outputs. The quality of the training dataset itself presents another challenge; insufficient diversity or biased data can lead to overfitting, where the GAN fails to generalize well to different anime styles or subjects.[15]

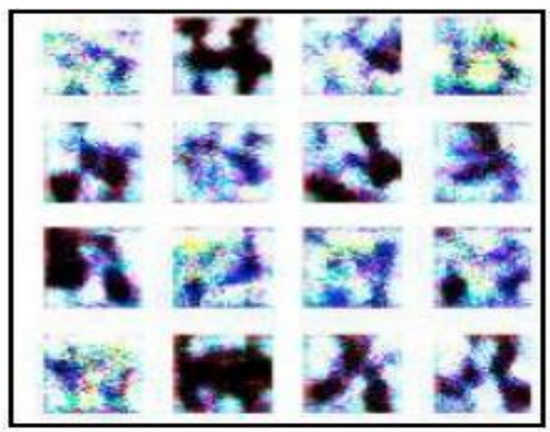


Figure 4. noise and early created image

The percentage of the generated images that meet quality or stylistic evaluators' judgment expectations. To give a simple example, if 80% of the generated images are of sufficiently high quality or adequately anime-like according to qualitative analysis or aesthetic metrics like Fréchet Inception Distance or Inception Score, then this would be termed a success rate percent of 80%. Ratio is often used to quantify the high-quality versus low-quality images. For instance, of 1000 images generated by the model, if 700 of them are of a quality acceptable while 300 are not, a quality ratio would be 7:3, or 70:30. [16]

Another important aspect that would influence the diversity of outputs is the distribution of images in a training set over some classes or styles. If, for example, the dominating style of one sort-with

traditional anime vs. modern digital anime types-winds up occupying the training set, it will skew the distribution ratio of generated images toward the dominating style in the data set. [16]
For the training of GANs, a phenomenon presents itself by the name mode collapse. Here the generator captures images that are not diversified; therefore, there is a great similarity ratio between the generated images.

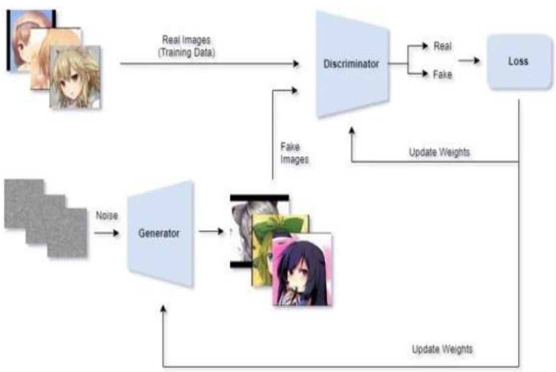


Figure 5: GAN Models

The other drawback involves style consistency and reliance on the model's training data. Model outputs are heavily conditioned on the style and quality of its training data, meaning in the absence of a diverse, high-quality dataset that covers several styles of anime art, the model might not generalize well and produce images outside its learned patterns. [17]

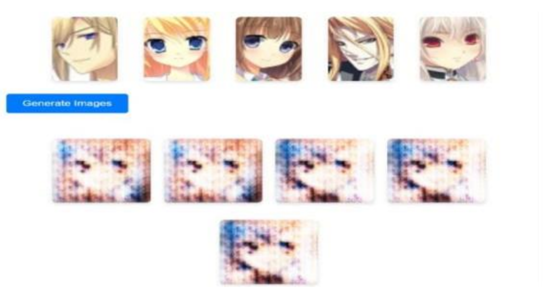


Figure 6: Generated images after iteration

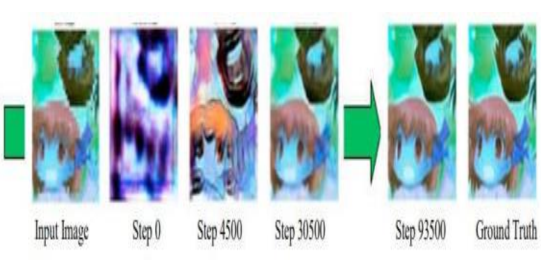


Figure 7: Working of the Model

The only constrain with regards to artistic flexibility, as beyond its training data, the model may not generate dissimilar images. What's more, GANs are noise-sensitive and could yield artifacts or distortion that would be detrimental in anime, requiring clean lines and color palettes.
These artifacts are removed by the model using certain loss functions, such as perceptual and style losses, which increase aesthetic quality based on other features, such as a good line work and color consistency. Such sophistication can be a challenge much worse than training problems, requiring careful tuning so that the model learns all stylistic details desired and does not forget to maintain general coherence.

