Signal, Image and Video Processing
https://doi.org/10.1007/s11760-020-01660-0

ORIGINAL PAPER

Face attribute editing based on generative adversarial networks

Xiaoxia Song[1] · Mingwen Shao[1] · Wangmeng Zuo[2] · Cunhe Li[1]

Received: 25 June 2019 / Revised: 15 January 2020 / Accepted: 13 February 2020

Abstract
Face attribute editing is to edit the face image by modifying single or multiple attributes while maintaining the face identity. In the paper, we propose a method for attribute editing of face images by using the generative adversarial networks: conditional generative adversarial nets is used as the backbone of the framework and input attributes as conditions to the generator, the generator combines the encoder–decoder with U-Net, and the attribute classifier is added to guarantee the correct attribute operation on the generated image. The receptive field of a single discriminator is very limited, especially when the size of the training picture becomes larger, which will affect the extraction of information. In this paper, we tackle these limitations by using multi-scale discriminators to guide the generator to generate better details. It can macroscopically grasp the global information of the generated pictures and obtain more information of the receptive field. We demonstrate the effectiveness of our method and generate well-preserved facial detail images on CelebA dataset. The fidelity of the generated image is improved, and the method has better flexibility. The experiments show that our method is effective on the real-world dataset.

Keywords Generative adversarial networks · Multi-scale · Face attribute editing · Deep learning

1 Introduction

Face attribute editing is a task of changing a face image to given attributes (e.g., hair color, expression, beard and age) and generating new face images with desired attributes and retain details. At the same time, it ensures the invariance of face identity information and attribute-independent areas, as shown in Fig. 1. Relevant research has been widely used in entertainment, social interaction, facial animation, facial expression recognition and other fields, such as the whitening of human figures, aging and smiling. In addition, it is also applied in the field of face recognition on the expansion of the face database, which has attracted more and more attention in recent years.

With the introduction of generative adversarial networks (GANs) [1], great progress has been made in image generation [2–7] and image super-resolution [5]. Face attribute editing with GAN has also achieved excellent performance. GAN regards face attribute editing as an unpaired image-to-image translation task, one of which is the combination of GAN and AutoEncoder [6]. In these models [7–10], the AutoEncoder acts as a generator in the GAN. Using the encoder–decoder architecture, the encoder encodes the original picture as the latent representation, which facilitates different methods to perform different operations on the latent representation. The decoder decodes the latent representation to generate new pictures based on the expected attributes, so as to realize the editing of face attributes.

As a special GAN, conditional GAN (CGAN) [11] adds additional conditional constraints to the original GAN to guide the generation of specific images with given attributes. In addition, invertible conditional GANs (IcGAN) [8] further combine encoder and CGAN. IcGAN is a multistage training algorithm, training CGAN first and then the encoder. As an unsupervised generation model, VAE/GAN [9] combines the advantages of GAN and VAE. But VAE/GAN has highly relevant attribute vectors, which may change other irrelevant attributes when editing attributes. For example, most blondes in training set are female characters, so when editing blonde attributes, gender changes will also occur. Fader Networks [10] uses the encoder–decoder structure and discriminator that forces latent representations to be invariant to attributes. However, the use of attribute-independent constraint on latent representation can lead to loss of fine-

✉ Mingwen Shao
smw278@126.com

[1] College of Computer Science and Technology, China University of Petroleum, Qingdao, China

[2] Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

Published online: 27 February 2020

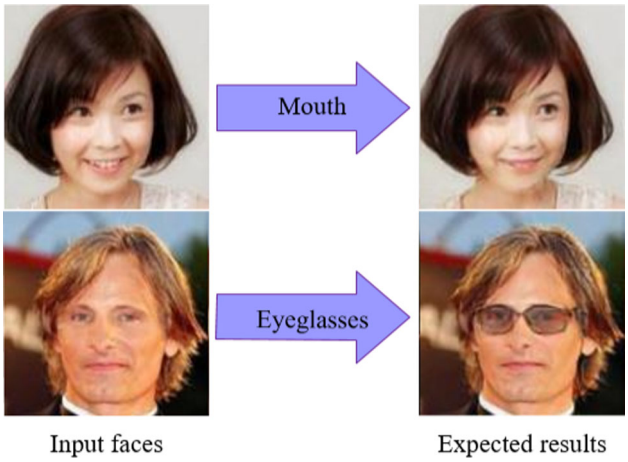

Fig. 1 Examples of face attribute editing

grained details and geometric artifacts. Therefore, AttGAN [7] proposes a new method to overcome the shortcomings of the above three methods in modeling the relationship between latent representations and attributes. It proposes an attribute classification constraint to ensure the correct editing of attributes. Furthermore, pix2pixHD [12] uses a novel multi-scale generator-discriminator structure to effectively help improve the quality of generated images, thus generating high-resolution and realistic images.

