Journal of Network Intelligence ©2021 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **6**, Number **3**, August **2021**

Efficient Face Attribute Editing Method Based on GAN

Hong-Bin Ma

Electronic Engineering College
Heilongjiang University
No.74 Xuefu Road, Harbin, China
mahongbin@hlju.edu.cn

Xu-Guang Chen

Electronic Engineering College
Heilongjiang University
No.74 Xuefu Road, Harbin, China
cxg190037571@163.com

Ying-Li Wang*

Electronic Engineering College
Heilongjiang University
No.74 Xuefu Road, Harbin, China
*Correspondence: wangyingli@hlju.edu.cn

Pu-Jun Ji

Electronic Engineering College
Heilongjiang University
No.74 Xuefu Road, Harbin, China
jipujun@foxmail.com

Received May 2021; revised July 2021

Abstract. *Facial image editing is a kind of image generation technology. Due to the emergence of generative adversarial networks, this research direction has developed rapidly. Its purpose is to complete the editing of the specified facial attributes while preserving the detailed information in the original image. In the method based on the generative adversarial networks, The important thing is the convergence of the complex network and the stability of the non-target attribute area. For this reason, this paper constructs a dual-module generator in generative adversarial networks, in which module 1 guarantees the editing effect of facial attributes, and the principle of module 2 is similar to positioning, ensuring that only the specified attribute area can be edited. In addition, to optimize the training process of the network, this paper introduces a channel normalization based on the convolution method(CNC). And we combine it with the BatchNorm to perform the normalization operation. Comparative experiments show that the model proposed in this paper can achieve better editing effects while maintaining the stability of non-target attribute regions. In addition, it can be more stable and converge faster in network training.*

Keywords: GAN; Face attribute editing; Feature description; Image generation

1. **Introduction.** Face attribute editing is a common form in image translation, and it is also one of the key research directions of image editing. Image is an important carrier of information [1]. Recently, with the rapid development of short video and other services, users have increasingly relied on image editing technology. Regrettably, some image editing software or systems often have many problems when performing complex image editing techniques such as face attribute editing and style transfer. More serious is that operations such as stitching and stretching are performed based on pixels, which results in obvious editing traces, which are not real and natural enough, and even the details of the original images will be lost [2]. Therefore, it is necessary to construct a network model based on deep features to improve complex image editing techniques such as facial attribute editing.

In early research, facial attribute editing was regarded as a regression problem. In most cases, researchers chose to use paired datasets for regression analysis [3], but such methods could not guarantee the quality of processed images. With the rapid development of generative adversarial networks [4], many excellent attribute editing models have been proposed. They appeared in some fields such as facial animation, facial expression recognition, and enhanced datasets, etc. Pixel2Pixel2 [5] used paired datasets to implement attribute editing and migration between faces, but it had very demanding requirements on the datasets, which limited the diversity of the network model. Later, the emergence of CycleGAN [6] eased the harshness of the datasets during the network training process and realized the face attribute editing under the unpaired datasets. With the development of generative adversarial networks, some scholars have proposed ideas of adding codecs to them. VAE/GAN [7] used an encoder to encode facial images into latent space vectors. For each attribute, the corresponding attribute vector could be obtained. Through the rendering process of the attribute vectors by the decoder, the same identifier with the specified attribute can be obtained. However, this type of network structure required some effort in network training. Otherwise, the authenticity of the images and the effect of attribute editing were not satisfactory. In addition, it did not consider the loss of original irrelevant information in the source image during the training process. IcGAN considers the independence between identity and attributes information. In this model, the encoder and the generative adversarial networks were independent of each other. During the training process, it inputs a random sample from a normal distribution into the generator. This uncertainty will lose the original image details [8]. AttGAN discards the concept of latent variables in the generative adversarial networks. Its encoder only contains a convolution structure without a fully connected layer. The input of the generator is the feature map generated by the convolution structure. Although this method retains the detailed information of the original image, it makes the separation of identity information and attributes information more difficult [9].

