Generative Adversarial Network for Multi Facial Attributes Translation

JUN LIU¹, XIAOYANG LIU¹, AND YANJUN FENG²

¹School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang 110159, China
²School of Information Science and Engineering, Shenyang Ligong University, Shenyang 110159, China

Corresponding authors: Xiaoyang Liu (67192856@qq.com) and Yanjun Feng (braverfyj@126.com)

This work was supported in part by Liaoning Provincial Natural Fund Guidance Program Projects under Grant 2019-ZD-0251, and in part by the Education Department of Liaoning Province Research Projects under Grant LG201909.

ABSTRACT

Recently, Generative Adversarial Network (GAN) based approaches are applied in facial attribute translation. However, many tasks, i.e. multi facial attributes translation and background invariance, are not well handled in the literature. In this paper, we propose a novel GAN-based method that aims to get the target image that performs better within modifying one or more facial attributes in a single model. The model generator learns multi-points by inputting a re-coded transfer vector, ensuring the single model could learn multiple attributes simultaneously. It also optimizes the cycle loss to enhance the efficiency of transferring multi attributes. Moreover, the method uses the adaptive parameter to improve the calculation method of the loss function of the residual image. The results are also compared with the StarGAN v2, which is the current state-of-the-art model to prove the effectiveness and advancedness. Experiments show that our method has a satisfactory performance in multi facial attributes translation.

INDEX TERMS
GAN, deep learning, facial attribute translation, loss optimization.

I. INTRODUCTION

With the improvement of generative models, such as the Generative Adversarial Networks (GANs) [1]–[3] and variational autoencoders (VAEs) [4], the processing of image-to-image translation have made considerable progresses. The translation includes photo style translation [5], [6], objects dyeing [7]–[9], and also facial synthesis [10]–[13].

Facial attribute translation is the process of changing the facial attribute in a given facial image (e.g., removing the glasses or adding one). And multi facial attributes translation could transfer more than one attribute in a given image simultaneously with a single model.

However, most existing methods cannot translate multi selected facial attributes in a single model simultaneously. Some of them employ two sets of images to train a dual generator, where the first one is for translating images from the source domain to the target domain, and another one learns backwards [6], [8], [14]–[17]. In this way, the number of generators increases exponentially if all the training attributes are put together. Others like StarGAN [18], [19] meet this requirement, but some problems are still encountered in practice. Method in [18] and [19] must change all the attributes to a learned result but cannot keep some attributes unchanged (Fig 1 illustrates its limitation). Meanwhile, because of the encoder-decoder process within the generator, some image details (pixel information) must be lost after passing through the generator such as, the background from the facial images, which are not relevant to attribute translation. Moreover, the size of the area that needs to be changed for different attributes is different. Therefore, it is unreasonable to use the same parameter for the residual loss function of different attributes.

In this paper, we propose a novel generative adversarial network that aims at modifying one or more facial attributes in a single model. Like StarGAN, our model takes both images and vectors as inputs and trains the mapping function for multi-attribute transformation. Specifically, we construct a sparser transfer vector by an elaborate combination of original attribute labels and target attribute labels. Such a vector is more convenient to change any specified attributes and keep the other attributes unchanged. We adopt the idea of learning residual images with corresponding loss functions to force the generator to generate a residual image and improve it using an adaptive parameter, which retains the original image’s detail well. We design a better cycle loss function.
Normally, if we train a dual model for translating only one attribute (i.e. [18], [20]), the model could learn to transfer this attribute only and barely influence other parts of the image. But for a multi attributes translation model, the attributes always affect other parts of the image. Such like fig. 1(b) shows, when the model tries to add eyeglasses to the face, the hair colour and the image background are also affected unexpectedly.

to alleviate the effects of interclass competition. We evaluate our method in the large-scale CelebFaces Attributes (CelebA) dataset [21]. Experiments show our approach can transfer multiple attributes efficiently, and generated results have a better visual effect.

This paper makes the following contributions:
1) We propose a novel GAN based framework with a residual layer to retain as many details of the original image.
2) We propose a transfer vector mechanism for multi facial attributes translation, which has a state-of-art performance.
3) We optimize the discriminator with a pre-trained VGG-face model to keep the effectiveness of transferring more than one attribute simultaneously.

The remainder of the paper is organized as follows. In Section 2, the related work is given in detail, including the GAN-based works, the development of the image-to-image translation, and the facial synthesis. Section 3, the proposed method, is described in detail. In Section 4, both experiments and results are discussed. Section 5 concludes this work and gives a further research plan.