Inspired by the encoder–decoder method, combining the advantages of AttGAN [7] and pix2pixHD [12], we propose a method for editing face attributes by combining the generative adversarial networks and AutoEncoder. It can produce more pleasant visual results via fine facial details. In short, our work can be summarized in three aspects:

1. We propose a face attribute editing model which combines the GAN and AutoEncoder. The AutoEncoder acts as the generator, and the input of our model consists of images and attributes. We use WGAN-GP [13] to optimize the loss of GAN and realize the task of editing multiple attributes using a single generator.
2. We use attribute classifier to make the generated image have the expected attributes correctly. Moreover, multi-scale discriminators are used to guide the generator to generate better details, which helps capture more details on the original image.
3. We combine reconstruction loss, attribute classification loss and multi-scale GAN loss in face attribute editing. The experiments on CelebA dataset show that our method performs well in single-attribute face editing, multi-attribute face editing and attribute strength control.

2 Related work

2.1 Encoder–decoder architecture

AutoEncoder (AE) [6] network consists of an encoder and a decoder. Later, the Denoising AutoEncoder (DAE) [14] is proposed to learn the original data of superimposed noise. Variational AutoEncoder (VAE) [15] is a variant of AE mainly used for data generation. In addition, skip connections [16] solve the problems that deconvolution cannot restore image details and the vanishing gradient problem of back-propagation when the network is very deep. And it also accelerates the convergence speed. In this paper, a symmetrical skip connection is added between the encoder and decoder, which is beneficial to improve the training stability and visual quality of the generated image.

2.2 Generative adversarial networks

Generative adversarial networks (GAN) [1] are composed of a generative network and a discriminative network, which can learn by game between the two networks. The generator learns to generate false samples that cannot be distinguished from true samples, while the discriminator learns to determine whether the input sample is true or false.

The optimization process of GAN model is a "mini-max two-player game":

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}\big[\log D(x)\big] + \mathbb{E}_{z\sim p_z(x)}\big[\log(1 - D(G(z))\big] \tag{1}$$

where generator G implicitly defines a probability distribution p_z, and we hope that p_z converges to the real data distribution p_{data}.

There are some problems in GAN, such as unstable training, mode collapse, vanishing gradient and exploding gradient. DCGAN [17] proposes a relatively stable network structure and uses batch normalization to help model convergence. WGAN [18] replaces sigmoid with restricted 1-Lipschitz at the last level of discriminator to realize a "range limitation" function. Using Wasserstein distance can provide meaningful gradient. The loss function proposed by WGAN can avoid mode collapse and improve training stability:

$$\min_G \max_{||D||_L \le 1} \mathbb{E}_{x\sim p_{\text{data}}(x)}[D(x)] - \mathbb{E}_{z\sim p_z(x)}[D(G(z))] \tag{2}$$

In addition, WGAN-GP [13] improves the condition of continuity restriction and proposes a gradient penalty to satisfy Lipschitz continuity condition. In this paper, we use WGAN-GP to optimize GAN loss.

The task of image-to-image translation using GAN-based architectures has achieved impressive results. Pix2pix [19]

uses CGAN between unpaired image data and combines adversarial loss with L1 loss. For unpaired data, additional constraints such as cyclic consistency [20] and shared latent space [21] are used to improve image translation. Pix2pixHD [12] uses a new multi-scale discriminator and coarse-to-fine generator architectures to effectively improve the visual quality of the generated image and generate high-resolution images. Multi-scale discriminators refer to a number of discriminators, which can distinguish true and false images with different resolutions. Multi-scale discriminators are helpful to guide generators to generate better details. They can not only capture the detail information on the original image, but also grasp the global information of the generated image macroscopically and obtain the information of the larger receptive field. In our framework, we use multi-scale discriminators to improve the quality of generation.

2.3 Face attribute editing

Currently, several GAN-based face attribute editing methods have been proposed. IcGAN [8], VAE/GAN [9] and Fader Networks [10] are constructed by combining the encoder–decoder structure with GAN, respectively. Then the attribute-independent constraint may impair the representational ability and result in poor performance of generated images. GeneGAN [22], DNA-GAN [23], and ELEGANT [24] exchange attributes between a given pair of images, where the encoder and decoder act as the generator. In particular, ELEGAN also uses the U-Net structure and multi-scale discriminators for better visual results. SaGAN [25] introduces spatial attention mechanism into GAN framework, which only changed the specific area of attributes and kept the remaining irrelevant areas unchanged. StarGAN [26] performs image-to-image translation for multiple domains using only a single model (a single generator and discriminator) and trains the network using domain classification loss and cycle consistency loss. In AttGAN [7], there is no cycle process or cycle consistency loss. It uses an encoder–decoder architecture and applies attribute classification constraint, reconstruction learning and adversarial learning to network training. In this paper, we use the encoder–decoder structure as the generator and use multi-scale discriminators to improve the quality of the generated images. And the attribute classifier is used to ensure the correct operation of the attribute.

3 Method

The method overview of face attribute editing model proposed in this paper is shown in Fig. 2. For a given input image X_a and attribute value $\mathbf{b}$, the goal of facial attribute editing is to transform X_a into a new image X_b. The detail architectures of our network are shown in Sect. 4.2.

3.1 Model

The data set consists of images and labels with n binary attributes. (The number of attributes set in this paper is $n = 13$, which will be described in detail in Sect. 5.)