Facial attribute editing technology does not require large-scale changes to the attribute vector of the face like face replacement and facial information concealment. Its purpose is to control and edit target attributes without changing other irrelevant face attribute areas. Therefore, two points need to be paid attention to in the design process, namely, the authenticity of the editing results and the stability of other irrelevant attributes areas [10]. Therefore, we build a dual-structure generator to optimize the effect of attribute editing. The function of module one is similar to regular attribute editing. It guarantees the authenticity of the editing effect through confrontation training with the discriminator. The principle of module two is similar to positioning. It guarantees that other irrelevant attribute areas remain stable during the editing process. In addition, we introduced a new normalization method to accelerate the convergence of complex network models, that is, channel normalization based on the convolution method(CNC).

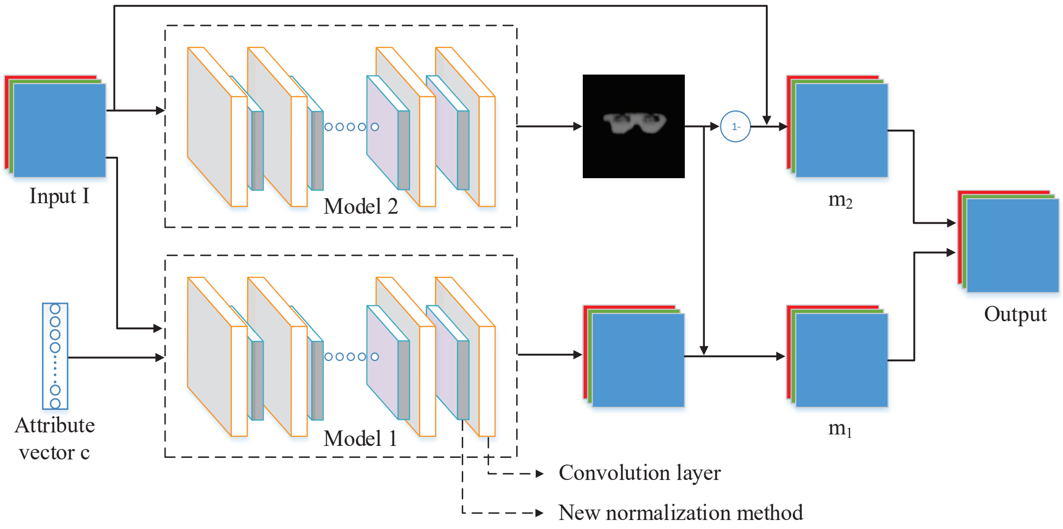

Figure 1. The network structure of generator

In brief, The contribution of this article has three aspects. Firstly, we build a dual-structure generator that uses its positioning convergence feature to implement editing on the target attribute area while ensuring the stability of other irrelevant attribute areas. Secondly, for complex network models, a channel normalization based on convolution is introduced, which can accelerate the convergence of the network during the training process. Thirdly, the proposed method can ensure the authenticity and stability of the editing results and is conducive to the expansion of the datasets.

2. The Structural Design of The Generative Adversarial Networks.

This section introduces the model proposed in this article and describes how it differs from other models. These differences are mainly in three aspects, the design of the generator, the design of the discriminator, and the way of optimizing training. Experiments show that the model proposed in this paper can ensure that other irrelevant attributes do not change while editing the specified facial attribute. And it can also converge quickly during the training process.

2.1. The Network Structure of Generator.

The purpose of the generator is to control and edit the target attributes without changing other irrelevant facial attributes [11]. It contains two modules. The first module guarantees the effect of attribute editing through iterative training, and the second module has a function similar to positioning. It uses a mask map to locate and mark the attribute area that needs to be edited. The network structure diagram of the generator is shown in Figure 1.