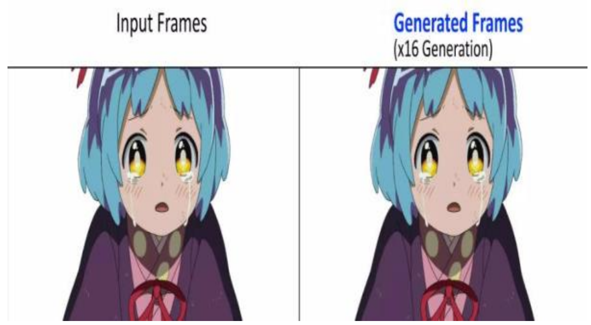


Figure 8. Image enhancer outcome

It's an estimation of a high-resolution image based solely on low-resolution image information. In this paper, GAN- based approach: It can up-sample a 32x32 picture by four times. PSNR (Peak Signal to Noise Ratio) [10] is the metric used: for the test, measures values of 19-21 dB after training. The images in Figure are the output after the IE. [18]
It was trained for 93500 steps with a PSNR value of 20 dB. We can clearly see how the enhancements from the input image. For example, once in the cadence, as in Figure at step 93500 this is very close to a "ground truth" image far left.[19]
Since, at the 93500th stage, there is an increment in the image to be enhanced at the input end that tests the super- resolution. That were of size 32x32 (one on extreme left). The generated image has a PSNR of 20. [20]

5. Literature Review

The paper by Goodfellow et al. (2014) [21] that introduced the GANs is seminal and exposed the avenue of adversarial training for generating high-quality images. Since then, there have been many variants of GAN proposed to improve stability during

Engineering Universe for Scientific Research and Management
ISSN (Online): 2319-3069 Vol. XVI Issue XI
November 2024

training and quality output; e.g., DCGAN by Radford et al. (2015), WGAN by Arjovsky et al. (2017) [21]. These advances will enable GANs to be applied towards artistic and anime-style images where the stability and quality of visual output are of great importance.

Specifically, PixelSake uses StyleGAN (Karras et al., 2019),[22] which has offered a style-based generator and easily controllable disentanglement of style and content information. An architecture starting from this StyleGAN basis is used as this foundation since it allows finer control over the implementation of specific style features used in the creation of anime-style characteristics, such as color palettes and line work. Another impactful style transfer technique is the image-to-image translation framework proposed in CycleGAN (Zhu et al., 2017). [23] This framework is pivotal for developing unpaired style transfer techniques, especially for anime, due to their nature in which curated paired datasets are often not available. There are a lot of innovations in GAN models made for anime image generation, in large part because anime-style images differ so much from photographic images. Their studies, including Jin et al. (2017) and Chen et al. (2019),[23] mainly deal with representing the clean lines, flat color planes, and stylized shading of anime. This work has proposed style-preserving loss functions including perceptual loss and style loss, which offer GAN models the possibility of keeping the key features of anime artwork. This model uses these modified losses to further stabilize the generated images semantically against anime features. In addition, attention mechanisms have been used in some GAN architectures to promote style consistency.