3.1.1 Encoder and decoder

The function of the encoder Enc is to map the real input image X_a with n binary attributes $\mathbf{a}$ to a latent representation $\mathbf{z}$:

$$z = \mathrm{Enc}(\mathrm{X}_a) \tag{3}$$

The decoder Dec decodes the latent representation $\mathbf{z}$ and another n binary attributes $\mathbf{b}$ to realize attribute editing, generating the generated image X_b with expected attributes $\mathbf{b}$:

$$\mathrm{X}_b = \mathrm{Dec}(\mathbf{z}, \mathbf{b}) \tag{4}$$

On the other hand, the decoder Dec decodes attribute $\mathbf{a}$ and latent representation $\mathbf{z}$ to reconstruct the original image and generates the reconstructed image $\mathrm{X}_{a'}$ of the real image X_a. The reconstructed image should be as similar as possible to the original image:

$$\mathrm{X}_{a'} = \mathrm{Dec}(\mathbf{z}, \mathbf{a}) \tag{5}$$

3.1.2 Attribute classifier

In order to make the image X_b generated by the real image X_a have the expected attribute $\mathbf{b}$ correctly, the attribute classifier C is used to ensure the correct attribute operation of the generated image:

$$\mathbf{b}' = C(\mathrm{X}_b) \tag{6}$$

The attribute $\mathbf{b}'$ obtained by the attribute classifier C should be similar to $\mathbf{b}$.

3.1.3 Multi-scale discriminators

In this paper, two-scale discriminators are used. The network structure of the two discriminators is identical, but the size of the input image is different. The input of D_1 is the original image of 128×128, which represents the discriminator for processing the larger resolution images. The input of D_2 is the 64×64 image after the down-sampling of the original image, representing the discriminator for processing smaller resolution images. D_1 can capture a lot of detail information on the original image, which is conducive to guiding the

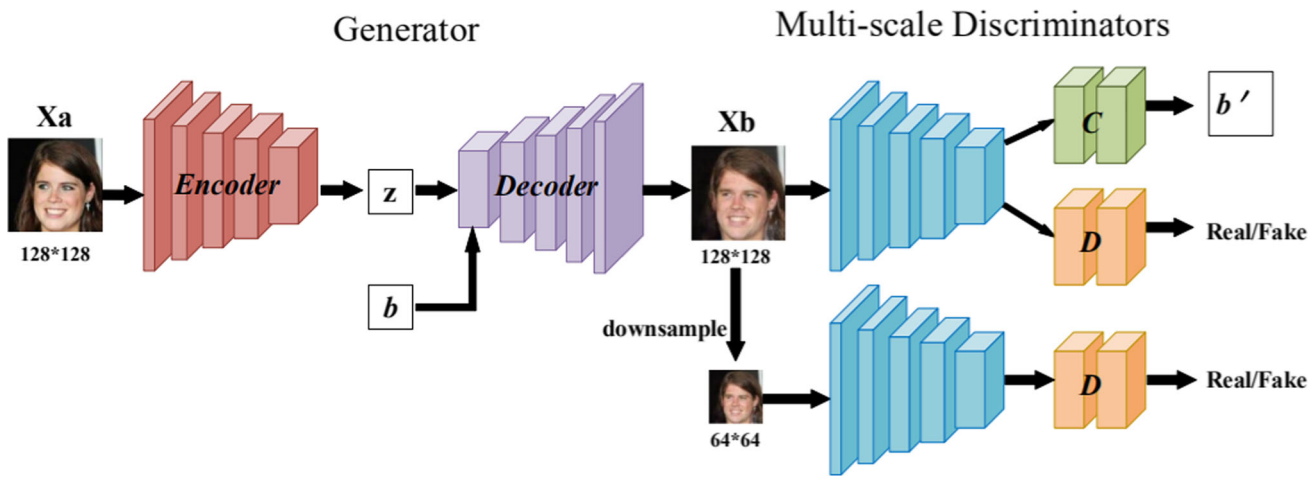

Fig. 2 Network architecture. Our generator contains encoder Enc and decoder Dec. The encoder accepts the image X_a as input and generates a latent representation **z**. The decoder takes **z** and the target domain label **b** as input and generates a fake image. Attribute classifier C learns to classify the attributes of the image. Multi-scale discriminators D_1 and D_2 are utilized to distinguish between the real and fake image

encoder and decoder to generate better details. D_2 can obtain more field information on the original image, so it can grasp the global information of the generated image.

3.2 Loss function

The loss function in this paper includes the following three types of loss: (1) GAN loss, which aims to make the generated image indistinguishable from the real image; (2) reconstruction loss, which is designed to evaluate the effect of reconstruction of the original input image after encoding and decoding; and (3) attribute classification loss, which is added to constrain the model to perform correct attribute operation on the generated image.

