Few shot generation of face images based on GAN inversion

Baisong Guo
stilltooyoung@sjtu.edu.cn

Yan Zhuang
zhuang00@sjtu.edu.cn

1 Introduction

Few-shot Image generation can generate a large number of additional training samples when the training samples are insufficient, which is very valuable for current deep learning methods that require a large amount of data for learning. Although it remains to be disputed whether these generated samples can improve the performance of the model, more and more evidence shows that the few-shot learning is a meaningful research work. Generative adversarial network(GAN) is now a very effective data generation method. The training process of GAN-based model is very unstable and also difficult to converge. Therefore, in the early exploration of GAN-based approach, the generated images are often either blurry or low-resolution. In recent years, with the continuous exploration of training methods of GAN, researchers have been able to generate more realistic images. Generally, GAN-based approach can only randomly generate a real image without any label based on noise. Recent work has shown that a method similar to gradient descent, which is called GAN inversion, can be used to control the generation of image. Because GAN-based approach itself has the property of interpolation, this paper takes advantage of this property and the latest GAN inversion technology to generate a large number of similar face photos from several photos of a person, so as to realize the few-shot generation of faces with the same ID.

2 Related work

2.1 GAN approach for face generation

GAN[1] is a very powerful generative model, which can generate images that are very closed to real images from random noise through adversarial learning. Early GAN-based approach can only generate very blurry pictures[2][3]. With the continuous advancement of technology, researchers mainly use two methods to improve the generation effect of GAN. One is the method of using a large number of computing resources and more parameters to improve the generation effect, represented by bigGAN[4], and the method of improving the generation

effect by improving the loss function of GAN, such as WGAN[5]. Unlike generative models in the ordinary sense, researchers have conducted specific studies on face generation. In order to improve the performance of GAN on the face generation, PGGAN[6] used a progressive generation method to increase face generation to a resolution of 1024 for the first time. However, there will still be artifacts in the generated photos of PGGAN. In order to solve this problem, styleGAN[7] draws on the form of progressive generation in PGGAN. In addition, styleGAN improves the introduction of random noise, using adaptive instance normalization to improve the generation performance. On this basis, styleGAN2[8] found that the powerful ability of AdaIN will guide the model to generate drop-shaped artifacts, so the author weakened the role of this module and added weight normalization to help the model generate better face images.

2.2 Few shot generation

At present, the few-shot learning mainly focuses on few-shot classification and few-shot image generation. This paper focuses on the latter task. Few-shot image generation is a very difficult task, and traditional image generation methods do not perform very well on this task. FIGR[9] was proposed to generate real images based on adversarial learning. DAWSON[10] uses the meta-learning method to generate new images and at the same time realizes domain adaptation between seen categories and unseen categories. MatchingGAN[11] tries to learn new metrics to generate new images from a small number of images. In the latest work, F2GAN[12] proposed a fusing-and-filling method to obtain new images by interpolating several input images at the feature map level.

2.3 GAN inversion

Traditional GAN-based approach can only generate an image randomly through noise, while in practical applications, we need to control the content of generation of GAN-based appraoch. ConditionalGAN[2] changes the input noise into a constraint condition, thus realizing the connection between the input and output of GAN. However, in the face of GANs with random noise, we still cannot control the connection between their input and output. In recent work, pulse[13] uses the gradient back propagation method to continuously optimize the input image by calculating the distance between the input and the target image. When this process converges, we can find the value in the hidden space for a given image. This paper tries to use this method and GAN interpolation to generate more similar images. Next, we will explain our method in detail.

3 Approach

Given a pretrained facial generate model G, such as styleGAN, and few photos $X = \{x_i, i = 1...n\}$ from a same person, here, the different photos are supposed to have different expressions. We first try to find out the corresponding latent

codes $Y = \{y_i, i = 1...n\}$ for X, satisfied $G(y_i) = x_i$, same as pulse, we use gradient descent to find the most closed latent codes $\hat{Y} = \{\hat{y_i}, i = 1..n\}$ satisfied that $G(\hat{y_i}) \approx x_i$, here the loss is

$$L_{recons} = MSE(G(\hat{y_i}), x_i)$$

however, sometimes x_i's size is different from the fixed resolution of the model, for example, the size of x_i is 256x256, but the pretrained GAN is trained on dataset whose resolution is 1024x1024, we can applied a differentiable degenerate methods such as bicubic to the output of the model, so the loss function becomes

$$L_{recons} = MSE(D(G(\hat{y_i}), x_i))$$

Here D is the differentiable degenerate methods such as bicubic. The pipeline to find out $\hat{Y}$ is shown in fig.1.