II. RELATED WORK
A. GENERATIVE ADVERSARIAL NETWORKS
Generative adversarial networks (GANs) [3] are effective in image generation [6], [22]–[25]. A standard GAN model learns a discriminator that tries to classify the input image as real or fake (from the generator). While simultaneously training a generative model to generate the image confusing the discriminator. Soon after, conditional GANs (cGANs) have been proposed. The generator of cGANs approximates the dependency of the controlled attributes and their corresponding targets. Various conditional image generation applications have shown promising results, such as, attributes transfer [23], [26], texts to image [24], [27], style transfer [6], [13], [28], and video prediction [25], [29], [30].

B. IMAGE-TO-IMAGE TRANSLATION
Recent approaches [9], [14], [31]–[33] of image-to-image translation use paired input-output examples to learn a mapping function applying a GAN based method. And more recent approaches [6], [8], [15]–[17] use the unpaired setting to learn a parametric translation function that does not need paired data samples. The supervised data is seldom available, so unpaired image-to-image translation has a wider application. All of them have two generators that can map each domain to its counterpart domain. Very recently, many state-of-the-art works are proposed, such as [34], [35], and [36].

C. FACIAL SYNTHESIS
Since image-to-image translation has shown a remarkable capability, facial synthesis also could be solved by such approaches. Besides facial attributes translation, several GAN based methods have also been developed for other facial generative tasks. Age progression [11], [37]–[40] is the process of aesthetically rendering a given facial image to represent an ageing effect. Applying for makeup [13] transfer the makeup style from one facial image to the other one.

As for facial attributes translation, [12], [41], [42] emphasizes the importance of retaining identity. [6], [18], [20] use the idea of cycle consistency to try to find a map that could turn the translated image back. Our facial attribute translation process also takes cycle consistency into account. But we further enhance it by adding extra loss constraints, which makes our method more potent in dealing with the core issues of multiple facial attribute translation.
III. APPROACH

A. OVERVIEW

The generative adversarial network is introduced in [43] and [42]. A classic GAN consists of a generator network $G$ and a discriminator $D$. Usually, they are convolutional neural networks. It is an adversarial process to let the generator $G$ learn a distribution $P_G(x)$ that matches the real data distribution $P_{data}(x)$. Concretely, the objective function of GAN can be written as:

$$V(G, D) = \min_G \max_D \left( \mathbb{E}_{x \sim P_{data}(x)} \log[D(x)] + \mathbb{E}_{z \sim P_z(z)} \log[1 - D(G(z))] \right)$$  \hspace{1cm} (1)$$

where $x$ and $z \sim \mathcal{N}(0, 1)$ denote a real input image and a noise sample, respectively. On convergence, the generator effectively learns the data distribution of real images and makes it possible to generate images with target attributes. Based on the principle of GAN, we design a new generator and discriminator to handle the facial attributes translation.

Our goal is to learn one mapping function $G$ that can change multi facial attributes. We first describe the input of $G$, which includes original images and transfer vectors. Then, we discuss how to use the adaptive parameter to improve the residual loss. Besides, an optimized cycle loss is proposed to perfect result images. The whole architecture of our method is presented in Fig. 2.

B. GENERATOR

The motivation of our approach is changing multi attributes in a single model and keeping other attributes unchanged. The inputs of the generator consist of two parts: original images and transfer vectors representing transfer attributes. Usually, the attribute label is a binary element, such as 0 means removing eyeglasses and one means wearing eyeglasses.

In existing methods, we take an image with eyeglasses as input of $G$ and a vector representing wearing eyeglasses. The style of the glasses in the generated image will change. This is because that the model may learn some common eyeglasses and will add these common eyeglasses into the original image when it finds the label of eyeglasses is set to 1 (one example is shown in Fig. 8). To tackle this problem, we use two elements to present a binary feature. We can set 00 to keep this attribute unchanged for a specific attribute, ten present having this attribute and 01 present removing it. In the training phase, the transfer vector can be calculated by the original vector $c_{ori}$ and the target vector $c_{tar}$, i.e., $f(c_{ori}, c_{tar}) \rightarrow v$. The transfer example is shown in Fig. 4. In the test phase, the vector can be set artificially, only the elements corresponding to the attributes to be changed need to be set, and the original vector does not need to be obtained. The mapping function can be applied as $G(x, v) \rightarrow y$.