3.2.1 GAN loss

In order to avoid mode collapse and training instability, the loss function proposed by WGAN [18] is used in this paper. The adversarial relationship between generator (including encoder Enc and decoder Dec) and multi-scale discriminators can be expressed as the following loss function:

$$\min_{||D_1||_L \leq 1} \mathcal{L}_{\text{adv}_{d1}} = -\mathbb{E}_{X_a \sim p_{\text{data}}}[D_1(\mathrm{X}_a)] + \mathbb{E}_{X_a \sim p_{\text{data}}, \boldsymbol{b} \sim p_{\text{attr}}}[D_1(\mathrm{X}_b)] \tag{7}$$

$$\min_{\text{Enc,Dec}} \mathcal{L}_{\text{adv}_{g1}} = -\mathbb{E}_{X_a \sim p_{\text{data}}, b \sim p_{\text{attr}}}[D_1(\mathrm{X}_b)] \tag{8}$$

$$\min_{||D_2||_L \leq 1} \mathcal{L}_{adv_{d2}} = -\mathbb{E}_{X_a \sim p_{\text{data}}}[D_2(\mathrm{X}_a)] + \mathbb{E}_{X_a \sim p_{\text{data}}, \boldsymbol{b} \sim p_{\text{attr}}}[D_2(\mathrm{X}_b)] \tag{9}$$

$$\min_{\text{Enc,Dec}} \mathcal{L}_{\text{adv}_{g2}} = -\mathbb{E}_{X_a \sim p_{\text{data}}, \boldsymbol{b} \sim p_{\text{attr}}}[D_2(\mathrm{X}_b)] \tag{10}$$

where p_{data} and p_{attr} represent the distribution of real images and attributes, X_a is the original input image, and **b** is the binary attribute. The input image of D_1 is the original image, and the input image of D_2 is the image down-sampled by the original image.

3.2.2 Reconstruction loss

Reconstruction learning requires that the generated image be similar to the original image. In order to ensure that the edited image retains the content of the input image and only changes the relevant parts of the input attributes, the cyclic consistency loss is applied to the generator, which is defined as:

$$\min_{\text{Enc,Dec}} \mathcal{L}_{\text{rec}} = \mathbb{E}_{X_a \sim p_{\text{data}}}\left[||\mathrm{X}_a - \mathrm{X}_{a'}||_1\right] \tag{11}$$

where $\mathrm{X}_{a'}$ is the reconstructed image. In this paper, ℓ_1 loss is adopted in our reconstruction loss.

3.2.3 Attribute classification loss

For a given input image X_a and target attribute **b**, our goal is to convert X_a into the output image X_b with the target attribute **b** and maintain the identity of X_a. Therefore, the attribute $\mathbf{b}'$ obtained by the attribute classifier C of generated image X_b should be similar to **b**, so the loss function of encoder–decoder is as follows:

$$\min_{\text{Enc,Dec}} \mathcal{L}_{\text{cls}_g} = \mathbb{E}_{X_a \sim p_{\text{data}}, \boldsymbol{b} \sim p_{\text{attr}}}\left[\ell_g(\mathrm{X}_a, \mathbf{b})\right] \tag{12}$$

$$\ell_g(\mathrm{X}_a, \mathbf{b}) = \sum_{i=1}^{n} -b_i \log C_i(\mathrm{X}_b) - (1 - b_i)\log(1 - C_i(\mathrm{X}_b)) \tag{13}$$

where $C_i(\mathrm{X}_b)$ denotes the prediction of the i-th attribute of image X_b by the attribute classifier C and $\ell_g(\mathrm{X}_a, \mathbf{b})$ denotes the sum of the binary cross-entropy losses of all attributes.

The attribute $\mathbf{a}'$ obtained from the original image X_a through the attribute classifier C should be approximated to $\mathbf{a}$, so the loss function of the attribute classifier C is as follows:

$$\min_{C} \mathcal{L}_{\mathrm{cls}_c} = \mathbb{E}_{X_a \sim p_{\mathrm{data}}}[\ell_r(\mathrm{X}_a, \mathbf{a})] \tag{14}$$

$$\ell_r(\mathrm{X}_a, \mathbf{a}) = \sum_{i=1}^{n} -a_i \log C_i(\mathrm{X}_a) - (1 - a_i) \log(1 - C_i(\mathrm{X}_a)) \tag{15}$$

where $C_i(\mathrm{X}_a)$ represents the prediction of the i-th attribute of the image X_a by the attribute classifier C.

3.2.4 Overall objective function

Finally, the objective function of optimizing encoder and decoder is as follows:

$$\min_{\mathrm{Enc,Dec}} \mathcal{L}_{\mathrm{enc,dec}} = \lambda_1 \mathcal{L}_{\mathrm{rec}} + \lambda_2 \mathcal{L}_{\mathrm{cls}_g} + \lambda_3 \mathcal{L}_{\mathrm{adv}_{g1}} + \lambda_4 \mathcal{L}_{\mathrm{adv}_{g2}} \tag{16}$$

The objective function of optimizing multi-scale discriminators $\{D_1, D_2\}$ and attribute classifier C is as follows:

$$\min_{D_1, D_2, C} \mathcal{L}_{\mathrm{dis,cls}} = \lambda_3 \mathcal{L}_{\mathrm{adv}_{d1}} + \lambda_4 \mathcal{L}_{\mathrm{adv}_{d2}} + \lambda_5 \mathcal{L}_{\mathrm{cls}_c} \tag{17}$$

where $\lambda_1 \sim \lambda_5$ denote the hyper parameters to balance losses. In our experiments, we set $\lambda_1 \sim \lambda_5$ to 150.0, 15.0, 1.0, 0.5 and 1.5, respectively.