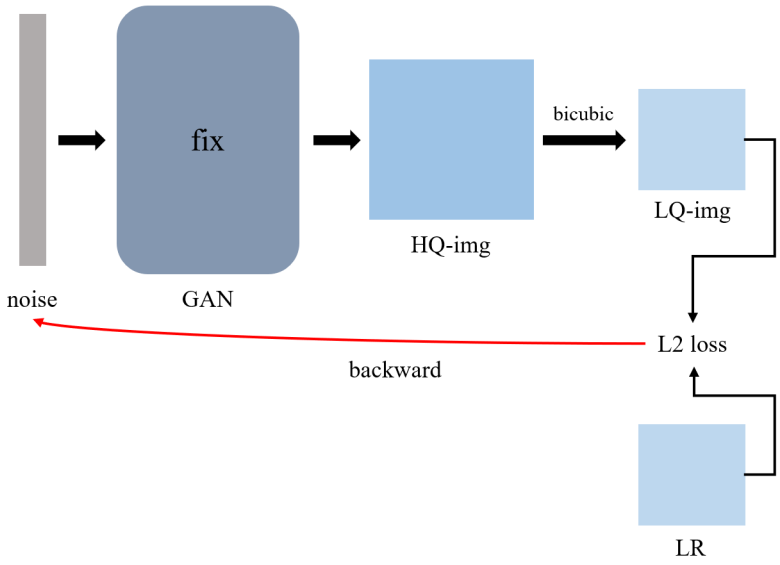

Figure 1: Pipeline to find out the corresponding latent variable for target image

Then given a set of weighted value $W = \{w_i, i = 1...n\}, \sum_{i=1}^{n} w_i = 1$, we can get a new latent codes and get a new image as $x_z = G(z)$, here

$$z = \sum_{i=1}^{n} w_i \hat{y_i}$$

4 Experimental Setting

Here we use a pretrained face generate model, more precisely, a styleGAN[7] trained on FFHQ[7]. FFHQ have about 60000 images from different persons, and the images all have a 1024x1024 resolution. As for test dataset, we use the celebA[14] dataset as our dataset. CelebA has 202,599 images from 10177

different persons, we randomly choose 5 images from each person.
To evaluate our methods, we use FID(Frechet Inception Distance score) to evaluate the generated images' quality. And we have to test the distance from our generated images to the origin images, actually this is hard for general image generation for there is no suitable metrics to evaluate the semantic distance between two images. Fortunately in face images, we can use FaceNet[15] to calculate the similarity between two face images.

4.1 Network Architecture

In this section we will introduce our network structure. We use the trained StyleGAN as our generate model, which was shown as fig.2. Here we show the generation of a 32x32 resolution image. Higher resolution images' generation model is similar to it. Different from traditional models, styleGAN have two inputs that are both sampled from normal distribution, one is the style noise s and the other is detail noise n. s is a 512 dimension vector and it will be copied into 18 copies and sent to every layer in the network. Actually it will be added to the weight of convolution kernel[8], this method has similar effect as adaptive normalization layer[7] on the feature map. So it can control the main component of the generated image. n is a random noise sampled from normal distribution with the same size (height and width) as feature map but only one channel. It will be copied at channel dimension and then added to the feature map. So it will help model to generate the details such as hair and pores.

4.2 Loss function

During the training process, n is cropped from a large pre-sampled noise map and s is copied as 18 copies, but in GAN inversion, to improve the fitting ability of the GAN model, we use different variable at each layer, that is, our variables to be optimized can be described as $S = \{s_i, i = 1...L\}, N = \{n_i, i = 1...L\}$, here L is the number of the layers in the model, in the L-st layer, the size of the feature map is $(2^L, 2^L)$. However, if we use n_i in all layers as our optimized variables, the generated image will be unnatural. So we only use first several layers' detail noises as our optimized variables. So the final optimized variables are $S = \{s_i, i = 1...L\}, N = \{n_i, i = 1...k\}$, in our experiments, we choose $k = 5$. Our target is to find out $\hat{S}$ and $\hat{N}$ that