It can be seen from Figure 1 that the final generated picture $\widetilde{I}$ is obtained by combining m_1 and m_2, where m_1 is obtained by formula 1:

$$m_1 = G_{M1}(I, c) \cdot G_{M2}(I) \tag{1}$$

Among it, G_{M1} and G_{M2} represent module one and module two in the generator, I represents the input image, and c represents the attribute vector.

m_2 is derived from formula 2:

$$m_2 = I \cdot (1 - G_{M2}(I)) \tag{2}$$

Such a dual-structure approach can facilitate operations on target attributes in attribute-specific areas, while others remain unchanged.

The method of face attribute editing based on generative adversarial networks ensures the authenticity of the generated image with the help of the confrontation between the generator and the discriminator [12]. However, it is not enough to satisfy the authenticity of such problems, and the stability of other irrelevant attributes needs to be ensured. Therefore, this article sets two types of losses in the training of the generator, and the final generator loss function is the superposition of both of them.

Firstly, we designed a loss function L_1^G, which guarantees the authenticity of the image generated by the generator and strengthens the prediction of the target attribute by the generated result. Its expression is as follows:

$$L_1^G = E_{\widetilde{I}}[-\log D(c|\widetilde{I})] + E_{\widetilde{I}}[-\log D(\widetilde{I})] \tag{3}$$

The loss in this part bases on the original generation adversarial network. Where $\widetilde{I}$ represents the realistic image, and c represents the target value.

Secondly, this article quotes a reconstruction loss L_2^G to ensure the stability of regions that are not related to the target attribute. Its expression is as follows:

$$L_2^G = \lambda_1 E_{I,c^o}[(||I - G(I,c^o)||_1)] + \lambda_2 E_{I,c,c^o}[(||I - G(G(I,c),c^o)||_1)] \tag{4}$$

Where c^o is the original attribute list of the input image, c is the target value, and two different λ are freely controllable variables. The loss function of this part has two components. The former is to ensure that the input image will not change when it is manipulated and edited by its original attributes, and the latter is used to ensure that when the edited image $\widetilde{I} = G(I,c)$ is re-operated by the original attribute vector c^o, it should be the same as the original image I.

2.2. **The Network Structure of Discriminator.** The discriminator is the commander of the generator, and its main purpose is to ensure the authenticity of the generated image [13]. In addition, to improve the performance of attribute editing, we have optimized its structure by adding the function of classifying the attributes of the generated images and the real images. The network structure uses the common CNN with the SoftMax function.

The loss function of the discriminator also consists of two parts, namely L_1^D and L_2^D, where L_1^D is used to ensure the authenticity of the generated images. It is optimized based on a standard cross-entropy loss. Its expression is as follows:

$$L_1^D = E_I[\log D(I)] + E_{\widetilde{I}}[\log(1 - D(\widetilde{I}))] \tag{5}$$

Where I represents the input image, and $\widetilde{I}$ represents the result image output by the generator.

In addition, L_2^D is used to supervise the classification of optimized attributes during the training process. It can also be represented by a standard cross-entropy loss. Its expression is as follows:

$$L_2^D = E_{I,c^o}[-\log D(c^o|I)] \tag{6}$$

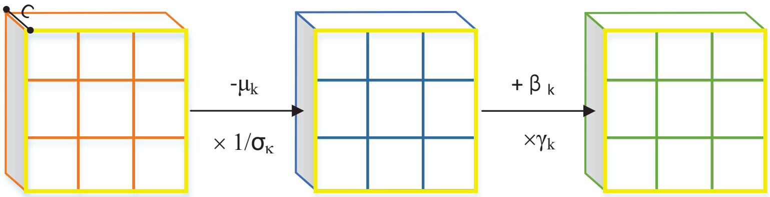

Figure 2. Principle of BatchNorm

3. **Normalization Method.** The self-defined generative adversarial network is easily affected by external conditions during the training process, and it is often difficult to converge. Because it contains multiple training modules, there are a lot of convolutional layers involved, and its requirements for network functions are also very strict. Experiments show that the training is difficult to converge if only the BatchNorm normalization method is used [14], and as the number of training rounds increases, there will be a sudden change in the loss value of the generator after the network converges briefly (we can see the experimental part for details). Therefore, in the normalization method, we use the combination of BatchNorm and CNC.

