DragGAN-Based Emotion Image Generation and Analysis for Animated Faces

Daqi Hu
The School of Computer and Artificial Intelligence, Nanjing University of Science and Technology Zijin College, Nanjing, China

Keywords: Generative Artificial Intelligence, DragGAN, StyleGAN, Motion Supervision.

Abstract: In recent years, generative artificial intelligence (AI) and its applications have become a hot topic among art designers and content creators. There is a need for a simpler and more direct method to slightly edit images. In this paper, author introduces Drag Your Generative Adversarial Network (DragGAN) and improve its discriminators and features to adapt to anime styles. This work consists of two main parts: algorithm design based on Style-Based GAN (StyleGAN) model and application to anime style images. Specifically, an analytical model is first constructed using DragGAN. The process is called motion supervision. The input image should match the trained model. Secondly, it uses point tracking to continuously iterate the generation process and gives the result of each iteration. Third, the analysis compares the predictive performance of different models and provides an interactive GUI application as a demo project. With this research, anyone can edit their anime style portraits with a few clicks and drags. This can help people to reduce the time spent on editing anime style images and increase their productivity and creativity.

1 INTRODUCTION

Anime style refers to a unique and vibrant art form that originated in Japan. Today it has become a global cultural phenomenon. Anime is characterized by colorful visuals, fantastical themes and rich expressions. It has captured the hearts of millions of anime fans worldwide. This unique style of animation has found its way into a variety of media, including television series, movies, manga (Japanese comics), video games and more.

Deep learning-based image-to-image translation has produced excellent results in the past few years (Schmidhuber 2015 & Chen et al 2020). Generative Adversarial Network (GAN) have been proved to have the strong ability in style conversion and have been the greatest solution for slightly image correction (Chen et al 2020, Goodfellow et al 2014 & Pan et al 2023). Instead of training raw pixel data, GAN pay more attention to the changes. In the context of style conversion, the origin and the result are “pairs”, GAN trains a discriminator to recognize the pair to convert origin image to the result. So, image generation with GAN method is faster and more precise than traditional reinforcement learning. Users can edit the content of any GAN-generated images with a single drag in Drag Your GAN (DragGAN) (Pan et al 2023). It provides a kind of way for users to get the demanded image in a user-friendly web GUI. DragGAN separated the approach into two parts to achieve the goal: a feature-driven motion supervision that directs the handle point toward the target position and an advanced point tracking method that continuously localizes the handle point's position using the discriminative generating features. It mainly focuses on image generation of in-real-life photos. However, in Anime-style image generation, the feature of image is quite different from that. Firstly, Anime-style images have sharp stroke edges. Those sketching represents ambient occlusion and subsurface scattering in real lighting environments. Second, the textures and diffuse of cloth and skin are simplified to single diffuse color. Since faces have to be more beautiful than real-world people, the discriminators and models of DragGAN cannot be used directly.

Most researchers have accomplished this by generating a 2D image using a 3D model or by generating the entire picture using a diffusion model (Deng et al 2020, Jascha et al 2015, Song et al 2020, Song 2011 & Teed and Deng 2020). Both of these methods do not fulfill the requirements of accuracy and flexibility. To solve this problem, this paper introduces DragGAN and improves its discriminators

Hu, D.
DragGAN-Based Emotion Image Generation and Analysis for Animated Faces.
DOI: 10.5220/0012799500003885

In *Proceedings of the 1st International Conference on Data Analysis and Machine Learning (DAML 2023)*, pages 234-239
ISBN: 978-989-758-705-4

and features to fit Anime-style. Specifically, first, DragGAN is used to construct the analysis model. The process is called Motion Supervision. The input image should match the trained model. Second, it uses Point Tracking to keep iterating the generating process and give the result of each iteration. Then users can stop at any step if the image fit their needs before the final iteration. Third, the predictive performance of the different models is analyzed and compared. As mentioned above, the official DragGAN requires and fits well on real image and its associated pretrained model. What this project does is expand the algorithm to stylized images, and use Anime style as an example. This research can help users get an easier way to alter image, especially in rotation and relocation requirements, which gets better results than traditional image editing algorithms.

2 METHODOLOGY

2.1 Dataset Description and Preprocessing

The dataset in this project is "Ganyu | Genshin Impact Anime Faces GAN Training" (Dataset 2023), which is downloaded from Kaggle. The pictures in this dataset represents a specific anime character, and each image is 512x512 pixels in size. All of the images should be preprocessed to align its portrait to the center and resize it to 512x512. There are different angles of the character face and they allow users to be exposed to different environments and lighting. To gain better results, additional styles of character are also recommended.