$$\hat{S}, \hat{N} = \arg\min_{S,N} L_2(Down(Gen(S, N)), T)$$

Here T means the target image, and $Down$ means the differential downsample function. When the size of target image is same as the output of the generator Gen, $Down$ is identity transformation. Note that during the training process, the style noise S is copied from the same one, but in GAN inversion, we set the s_i as different optimization variables to get more powerful fitting ability, so we add an extra loss function to control the stability of the generated images, so the final optimization target is as follows, Here L_{arc} means the arc distance of s_i and s_j, for $i \neq j$:

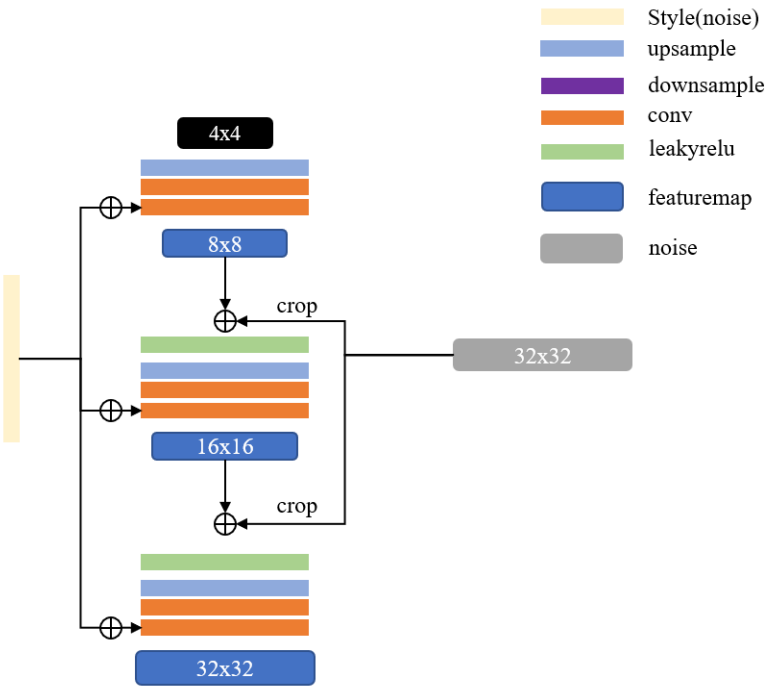

Figure 2: A detailed illustration of our generated model for 32x32 resolution, the models with higher resolution is similar to this structure

$$\hat{S}, \hat{N} = \arg\min_{S,N} L_2(Down(Gen(S,N)), T) + \alpha L_{arc}(S)$$

4.3 Interpolation

Our test images are from real world videos, and we only need two frames from the video to generate plentiful similar images with high quality. Actually, set all any two frames as f_0 and f_1, firstly we use GAN inversion to find out the corresponding latent code for them, called $\{S_0, N_0\} and \{S_1, N_1\}$, then we can get the interval latent codes by:

$$S^\alpha = \{s_i^\alpha = \alpha s_i^0 + (1-\alpha)s_i^1, i = 1...L\},$$
$$N^\alpha = \{n_i^\alpha = \alpha n_i^0 + (1-\alpha)n_i^1, i = 1...k\},$$

As for $\{n_i^x, i = k+1, ...L\}$, we set them as same variables sample from normal distribution for any x. As shown in fig.3, given two frames or two similar images from one person, we can first find out the latent code for them and then use interpolation to get several similar images.