3.1. **Principle of BatchNorm.** Batch Normalization is a training optimization method proposed by google [15]. In practical applications, it accelerates network convergence by normalizing the data and can control over-fitting to a certain extent. Its working principle diagram is shown in Figure 2.

Although simply normalizing the input of the convolutional layer can remap the input range and change the gradient dilemma, it will affect the features learned by the upper network and hinder the overall training progress. BatchNorm cleverly uses transform reconstruction and introduces learnable parameters $\gamma^{(k)}$ and $\beta^{(k)}$ to solve this potential problem. The algorithm expression is as follows:

$$y^{(k)} = \gamma^{(k)} \left(\frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}} \right) + \beta^{(k)} \tag{7}$$

3.2. **Channel Normalization Based on Convolution Method(CNC).** To achieve fast convergence and improve the performance of generalization and robustness, this paper introduces a channel normalization based on the convolution method, which uses the translation invariance characteristics of the convolution operator. In principle, it first normalizes each output channel of each layer for convolutional networks. This part is the re-parameterization of the kernel. It normalizes the weight of each channel into a tight frame.

The whole process will involve complex matrix operations, but fortunately, it can be mapped to the Fourier transform domain for execution, which reduces the amount of computation. After performing the above operations, immediately use another kernel that can be learned to complete the affine transformation of the output result of the previous step. This part is somewhat different from the principle of the BatchNorm. BatchNorm only learns two parameters in the normalization process, which is not representative. Compared with it, CNC needs to learn the entire kernel.

It uses the pre-calculated kernel v to convolve each channel, normalizes the frequency spectrums of the weight matrixes of the convolutional layers, and then uses the learned kernel r as an affine transformation to perform channel convolution. In actual calculations, CNC maps some complex matrix operations to the Fourier domain, which greatly reduces the complexity of operations. Many articles also use similar Fourier domain mapping methods [16]. The specific algorithm is shown in Algorithm 1.

Algorithm 1 Channel normalization based on convolution method

```
Description: $z \in R^{BatchSize \times C_o \times W \times H}$ represents the output of the previous layer, which is the input of this layer; Both $a$ and $r$ are convolution kernels that can be learned;
$z_{out} = Conv2D(z)$
for k in [1, ..., $c_o$] do
    // $c_o$ is the number of output channels of the convolutional layer
    $\hat{z}_{k,out} \leftarrow FFT(z_{k,out})$
    $\hat{a}_k \leftarrow FFT(a_k)$
    // $a_k$ is the weight data of this channel
    $\hat{a}_k \leftarrow stop\_gradient(\hat{a}_k)$
    $\hat{v}_k \leftarrow (\sum_{m=1}^{C_i} |\hat{a}_{km}|^{\odot 2})^{\odot -1/2}$
    $\tilde{z}_{k,out} \leftarrow IFFT(\hat{z}_{k,out} \odot \hat{v}_k)$
    $\bar{z}_{k,out} \leftarrow r_k * \tilde{z}_{k,out}$
end for
Return: $\bar{z}_{k,out}$
```

4. **Experimental Details.** This section verifies the feasibility of the model proposed in this paper through experiments. We first introduced the environment in which the program run, and then detailed the dataset used to train the network. Finally, we made a detailed experimental comparison with the existing facial attribute editing model through three aspects. The results show that the model proposed in this paper is more effective in editing facial attributes.

4.1. **Experimental Environment.** This experiment is run on a window system and implemented with PyTorch as a framework for network learning. The training and implementation process of the neural network is mainly carried out on the GPU, the graphics card model is RTX2060, and the graphics card memory is 6GB.