4 Implementation details

4.1 Optimization

In order to stabilize the training process and generate higher-quality images, we use WGAN-GP [13] to optimize GAN losses. Therefore, the objective functions in Eqs. (7) and (9) are represented as follows:

$$\begin{aligned} \min_{\|D_1\|_L \le 1} \mathcal{L}_{\mathrm{adv}_{d1}} = & -\mathbb{E}_{X_a \sim p_{\mathrm{data}}}[D_1(\mathrm{X}_a)] \\ & + \mathbb{E}_{X_a \sim p_{\mathrm{data}}, \boldsymbol{b} \sim p_{\mathrm{attr}}}[D_1(\mathrm{X}_b)] \\ & + \lambda_{\mathrm{gp}} \mathbb{E}_{x^{\sim} \sim p_{\mathrm{data}}^{\sim}} \left[\left(\| \nabla_x D_1(x^{\sim}) \|_2 - 1 \right)^2 \right] \end{aligned} \tag{18}$$

Table 1 Encoder network architecture

Layer	Kernel	Stride	Output shape	BN	Activation
Conv	4×4	2×2	[64, 64, 64]	Yes	Leaky ReLU
Conv	4×4	2×2	[128, 32, 32]	Yes	Leaky ReLU
Conv	4×4	2×2	[256, 16, 16]	Yes	Leaky ReLU
Conv	4×4	2×2	[512, 8, 8]	Yes	Leaky ReLU
Conv	4×4	2×2	[1024, 4, 4]	Yes	Leaky ReLU

Table 2 Decoder network architecture

Layer	Kernel	Stride	Output shape	BN	Activation
DeConv	4×4	2×2	[1024, 8, 8]	Yes	ReLU
DeConv	4×4	2×2	[512, 16, 16]	Yes	ReLU
DeConv	4×4	2×2	[256, 32, 32]	Yes	ReLU
DeConv	4×4	2×2	[128, 64, 64]	Yes	ReLU
DeConv	4×4	2×2	[3, 128, 128]	No	Tanh

$$\begin{aligned} \min_{\|D_2\|_L \le 1} \mathcal{L}_{\mathrm{adv}_{d2}} = & -\mathbb{E}_{X_a \sim p_{\mathrm{data}}}[D_2(\mathrm{X}_a)] \\ & + \mathbb{E}_{X_a \sim p_{\mathrm{data}}, \boldsymbol{b} \sim p_{\mathrm{attr}}}[D_2(\mathrm{X}_b)] \\ & + \lambda_{\mathrm{gp}} \mathbb{E}_{\hat{X} \sim p_{\mathrm{data}}} \left[\left(\| \nabla_{\hat{X}} D_2(\hat{X}) \|_2 - 1 \right)^2 \right] \end{aligned} \tag{19}$$

where $\hat{X}$ is sampled uniformly along a straight line between the generated image and the real image.λ_{gp} is the coefficient of the gradient penalty, which is empirically set to 10.

4.2 Network architecture

The detailed architecture of the generator is shown in Table 1 and Table 2. Among them, the encoder Enc uses five convolution layers, each layer of convolution is followed by BN (batch normalization) [27] and Leaky ReLU, and the decoder Dec uses five deconvolution layers, the first four layers of deconvolution followed by BN [27] and ReLU, and the fifth layer of deconvolution is followed by Tanh. In addition, the model uses the structure of U-Net [16], using symmetrical skip connections between the encoder and the decoder, which can generate better images in the image translation task.

The detailed structure of multi-scale discriminators and attribute classifier is shown in Table 3. The multi-scale discriminators and attribute classifier share five convolution layers, followed by different full connection layers. Each convolution layer is followed by LN/IN and Leaky ReLU, where LN is layer normalization [28] and IN is instance normalization [29]. In Table 3, FC (D_1,D_2) represents the full connection layer of multi-scale discriminators, in which the first layer is closely followed by LN/IN and Leaky ReLU. FC

Table 3 Network structure of multi-scale discriminators and attribute classifier

Layer	Kernel	Stride	Output (D_1)	Output (D_2)
Conv	4×4	2×2	[64, 64, 64]	[64, 32, 32]
Conv	4×4	2×2	[128, 32, 32]	[128, 16, 16]
Conv	4×4	2×2	[256, 16, 16]	[256, 8, 8]
Conv	4×4	2×2	[512, 8, 8]	[512, 4, 4]
Conv	4×4	2×2	[1024, 4, 4]	[1024, 2, 2]
FC(D_1,D_2)			1024	1024
FC(D_1,D_2)			1	1
FC(C)			1024	1024
FC(C)			13	13

(C) represents the full connection layer of attribute classifier C, in which the first layer is followed by LN/IN and Leaky ReLU, and the second layer is followed by Sigmoid.