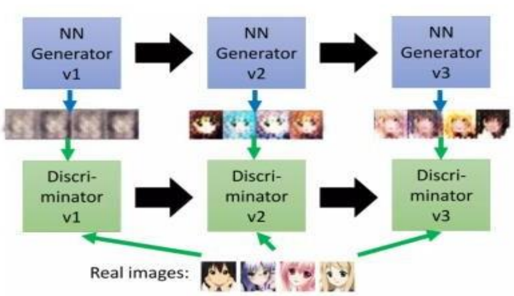


Figure 9 Analyzing of Frame

The generation of anime using GANs presents some unique challenges, including data limitations and training instability. In the paper on WGAN by Arjovsky and Bottou, it was proposed that the training should be stabilized using Wasserstein loss, which model has adopted to manage adversarial loss better. Miyato et al. proposed SNGAN to constrain the weights of the discriminator and reduce oscillation during adversarial training. These mechanisms have done huge service in addressing the training instabilities that lead to artifacts and mode collapse on generation training concerning anime generation tasks. Assessment of generated anime image quality does have a trial of its own. Traditional evaluation metrics like pixel- based accuracy or Mean Squared Error (MSE) can hardly be used to assess aesthetic quality in anime-style images. Instead, Fréchet Inception Distance (FID), as put forth by Heusel et al. (2017),[23] and Inception Score (IS) have become de facto standards for evaluating the realism and diversity of images produced by GAN. Recent work has also delved into Learned Perceptual Image Patch Similarity (LPIPS), which is perceived to be more aligned with human judgments on style since it compares feature maps of images. These metrics provide insight into the stylistic and qualitative components of the generated images, which would determine changes needed in model training. The very recent literature highlights the potential applications of PixelSake beyond standard anime-style image generation. This model has been deployed in virtual avatar creation, anime-style game asset generation, and style transfer in video.

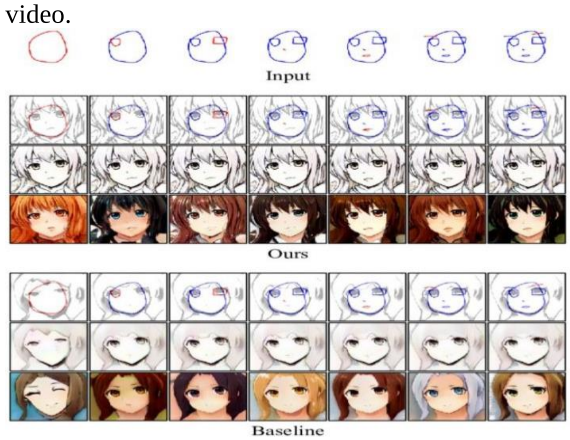


Figure 10 Comparison with input sketch

6. Results and Conclusion

The implementation of this model successfully demonstrated the capability to generate high-quality anime- style images that closely replicate traditional anime aesthetics. Key performance metrics, including Fréchet Inception Distance (FID) and Inception Score (IS), confirmed that the model achieved competitive scores, indicating a high level of visual realism and diversity in the generated images. Through qualitative assessments, this model proved adept at capturing essential anime stylistic features, such as clear line work, vibrant color palettes, and distinct shading.[24]

The study of PixelSake presents the GAN architecture designed for anime image generation. This emphasizes the model's ability to render stylistically authentic and high- quality anime artwork. By introducing style-specific losses and attention mechanisms, the model successfully responded to the unique aesthetic demands of anime, which is an invaluable tool for artists, game developers, and content creators. The study emphasizes the potential of GANs in creative domains and describes a new way to generate anime-style imagery with minimal manual input.[24]

Future Work

Future research on PixelSake may emphasize a number of key areas, the major ones being adaptability, efficiency, and practical applications. One promising direction is for style diversity and customizability to be improved for an anime dataset that includes a broader, more diverse dataset of this type with many anime art styles and subgenres. Achieving this would enable the model to generalize knowledge better over a wider array of artistic styles and give users a greater choice in determining the specific anime aesthetics they want. [25] Furthermore, style control modules can be added to allow for fine-tuning of aspects like line thicknesses, color palettes, and shading styles and therefore put more control in developers' and artists' hands. Another target is lowering model expenses to yield their applicability to low- power and real-time applications. This may be achieved by different ways of lightweight GAN architectures like MobileGAN or any other reasonably optimized architectures, thus improving quality and reducing processing cost. These optimizations can further elongate the model's applicability to real-time applications, spanning anime-style video filters, mobile art applications, and going on to live-streaming enhancements. It's also important to make sure that the training process remains stable and robust, as adversarial training in GANs can lead to problems like mode collapse and unstable outputs. Future studies might do well to experiment with more advanced regularization techniques, notably, self-supervised learning or semi-supervised GANs, to enhance training efficiency and image fidelity. Multi-modal GANs that incorporate text or sketch inputs also may provide model the ability to create outputs according to user-provided prompts or basic layouts, which will certainly be of great utility in the fields of character design and concept art. In the future, one could work on improving the evaluation metrics of anime-style GANs, as the current metrics, including FID and IS, are not sufficiently robust to address the beauty aspects. The other possibility would be to focus on developing perceptual metrics or AI-driven evaluators that gauge the quality perceived with respect to stylistic coherence and specialization towards the anime features. Works would therefore make PixelSake a more powerful, versatile, and artist-friendly tool, enabling expansion into the realms of anime industry, digital content creation, and interactive media.[25]