The training dataset used in the experiment in this article is CelebA, which is a kind of open dataset of the Chinese University of Hong Kong. The dataset contains 202,599 color face images from 10,177 celebrity identities, and each picture corresponds to a binary attribute vector containing 40 attributes. The face images in this dataset cover many pose changes and various complex background situations. It can be applied to a variety of processing tasks in the field of computer vision.

4.2. **Comparison of Experimental Results.** Now, we will verify the attribute editing effect of the model proposed in this article, and then analyze the role of feature maps in attribute editing. The feature map is the product of the generator, which is created by Model 2 during the training process. Finally, we compare the editing effects of the model in this article with the existing mature models. In addition, all these methods use the same training data for training.

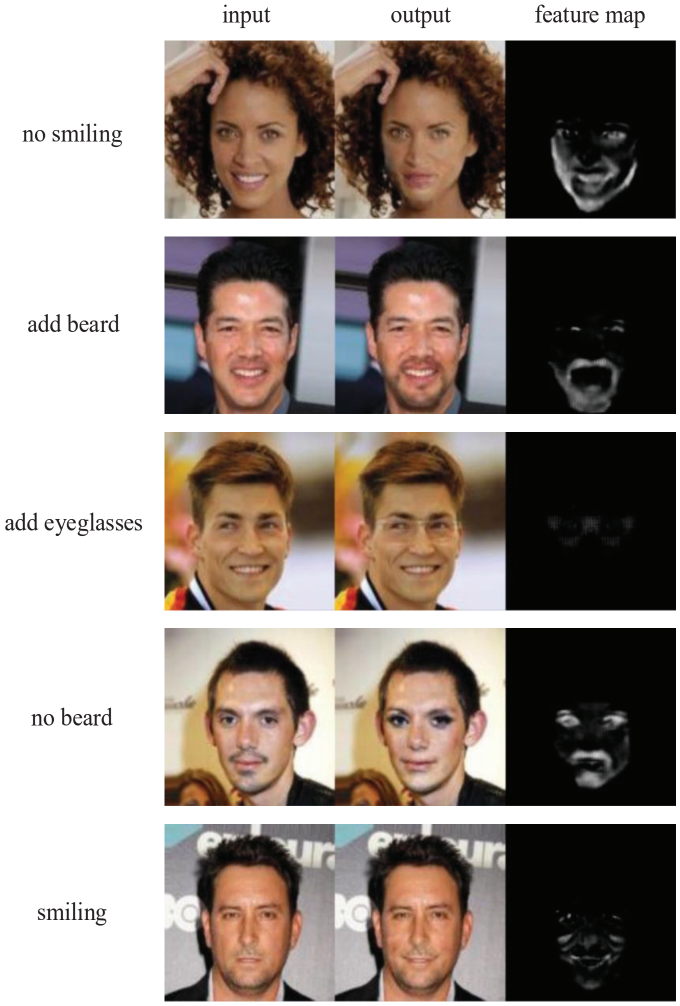

Figure 3. The effect of face attribute editing

4.2.1. *Attribute Editing Effect Comparison.* The common evaluation indexes of generative adversarial networks are Inception Score (IS) and Frechet Inception Distance (FID). Both of them rely on the Inception classification network to weigh the visual authenticity and category diversity of the generated pictures, so they are not suitable for the facial attribute editing tasks involved in this article.

We first use intuitive visual effects to show the facial editing effects of the model in this article. It can be seen from Figure 3 that the model in this paper has a good effect on attribute editing, and in the editing process, other irrelevant attributes are also retained. This section compares the model proposed in this paper with the existing face attribute editing models in the presentation of the CelebA, including AttGAN [17] and StarGAN [18]. The detailed comparison is shown in Figure 4.