4.3 Algorithm

The training process of the face attribute editing network is shown in Algorithm 1. We set $n\_d = 5$ to indicate that the generator is updated once and the discriminator is updated five times. In the first stage, the model trains the discriminator networks $\{D_1, D_2\}$ and the attribute classifier C and updates the discriminators and attribute classifier by using adversarial loss and attribute classification loss. In the second stage, the generator network (i.e., encoder–decoder) is trained. The whole training process is completed by back-propagation. The input of the encoder is 128×128 face images, and the input of the multi-scale discriminators is 128×128 original image for D_1 and 64×64 image for D_2, respectively.

4.4 Training settings

The model is trained by Adam optimizer [30] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and batch size is set to 32. The learning rate is initialized to 2×10^{-4} and then decays to 2×10^{-5} for fine-tuning after 100 epochs.

Algorithm 1 Training procedure of the face attribute editing network.

Initialize $it = 0$, and $n\_d = 5$.
for number of training iterations **do**
  if $(it + 1)\%(n\_d + 1)! = 0$ **then**
    Sample real image $\mathrm{X_a} \sim p_{data}$, binary attributes $\mathbf{b} \sim p_{attr}$.
    $\mathrm{X}_b \leftarrow Dec(Enc(\mathrm{X}_a), \mathbf{b})$
    Down-sample the images x by 2 times
    Update the two discriminators D_1, D_2 and attribute classifier C from Eq. (17)
  else
    $\mathbf{z} \leftarrow Enc(\mathrm{X}_a)$
    $\mathrm{X}_{a'} \leftarrow Dec(\mathbf{z}, \mathbf{a})$
    $\mathrm{X}_b \leftarrow Dec(Enc(\mathrm{X}_a), \mathbf{b})$
    Down-sample the images x by 2 times
    Update the encoder network Enc and the decoder network Dec from Eq. (16)
  end if
  Let $it \leftarrow it + 1$
end for

5 Experiments

The experiment in this paper is based on the Pytorch environment, which is implemented on Ubuntu 16.04 operating system with Intel (R) Xeon (R) CPU E5-2678 V3@2.50 GHz and Nvidia RTX 2080Ti-11G GPU (graphics processing unit).

On the CelebA dataset [31], our model and AttGAN were trained for about 8 days. In addition, the training times of IcGAN, Fader Networks and StarGAN were about 4 days, 6 days and 7 days, respectively. The training time of IcGAN is short, because it adopts simple objective function and lacks constraints such as attribute classification loss to train the network. With IcGAN as a baseline, Fader Networks learns attribute-invariant latent representations, so the training time is longer than IcGAN. Both StarGAN and AttGAN take a long time to train because they share some similar objective functions such as attribute classification loss and generate higher-quality images.

5.1 Dataset

CelebFaces Attributes (CelebA) dataset [31] is a large dataset of face attributes, which contains 202,599 face images with 10,177 identities. Each image has 40 binary attributes (yes/no) annotations and 5 landmark locations. CelebA is divided into training set, verification set and test set. In this paper, training set and verification set are used to train the model, and test set is used to evaluate.

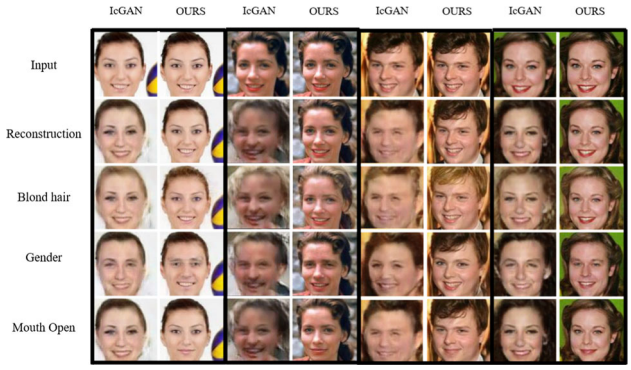

Fig. 3 Comparisons with IcGAN

Fig. 4 Comparisons with the Fader Networks

5.2 Qualitative evaluation

The Adam optimizer with batch size 1 was used in all experiments, setting $\lambda_1 = 10$, $\lambda_2 = 5$. All networks are trained from the ground up, and the learning rate is 0.0002. The same learning rate is maintained for the first 100 training rounds, and the rate is linearly attenuated to zero for the last 100 training rounds.

5.2.1 Single-attribute face editing

In this section, the face attribute editing method proposed in this paper is qualitatively compared with IcGAN [8] and Fader Networks [10]. The experimental results are shown in Fig. 3 and Fig. 4. It can be seen from these figures that the experimental results of face attribute editing using the proposed method have been significantly improved. Figure 5 shows the experimental results of the proposed method in single-attribute face editing.