2.2 Proposed Approach

This project focuses on providing a method of anime image transition, which is based on Style-Based GAN (StyleGAN) network and some application work from DragGAN. StyleGAN3 has been a state-of-the-art method in GAN field since it releases its first version StyleGAN in 2019, and it released a better version recently which fixes some issues and add support to PyTorch and Ada Architecture graphics cards. DragGAN supports StyleGAN2 and StyleGAN3 at the same time. Either repository is available to train model or use the pretrained models from them. One of the most well-known StyleGAN3 model is a real-human image generation model. The whole process of this implementation comes from datasets, which usually is the pretrained models. In DragGAN, the Generator of exponential moving average (Gema) in StyleGAN model will be used to manipulate the source and target points from user interface. Then the model will predict the movement of next frame's generated image. The data is from Gema, which will be discussed later in this passage. Finally, it tracks the manipulated point and update every frame. The process is shown in the Fig.1.

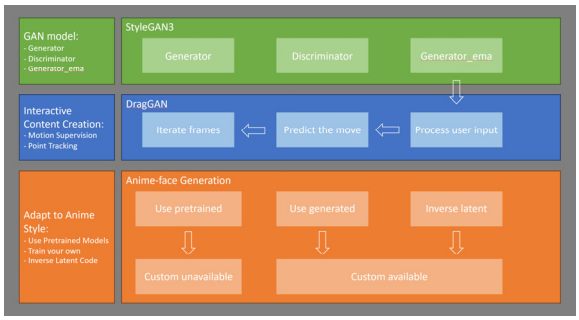

Figure 1: This project's architecture (Picture credit: Original).

2.2.1 StyleGAN3

The StyleGAN is a new generator and discriminator architecture for GANs. GAN is a generative model for deep learning that is characterized by generating realistic samples by means of adversarial training.GAN consists of two parts: a generator and a discriminator. The generator is responsible for generating false samples, while the discriminator is responsible for classifying the real and generated samples. The Generator is responsible for converting random noise inputs into spurious samples. The discriminator is responsible for determining whether the input sample is a true sample or a false sample generated by the generator. The generator receives a random noise vector as input and generates a spurious sample. The discriminator receives the true sample and the false sample generated by the generator and tries to distinguish them accurately. The goal of the generator is to generate false samples that can deceive the discriminator, while the goal of the discriminator is to accurately classify true and false samples. By repeatedly and iteratively training the generator and discriminator, the performance of the GAN gradually improves, and the generated false samples become more and more realistic.StyleGAN3 is an improved version of the GAN, which breaks through in terms of the quality and diversity of the generated images. Compared with the traditional GAN, StyleGAN3 has 3 features higher quality of generated images. StyleGAN3 generates more realistic, clear and detailed images by introducing new architectures and

training techniques. Better control of generation.StyleGAN3 allows fine control over different attributes of the generated samples, such as facial expressions, hairstyles, etc., providing more personalization options. Super-resolution generation: StyleGAN3 can generate high-resolution images, including image generation for detail enhancement and super-resolution reconstruction tasks. styleGAN3's thought process is similar to that of traditional GANs, but it introduces a number of improvements and innovations in the model architecture and training process. These improvements include architectural optimization of generators and discriminators, feature alignment mechanisms, regularization methods, etc., aiming to improve the quality and diversity of generated results. It produces an automatically taught, unsupervised separation of high-level features and random variation in the resulting images and enables straightforward, scale-specific synthesis management. StyleGAN3 optimized from Progressive GAN's method, it leverages modern GPU architecture and is implemented on modern machine learning framework. Every StyleGAN model provides a Generator, a Discriminator and a Generator which focuses on exponential moving average, denoted as Gema. So, these three parts in pickle file is called G, D, and Gema. The point that why DragGAN is implemented on StyleGAN3 is because the third version of StyleGAN have obvious improvements on video content generation, and they share the both backend and training utilities. In origin StyleGAN's implementation, it use a style-based generator to force the generated image stick to the given sample. This is the reason why StyleGAN3 is the state-of-the-art and is different from those previous GAN. Traditionally, the generator receives the latent code through an input layer. However, StyleGAN includes it with the first constant image. That is:

$$AdaIN(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i} \quad (1)$$

where each feature map x_i is normalized separately, and $y = (y_s, y_b)$ takes specialized w and act as a style to control adaptive instance normalization (AdaIN).