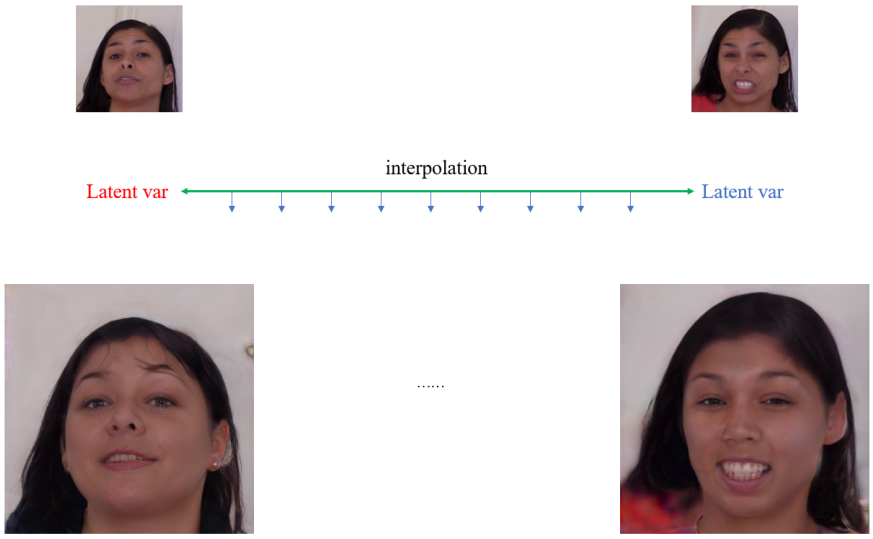

Figure 3: The interpolation process between two different images

4.4 Experiment Results

Figure 4: The intermediate results at different steps and the target

4.4.1 GAN inversion

Here we use Adam optimizer, and the learning rate is initial as 0.1, with cosine optimizer scheduler, the total step is 3000. Fig.4 is the visualization of some intermediate results at different steps and the target. All the data are from one real world video. The noise is initial from normal distribution so that the first image is very different from the target image but the difference will gradually decrease as the number of steps increases.

4.4.2 Few shot generation

Assume that we have two different frames or images of one person, we can use GAN inversion and GAN interpolation to generate lots of similar images as we have showed before. Some results are shown in fig.5. We have tested our result with FID and compared the results with two face generation model. The result is shown in table.1. Although our results are worse than the other two models, but we still got a pretty good FID. For our test images are from real world videos, the quality of our model's input is bad.

Models	StyleGAN	PGGAN	ours
FID	8.04	4.40	11.36

Table 1: The results of different models, note that the first two models are generation models, and their output images are based on the random noise.

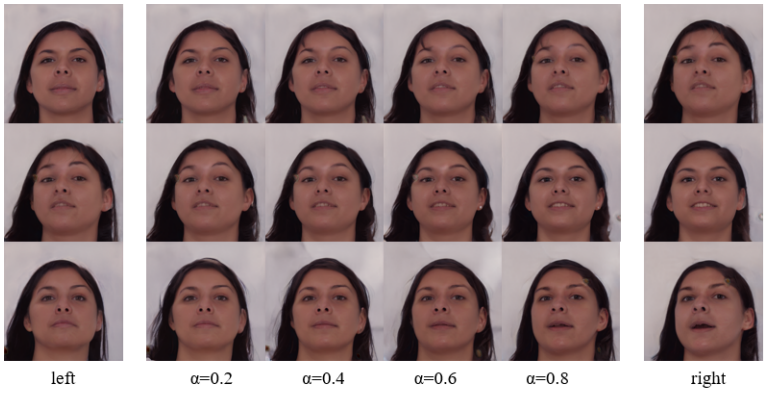

Figure 5: The interpolation results between two different images, the one on the far left and the one on the far right are the original images

5 Conclusion

We have proposed a new method to generate similar face images from few existing images. Based on a pretrained Generation model, we can divide the generation process to two steps: GAN inversion and GAN interpolation. We have achieved pretty good performance on FID metric, though it's worse than the original generation model, but the gap may be caused by the input of the model, which is from the real world videos with low quality. This method can produce plentiful high quality images from few shot images, so that it may be helpful for the down stream tasks in face vision field.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[3] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[5] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[6] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[8] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[9] Louis Clouâtre and Marc Demers. Figr: Few-shot image generation with reptile. *arXiv preprint arXiv:1901.02199*, 2019.

[10] Weixin Liang, Zixuan Liu, and Can Liu. Dawson: A domain adaptive few shot generation framework. *arXiv preprint arXiv:2001.00576*, 2020.

[11] Yan Hong, Li Niu, Jianfu Zhang, and Liqing Zhang. Matchinggan: Matching-based few-shot image generation. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.

[12] Yan Hong, Li Niu, Jianfu Zhang, Weijie Zhao, Chen Fu, and Liqing Zhang. F2gan: Fusing-and-filling gan for few-shot image generation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2535–2543, 2020.

[13] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2445, 2020.

[14] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.