As seen from Fig. 3, the images generated by IcGAN produce distortion and facial identity changes. This is because the attribute-independent constraint and the normal distribution constraint of the latent representation in IcGAN are excessive, which impair the representation ability of the model and lead to the loss of details. In Fig. 4, Fader Networks performs better than IcGAN in editing attributes accurately. However, since Fader Networks also performs mandatory attribute-invariant operation for the latent representation, it can be seen from the experimental results that many generated images are fuzzy. Some generated images also produce artifacts and loss of details. In addition, it should be noted that when we change the hair color to blonde, the male of Fader Networks in Fig. 4 becomes female. Since most blond characters in the training set are female, the two attributes of blonde and female are highly correlated. Compared with IcGAN and Fader Networks, our method uses an attribute classification constraint instead of the attribute-independent constraint, thus ensuring correct changes of attributes and accurately editing facial attributes.

Fig. 5 Single-attribute face editing result

5.2.2 Multi-attribute face editing

The method proposed in this paper can edit multiple attributes simultaneously. Four groups of experimental results of editing two attributes at the same time are listed in Fig. 6. The first column of each group is the original image, and the second column is the result image of multi-attribute editing. It can be seen that this method still performs well in the case of multi-attribute combination, which is due to the appropriate modeling between the attributes and the latent representation.

5.2.3 Attribute intensity control

Figure 7 shows the results of the attribute intensity control experiment, where the first column is the original image, the first row represents the result of gradual change from "No Pale Skin" to "Pale Skin," and the second row represents the result of gradual change from "Female" to "Male." It can be seen that the image generation is natural and smooth.

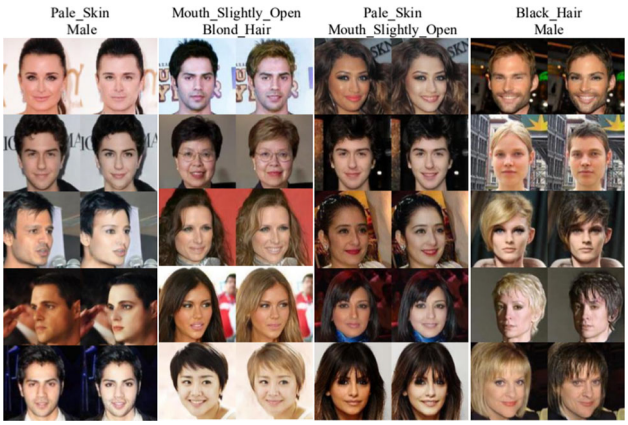

Fig. 6 Multi-attribute face editing results

Fig. 7 Results of attribute intensity control

Table 4 Reconstruction quality of several methods (higher the better)

Methods	IcGAN	StarGAN	AttGAN	Ours
PSNR	15.28	22.80	24.07	25.26
SSIM	0.430	0.819	0.841	0.854

5.3 Quantitative assessment

The performance of attribute editing can be evaluated in terms of image quality. The PSNR/SSIM experiment results of the reconstructed images and original images are shown in Table 4. It can be seen from Table 4 that the reconstruction ability of IcGAN is very limited due to the limitation of training process. Compared with AttGAN, our method can generate better details by using multi-scale discriminators. In addition, it can grasp the overall information of the generated picture macroscopically and obtain more information of the receptive field, so as to achieve better reconstruction.

The PSNR/SSIM results of several methods are shown in Tables 5 and 6. We randomly sampled 2000 images from the test set. For each method, we test these images at different training epochs and obtain their reconstructed images. By calculating the PSNR/SSIM values of the images and their reconstructed images, the average values at different epochs are obtained. It can be seen that with the increase in the number of training epochs, the quality of image generation is constantly improving, and our method outperforms all competing methods.

Table 5 PSNR results of IcGAN, StarGAN, AttGAN and our method (higher the better)

Methods	Epochs			
	50	100	150	200
IcGAN	8.41	10.28	13.30	15.28
StarGAN	14.29	16.71	20.59	22.80
AttGAN	15.43	17.98	22.26	24.07
Ours	16.25	18.87	22.95	25.26

Table 6 SSIM results of IcGAN, StarGAN, AttGAN and our method (higher the better)

Methods	Epochs			
	50	100	150	200
IcGAN	0.335	0.391	0.409	0.430
StarGAN	0.701	0.757	0.803	0.819
AttGAN	0.714	0.787	0.824	0.841
Ours	0.734	0.798	0.836	0.854

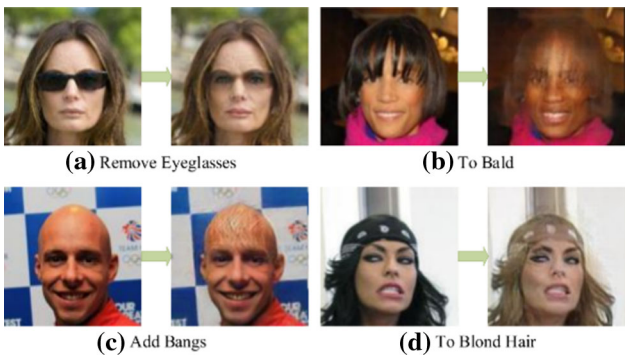

Fig. 8 Failure cases of our method

5.4 Failure cases

Although our method ensures that other details are not changed and the high-quality images are generated when the images are edited correctly, there are still some failure cases, as shown in Fig. 8. When we want to make large appearance changes to the facial image, our results may not be ideal. This will be the problem we need to overcome in the future.