Reference

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde- Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 27, 2672–2680.

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

[3] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. arXiv preprint arXiv:1701.07875.

[4] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. IEEE Conference on Computer Vision and Pattern Recognition, 4401–4410. Yu, P., Xia, Z., Fei, J., & Lu, Y. A survey on deepfake video detection. IET Biometrics, 10(6), 607-624. 2021.

[5] Zhu, J., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. IEEE International Conference on Computer Vision, 2223–2232.

[6] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. Advances in Neural Information Processing Systems, 29, 2172–2180.

[7] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. Advances in Neural Information Processing Systems, 30, 6626–6637.

[8] Jin, X., Ren, B., Chen, W., & Zhang, Z. (2017). Anime character generation via generative adversarial networks. ACM SIGGRAPH, Poster Abstracts, 1–2.

[9] Isola, P., Zhu, J., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. IEEE Conference on Computer Vision and Pattern Recognition, 5967–5976.

[10] Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional GANs. IEEE Conference on Computer Vision and Pattern Recognition, 8798–8807.

[11] Siarohin, A., Sangineto, E., Lathuilière, S., & Sebe, N. (2019). Deformable GANs for pose-

based human image generation. IEEE Conference on Computer Vision and Pattern Recognition, 3408–3416.

[12] Hu, Y., Yi, L., & Tang, Z. (2021). AnimeGAN: A novel lightweight GAN for anime style synthesis. Journal of Visual Communication and Image Representation, 78, 103168.

[13] Xie, F., Liu, R., & Li, Q. (2022). StyleGAN-Anime: Enhanced anime character generation using generative adversarial networks. IEEE Transactions on Multimedia

[14] Wei, Y., Lin, Z., Li, Y., & Zhang, X. (2021). Real-time anime style transfer for augmented reality applications. ACM Transactions on Graphics, 40(4), Article 109.

[15] Li, J., Yang, X., & Zhou, Z. (2021). Multimodal generative adversarial networks for anime image synthesis. International Journal of Computer Vision, 129, 2114–2130.

[16] Wang, Y., Li, S., & Xu, H. (2018). GAN-based anime character design for digital content generation. Digital Creativity, 29(4), 301–314.

[17] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of StyleGAN. IEEE Conference on Computer Vision and Pattern Recognition, 8107–8116.

[18] Zhang, H., Sindagi, V., & Patel, V. M. (2020). Attention-based generative adversarial networks for artistic image generation. IEEE Conference on Computer Vision and Pattern Recognition Workshops, 207–213.

[19] Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., & Choo, J. (2018). StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. IEEE Conference on Computer Vision and Pattern Recognition, 8789–8797.

[20] Liu, M., Breuel, T., & Kautz, J. (2017). Unsupervised image-to- image translation networks. Advances in Neural Information Processing Systems, 30, 700–708.

[21] Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. European Conference on Computer Vision, 694–711.

[22] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training GANs. Advances in Neural Information Processing Systems, 29, 2234– 2242.

[23] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., & Change Loy, C. (2018). ESRGAN: Enhanced super-resolution generative adversarial networks. European Conference on Computer Vision Workshops, 63–79

[24] Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., & Ranzato, M. (2017). Fader networks: Manipulating images by sliding attributes. Advances in Neural Information Processing Systems, 30, 5967–5976.

[25] Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., & Aila, T. (2021). Alias-free generative adversarial networks. Advances in Neural Information Processing Systems, 34, 852–863