It can be seen from Figure 4 that when StarGAN realizes the task of facial attribute editing, the generated facial image has obvious partial blur, which is especially obvious when adding glasses and other complex operations. The results show that the more difficult the editing task, the worse the quality of the image produced [19]. During the test, AttGAN will have a similar phenomenon, but the probability of this is low. For example, this phenomenon appeared when editing a smiling emoticon in the picture. In addition, it can be found from the comparison of Figure 4 that the model proposed in this

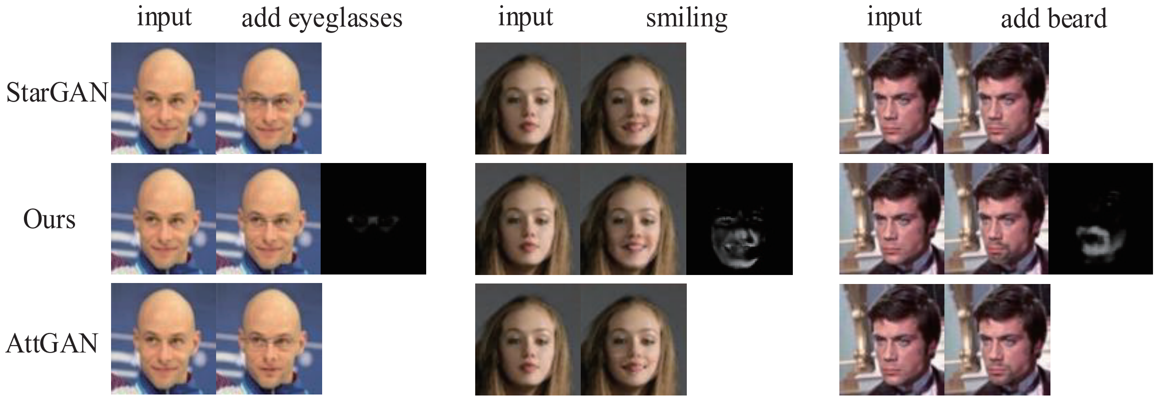

Figure 4. Comparison of the effect of face attribute editing

paper can retain irrelevant attributes when editing facial attributes. There are roughly two problems with other models. The first is improper processing of the area around the target attribute, which will cause a large number of spots in the image and affect the final quality. The second is that the stability of the non-target attribute area cannot be guaranteed during the editing process. For example, when the operation of adding glasses is performed in AttGAN, the expression of the character is changed.

In addition, to improve the accuracy of comparison, we set the accuracy of attribute editing as a judgment index. It is determined by the attributes of the input image, the specified attributes, and the attributes of the output image, where a represents the attribute of the input image, b represents the specified attribute, and c represents the attribute of the output image. A and b should only have different values in the kth dimension, and in this dimension, c and b have the same value. In addition, when the similarity between c and a reaches a certain threshold, we consider this to be an accurate face editing task. In the experiment, 1000 pictures in the dataset were randomly edited and tested, and the comparison result is shown in Figure 5.

It can be seen from Figure 5 that when dealing with some simple facial attribute editing tasks, the networks listed can achieve high accuracy, but the model proposed in this article can also achieve high accuracy in some complex situations. The graph shows that our model only lost to the AttGAN model when editing the smile emoticon.

4.2.2. *Comparison of The Impact of Normalization Methods on Training.* To test the influence of the combined normalization method on the training process, we use a separate BatchNorm normalization method, a separate CNC, and a combination of the two to train the network. The changes in network performance during the training process are recorded, and the comparison results are shown in Table 1.