6 Conclusion

In this paper, we propose a conditional generation model, which combines the GAN and the encoder–decoder architecture to realize face attribute editing and generate high visual quality images. The encoder–decoder structure combined with U-Net is used as generator, and the input of the generator is face image and binary attributes. We use attribute classifier to ensure that attributes are correctly changed and multi-scale discriminators to generate better details. The experiments on

CelebA dataset show that our model generates high-quality face images on the basis of having the expected attributes correctly. In the next step, we will try to apply the model to general image editing tasks and study more complex models to further improve the stability of training and image generation quality.

Acknowledgments The authors are very indebted to the anonymous referees for their critical comments and suggestions for the improvement of this paper. This work was supported by the grants from the National Natural Science Foundation of China (Nos. 61673396, U19A2073, 61976245) and the Fundamental Research Funds for the Central Universities (18CX02140A).

References

1. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. Adv. Neural Inf. Process. Syst. 2672–2680 (2014)
2. Denton, E., Chintala, S., Szlam, A., Fergus, R.: Deep generative image models using a laplacian pyramid of adversarial networks. Adv. Neural Inf. Process. Syst. **1**, 1486–1494 (2015)
3. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. Adv. Neural Inf. Process. Syst. 2234–2242 (2016)
4. Berthelot, D., Schumm, T., Metz, L.: BEGAN: boundary equilibrium generative adversarial networks. IEEE Access (2017). https://doi.org/10.1109/ACCESS.2018.2804278
5. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-January, pp. 105–114 (2017). https://doi.org/10.1109/CVPR.2017.19
6. Hinton, G.E., Zemel, R.S.: Autoencoders, minimum description length and Helmholtz free energy. Adv. Neural. Inf. Process. Syst. **6**, 3–10 (1994). https://doi.org/10.1021/jp906511z
7. He, Z., Zuo, W., Kan, M., Shan, S., Chen, X.: AttGAN: Facial Attribute Editing by Only Changing What You Want. arXiv Prepr (2017). arXiv:1711.10678
8. Perarnau, G., van de Weijer, J., Raducanu, B., Álvarez, J.M.: Invertible conditional GANs for image editing. arXiv Prepr (2016). arXiv:1611.06355
9. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. arXiv Prepr. (2015). arXiv:1512.09300
10. Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., Ranzato, M.: Fader Networks: Manipulating images by sliding attributes. Adv. Neural Inf. Process. Syst. 5967–5976 (2017)
11. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv Prepr (2014). arXiv:1411.1784
12. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8798–8807 (2018). https://doi.org/10.1109/CVPR.2018.00917
13. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein GANs. Adv. Neural Inf. Process. Syst. 5767–5777 (2017)
14. Vincent, P., Larochelle, H.: Extracting and composing robust features with denoising autoencoders, pp. 1–23 (2010). https://doi.org/10.1145/1390156.1390294
15. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. arXiv Prepr. (2013). arXiv:1312.6114
16. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9351. Springer, Cham, pp. 234–241 (2015). https://doi.org/10.1007/978-3-319-24574-4_28
17. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv Prepr. (2015). arXiv:1511.06434
18. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. arXiv Prepr (2017). arXiv:1701.07875
19. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. 2017-January, pp. 5967–5976 (2017). https://doi.org/10.1109/CVPR.2017.632
20. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, 2017-October, pp. 2242–2251 (2017) https://doi.org/10.1109/ICCV.2017.244
21. Liu, M.-Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. Adv. Neural Inf. Process. Syst. (2017). https://doi.org/10.1109/ICAIIT.2019.8834613
22. Zhou, S., Xiao, T., Yang, Y., Feng, D., He, Q., He, W.: GeneGAN: Learning object transfiguration and attribute subspace from unpaired data. arXiv Prepr. (2017). arXiv:1705.04932
23. Xiao, T., Hong, J., Ma, J.: DNA-GAN: Learning disentangled representations from multi-attribute images. arXiv Prepr. (2017). arXiv:1711.05415
24. Xiao, T., Hong, J., Ma, J.: ELEGANT: exchanging latent encodings with GAN for transferring multiple face attributes. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11214. Springer, Cham, pp. 172–187 (2018). https://doi.org/10.1007/978-3-030-01249-6_11
25. Zhang, G., Kan, M., Shan, S., Chen, X.: Generative adversarial network with spatial attention for face attribute editing. In: LNCS, vol. 11210, pp. 422–437. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01231-1_26
26. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8789–8797 (2018). https://doi.org/10.1109/CVPR.2018.00916
27. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv Prepr. (2015). arXiv:1502.03167
28. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv Prepr. (2016). arXiv:1607.06450
29. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv Prepr. (2016). arXiv:1607.08022
30. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv Prepr. (2014). arXiv:1412.6980
31. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In Proceedings of the IEEE international conference on computer vision, pp. 3730–3738 (2015). https://doi.org/10.1109/ICCV.2015.425