2.2.2 Latent Code

In the context of GANs, the Generator plays a crucial role in producing the generated images, while the Discriminator is responsible for training the Generator by providing feedback and ensuring that the generated results resemble the desired outcomes. However, GANs have expanded beyond simple image generation and can now be used to alter existing images from a given source to a target image. The connection between the Generator and Discriminator is achieved through latent code, which serves as an intermediary during the training process and remains invisible until all updates have been completed. As a result, the generated or altered images are derived from the datasets used for training, meaning that the outcomes are limited to what was defined within those datasets. Consequently, if users aim to alter an image that is not part of the training dataset, the original DragGAN method is unable to provide a viable solution. However, the Pivotal Tuning for Latent-based editing (PIL) method offers a solution to this limitation (Roich 2021). PIL enables the alteration of a pretrained model without the need for retraining. All that is required is the input image that the user wishes to edit. Figure 1 showcases the effectiveness of the Inverse Latent approach, which allows for customizable image editing without the need to retrain the model. This method seamlessly integrates into the entire process, providing a user-friendly and efficient solution for image manipulation. By leveraging PIL, users now have the ability to modify images that were not part of the original training dataset, expanding the scope of possible image alterations and offering greater flexibility in creative expression. This advancement introduces a significant shift in latent-based editing techniques, granting users the power to customize and transform images with ease and innovation.

How PIL affects pretrained models is in three steps, inversion, tuning and regularization. It is used to approach the origin StyleGAN model and adapt it to the given image. Inversion thus serves to improve the tuning phase's starting point. The most editable latent space is StyleGAN's native latent space W. Consider that the implementation should reconstruct s, which is the input image, with optimizing the latent code W, the pivot code p and the noise vector v. The following objective defines the optimization:

$$W_p, v = \arg\min L_{LPIPS}\big(s, G(w, v; \theta)\big) + \lambda_v L_v(v) \quad (2)$$

in which $G(w, v; \theta)$ defines that generator G and weights θ create the generated image. Instead of traditional StyleGAN-based methods using its mapping network, The PIL uses three individual networks. L_{LPIPS}, the perceptual loss, L_v, the noise regularization term and λ_v the hyperparameter.

Tuning and regularization phase are mostly same as origin StyleGAN.

2.2.3 DragGAN

DragGAN is an image editing method based on GAN, which consists of two main components: a Generator and a Discriminator. The Generator is responsible for generating realistic images, while the Discriminator guides the training of the Generator by evaluating the generated images to produce more realistic images.DragGAN extends the functionality of the GAN by allowing the user to perform image editing between a given source image and a target image. Specifically, DragGAN can generate editing results similar to the target image by learning the differences between the source and target images based on the source image provided by the user. This allows the user to edit the image in terms of morphology, color, and texture by using DragGAN for applications such as image style conversion, image enhancement, and image restructuring.

In DragGAN, the generator generates an image by sampling from the latent space (latent space). Latent space is a low-dimensional vector space in which each vector corresponds to a unique image style. By adjusting the values of the latent vectors, the user can control the different features and styles of the generated image.

The advantage of DragGAN over traditional rule-based editing methods is that it learns the distribution of the input image and edits according to the distribution of the target image. As a result, the editing results are more natural and realistic, and can be adapted to a wider range of image styles and contents.

The application areas of DragGAN include computer vision, image processing, and artistic creation. It provides users with an intuitive, flexible and efficient image editing tool that empowers them to create unique and innovative visual effects.

An image $I \in R^{3\times H\times W}$ generated by latent code L, are manipulated by input data points. The source points are defined as $\{s_i = (x_{s,i}, y_{s,i})|i = 1, 2, \dots n\}$, and the target ones are $\{t_i = (x_{t,i}, y_{t,i})|i = 1, 2, \dots n\}$. The goal is to move the points from source locations to target locations. In one optimization step, the model gets a iterated code L' and a result image I'. Theoretically this iteration is in every frame the GUI rendered, but it heavily depends on computer performance. In each iteration, the output of 6th block of StyleGAN2 which results in the feature maps F is forced to be aligned with the original image. So this is called motion supervision, and the loss function is defined as follows:

$$L = \sum_{i=0}^{n} \sum_{u_i \in Q_1(v_i, r_1)} \left|\left|F(u_i) - F(u_i + N_i)\right|\right|_1 + \lambda \left|\left|(F - F_0)\right|\right|_1 \quad (3)$$

where $F(u)$ means the feature values, which is in F at pixel u. And $N_i = \frac{t_i - s_i}{||t_i - s_i||_2}$ is a normalized vector.

Due to the actual processing phase is a supervision learning problem, the program can be easily running on modern desktop computers, seems like a nearly real-time image editing resolution. Finally, the reason why a point tracking phase is required in a loop is that motion supervision cannot ensure the source and target point are just correct with the position user defined in the beginning. Thus another optimization function is required:

$$p_i = argmin\left|\left|F'(u_i) - f_i\right|\right|_1 \quad (4)$$

which p_i referred to tracking point in each iteration. The tracked point is calculated from the nearest neighbor of f_i.