In the early stage of network training and when the training process collapses, the edited images will have some redundant outlines, resulting in poor image editing effect and quality. Therefore, there is no way to judge the accuracy of attribute editing. The results of this part are replaced by in Table 1. In addition, I in Table 1 represents the number of iterations, C represents the normalization method, and the accuracy is from the previous section. It can be seen from Table 1 that if only the BN normalization method is used in the network, some uncontrollable problems will occur, such as sudden changes in the loss value of the generator, poor overall stability, and slow convergence

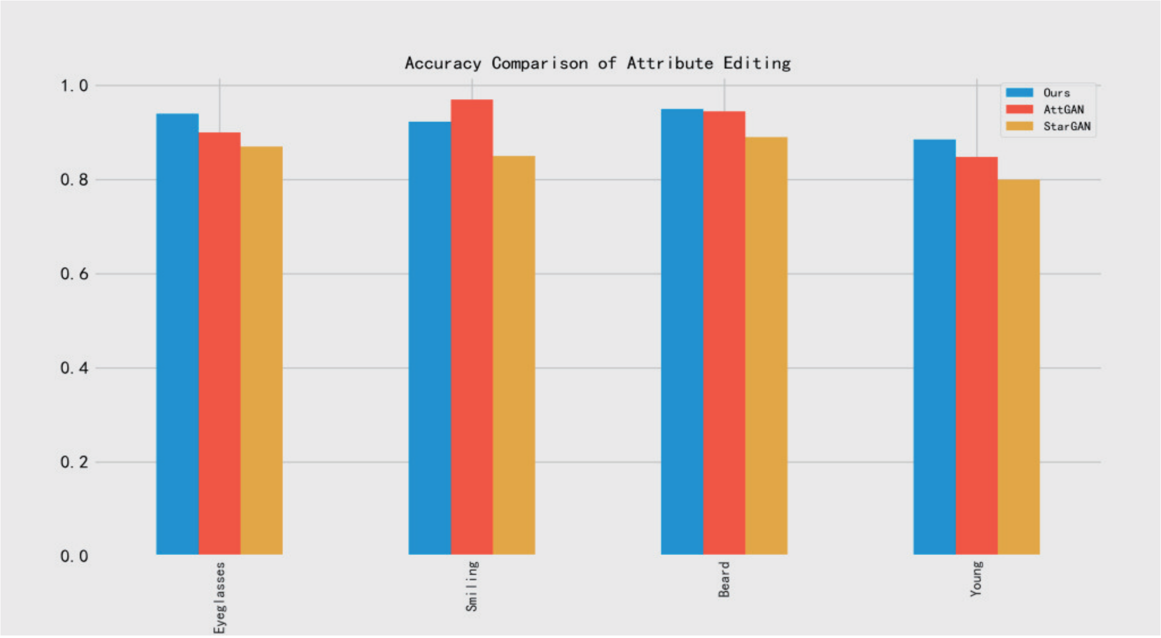

Figure 5. Accuracy of different models

Table 1. Comparison of training results of different normalization methods

g-loss				Accuracy(%)		
I\C	BN	CNC	BN+CNC	BN	CNC	BN+CNC
200	4.0353	5.0141	4.2373	\	\	\
1000	0.3012	0.2196	0.2735	\	\	\
2000	0.3517	0.0835	0.0441	73.44	82.86	88.76
3000	0.1502	0.0359	0.0219	82.38	86.43	90.21
4000	0.0215	0.0213	0.0182	91.96	94.02	92.93
5000	15.3326	0.0195	0.0197	\	92.85	93.27

during training. When the CNC normalization method proposed in this paper is used alone, the problem of loss mutation is solved, and the network is easier to converge during the training process. When the two normalization methods are combined, the network can converge faster, greatly improving the efficiency of the training process.

5. **Conclusions.** To achieve an effective face attribute editing method, we construct a generative adversarial network with a dual-structure generator. The first module of the generator is mainly responsible for the effect of attribute editing. The second is responsible for ensuring that only the specified attribute area can be edited and that other attributes remain unchanged. In addition, to make the network training more efficient and converge faster, we introduce a channel normalization based on the convolution method and combine it with BatchNorm to complete the network training. Experiments show that this normalization method can make the network faster and more stable. In brief, the effect of facial attribute editing based on the model in this paper is very outstanding. However, the model proposed in this paper also has many aspects that need to be improved. For example, the effect is not very good when editing multiple attributes. This situation will

have an impact on the practicality of the model. In addition, compared to the improvement of the generator, the function of the discriminator is single, which can easily lead to an imbalance in the process of network training.

REFERENCES

[1] T. Y. Wu, X. Fan, K. H. Wang, J. S. Pan, and C. M. Chen, Security Analysis and Improvement on an Image Encryption Algorithm Using Chebyshev Generator, *Journal of Internet Technology*, vol. 20, no. 1, pp. 13-23, 2019.

[2] E. K. Wang, X. Zhang, F. Wang, T. Y. Wu, and C. M. Chen, Multilayer Dense Attention Model for Image Caption, *IEEE Access*, vol. 7, pp. 66358-66368, 2019.

[3] T. Portenier, Q. Hu, A. Szabo, FaceShop: Deep Sketch-based Face Image Editing, *ACM Transactions on Graphics*, vol. 34, no. 4, 99, 2018.

[4] I. J. Goodfellow, A. Pouget-Abadie, M. Mirza, Generative Adversarial Nets, *MIT Press*, 2014.

[5] P. Isola, J. Y. Zhu, T. Zhou, Image-to-Image Translation with Conditional Adversarial Networks, *arXiv preprint*, arXiv:1611.07004v3, 2017.

[6] J. Y. Zhu, T. Park, P. Isola, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, *Institute of Electrical and Electronics Engineering*, 2017.

[7] L. A. Larsen, S. K. Sønderby, H. Larochelle, Autoencoding beyond pixels using a learned similarity metric, *International conference on machine learning. PMLR*, pp.1558-1566, 2016.

[8] S. Liu, D. Li, T. Cao, GAN-based face attribute editing, *IEEE Access*, vol. 8, pp. 34854-34867, 2020.

[9] E. K. Wang, C. M. Chen, M. M. Hassan, A. Almogren, A deep learning based medical image segmentation technique in Internet-of- Medical-Things domain, *Future Generation Computer Systems*, vol. 108, pp. 135-144, 2020.

[10] Z. Xu, X. Yu, Z. Hong, FaceController: Controllable Attribute Editing for Face in the Wild, *arXiv preprint*, arXiv:2102.11464, 2021.

[11] L. Liu, L. Chen, J. J Yan, Design of Real-time Face Position Tracking and Gesture Recognition System based on Image Segmentation Algorithm, *Journal of Network Intelligence*, vol. 5, no. 4, pp. 226-239, 2020.

[12] Y. Shen, J. Gu, X. Tang, Interpreting the latent space of gans for semantic face editing, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.9243-9252, 2020.

[13] C. Gadea, M. Trifan, D. Ionescu, A microservices architecture for collaborative document editing enhanced with face recognition, *2016 IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp.441-446, 2016.

[14] J.M.T. Wu, Z. Li, N. Herencsar, B. Vo, J.C.W. Lin, A graph-based CNN-LSTM stock price prediction algorithm with leading indicators, *Multimedia Systems*, 2021, https://doi.org/10.1007/s00530-021-00758-w

[15] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, *International Conference on Machine Learning*, pp.448–456, 2015.

[16] A. Trockman, J. Z. Kolter, Orthogonalizing convolutional layers with the cayley transform, *arXiv preprint*, arXiv:2104.07167, 2021.

[17] Z. He, W. Zuo, M. Kan, S. Shan, X. Chen, AttGAN: Facial Attribute Editing by Only Changing What You Want, *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5464-5478, 2019.

[18] Y. Choi, M. Choi, M. Kim, StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation, *arXiv preprint*, arXiv:1711.09020, 2018.

[19] Y. Jo, J. Park, SC-FEGAN: face editing generative adversarial network with user's sketch and color, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp.1745-1753, 2019.