2.3 Implementation Details

This project is based on PyTorch 2.0. This project needs to install requirements of StyleGAN3 first, this is the base environment of all of the projects, including StyleGAN3, DragGAN and PTI. After ensuring all requirements are satisfied, please review the system environment PATH and CUDA toolkit version. The CUDA toolkit must be 11.8. Although it is not optimized for the latest Ada GPUs (e.g. RTX 4090, H100), but the source code of the SDK changed in 12.x and the PyTorch plugins in this project cannot be compiled successfully. Then, check whether current PyTorch version supports GPU, which should be torch==2.0+cu118. The training process of StyleGAN is known to be long and annoying, so high-end GPUs are recommended. The pillow package should be downgraded to version 9.5.0. If the source model (.pkl file) comes from other pretrained datasets, please follow the naming format as "<model_type>-<custom_name>-<512x512>.pkl". <model_type> can be "stylegan2", "stylegan-human" or "stylegan3". Other names cannot be recognized and will receive an exception during the runtime. Although those projects support both conda environment and pypi environment, native python usually do not include C debug symbols (this can be checked during the first installation, just updating it cannot recover). And it may pop up "cannot open file python311.lib" issue

when attempting to build the PyTorch plugin. So conda environment is recommended.

3 RESULTS AND DISCUSSION

As depicted in Figure 2, it observes the utilization of a red point as the source point and a blue point as the target point. The Drag process initiates with the objective of bringing the red point and its neighboring points closer to the blue point. This process relies on the Generator component within the GAN, specifically known as Gema. It can be likened to watching a generated video that can be halted once the red point reaches the blue point, or manually interrupted at an intermediate stage.

In comparison to the original DragGAN method, this project ensures that stylized images can be both edited and generated using the StyleGAN3 model. Additionally, they can be manipulated using the DragGAN framework and its accompanying program. This advancement makes significant contributions to the editing of anime characters, negating the requirement for precise 3D models or professional sketching skills. It facilitates a quick preview of how different emotions or physical features manifest on the same character.

Compared to diffusion models, GAN-based models offer more precise control over details, which is a notable advantage. This enables the model to finely manipulate and generate stylized images, further enhancing the capabilities of image editing techniques. The marriage of the GAN-based approach with the DragGAN method opens up new possibilities in the realm of anime character customization and facilitates the exploration of diverse artistic expressions.

Figure 2: Result of drag (Picture credit: Original).

4 CONCLUSION

This project mainly focuses on reattain the program and adapt it to stylized images or unreal contents. This research uses anime style as a reference and example, to show how easy and direct it will be to alter the image through a simple drag, compared to traditional image editing methods which requires a fulfil background of user aesthetic experience and art study background. Meanwhile, this will not take sketchers' place, because GAN models actually cannot create an image. In fact, although it can generate images, but this generate process is mostly named after its programming appearance, which is compared to traditional machine learning methods. So compared to diffusion models, which can really create image from scratch, this project can protect the design and right of original creator and art designer, while providing necessary and functional altering methods, which in fact take the previous CPU-algorithm-based editing methods. In the future, author will connect StyleGAN3, DragGAN and PTI together, and make a combination that user just need to upload their image they want to alter, and get the result in the same program.

REFERENCES

J. Schmidhuber, "Deep learning in neural networks: an overview," Neural Netw, vol 61, 2015, pp. 85–117

J. Chen, G. Liu, X. Chen, "AnimeGAN: a novel lightweight GAN for photo animation," International symposium on intelligence computation and applications, vol.1205, 2020. pp. 242-256

I. J. Goodfellow, et al, "Generative adversarial nets," In: Proceedings 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014, Montreal, QC, Canada, pp. 2672–2680

D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," arXiv 2016, unpublished.

X. Pan, A. Tewari, T. Leimkühler, "Drag your gan: Interactive point-based manipulation on the generative image manifold," ACM SIGGRAPH 2023 Conference Proceedings, vol. 2023, pp. 1-11

Y. Deng, J. Yang, D. Chen, "Disentangled and controllable face image generation via 3d imitative-contrastive learning," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 5154-5163

S. Jascha, W. Eric, M. Niru, G. Surya, "Deep unsupervised learning using nonequilibrium thermodynamics," In International Conference on Machine Learning. PMLR, vol. 2015, pp. 2256–2265

J. Song, C. Meng, S. Ermon, “Denoising diffusion implicit models,” arXiv, 2020, unpublished.
Y. Song, J. Sohl-Dickstein, D. Kingma, “Score-based generative modeling through stochastic differential equations,” arXiv, 2011, unpublished.
Z. Teed, J. Deng, “Raft: Recurrent all-pairs field transforms for optical flow,” Computer Vision–ECCV 2020: 16th European Conference, Proceedings. Springer International Publishing, 2020, pp. 402-419
Dataset, https://www.kaggle.com/datasets/andy8744/ganyu-genshin-impact-anime-faces-gan-training, last accessed 2023/08/25
D. Roich, R. Mokady, A. H. Bermano, and D. Cohen-Or, “Pivotal tuning for latent-based editing of real images,” arXiv, 2021, unpublished.