Meanwhile, we hope the generator only changes the part of the image related to facial attributes and remain other parts unchanged. But a normal generator with several convolutions and deconvolution layers can change other details in generated images which is undesirable. Thus, we adopt the residual method in [x] to first force the generator to generate the residual image. At the end of the architecture, add the residual image $r$ and original image $x$ together to generate the target image, $G(x, v) \rightarrow r$ and $r + x \rightarrow y$. Fig. 3 shows the residual image, which consists only of the key part.
The residual image should have a sparseness constraint in training progress. In method \( [x] \), it is an L-1 norm regularization for residual image.

\[
\mathcal{L}_{spr} = \| r \|_1
\]

(2)

For a similar requirement with \([x]\) of single attribute transformation, this constraint is reasonable. But in the condition of a multi-attribute transfer corresponding to our method, it is unreasonable to use the same degree of sparsity constraint for different attributes. The change range of residual images of changing hair colour is much more extensive than that of changing eyeglasses. Using the same constraint parameters is likely to make the hair colour change not thorough enough, and the eyeglasses change too much. Therefore, we have to vary parameters for different attributes. The category of attribute determines the size of the parameter. At the same time, for multiple attributes, the parameter value can be accumulated. When this loss is smaller, the constraint for the generated residual image is smaller, and the area changed by the generated residual image is larger. So, we have the loss function of sparseness constraint as follows:

\[
\mathcal{L}_{spr} = \left( \sum_{i=0}^{l} \| v_i s_i \| \right) \| r \|_1
\]

(3)

where \( l \) is the length of the transfer vector \( v \), \( s \) is a vector in which the measurement is \( l \), and the value of each element in \( s \) depends on the attribute corresponding to that position in \( v \). For example, the value of the element corresponding to the hair colour is 5, and that of eyeglasses is 1. To be more stable during training and to generate higher quality results, we replace the log-likelihood objective with WGAN-GP \([44]\) defined as:

\[
\mathcal{L}_{GAN-GP} = \mathbb{E}_{x,v} \left[ D_{src}(G(x, v)) \right]
\]

(4)

where \( D_{src} \) is part of the discriminator and would be illustrated in next section.

This network contains two convolutions with stride 2, 6 residual blocks \([45]\), two fractionally-strided convolutions with stride 1/2. Similar to \([46]\), we use instance normalization \([47]\). The architecture is shown in Table 1.

| Layer | Activation size |
|-------|-----------------|
| Input | \((3 + n_a) \times 128 \times 128\) |
| 7 × 7 × 64 conv, pad 1, stride 1, IN, ReLU | 64 × 128 × 128 |
| 4 × 4 × 128 conv, pad 1, stride 2, IN, ReLU | 128 × 64 × 64 |
| 4 × 4 × 256 conv, pad 1, stride 2, IN, ReLU | 256 × 32 × 32 |
| Residualblock 3 x 3 x 256, pad 1, stride 2 | 256 × 32 × 32 |
| Residualblock 3 x 3 x 256, pad 1, stride 2 | 256 × 32 × 32 |
| Residualblock 3 x 3 x 256, pad 1, stride 2 | 256 × 32 × 32 |
| Residualblock 3 x 3 x 256, pad 1, stride 2 | 256 × 32 × 32 |
| Residualblock 3 x 3 x 256, pad 1, stride 2 | 256 × 32 × 32 |
| 4 × 4 × 128 deconv, pad 1, stride 2, IN, ReLU | 128 × 64 × 64 |
| 4 × 4 × 64 deconv, pad 1, stride 2, IN, ReLU | 64 × 128 × 128 |
| 7 × 7 × 3deconv, pad 3, stride 1 | 3 × 128 × 128 |

**TABLE 1. Generator network architecture.**

C. CYCLE LOSS

Typically, the cycle loss in \([6]\) has a good performance applied in a single attribute translation model. But it is insufficient for multi-attribute translation. When there have more than one attribute need to change, the result seems to be not satisfactory. So that we set a constraint to ensure a two-attribute transfer has the same result as using the same transfer twice and once only change one attribute. The loss function for the generator in this phase is expressed as:

\[
\mathcal{L}_{cyc} = \mathbb{E}_{x,v,v'} \left[ \| x - G(x, v), v' \|_1 \right] \\
+ \lambda_{cyc1} \mathbb{E}_{x,v_1,v_2,v_{12}} \left[ \| G(x, v_1), v_2 - G(x, v_{12}) \|_1 \right] \\
+ \lambda_{cyc2} \mathbb{E}_{x,v_1,v_{12}} \left[ \| x - G(x, v_{12}), v'_{12} \|_1 \right]
\]

(5)

As shown in Fig 6, \( G \) change the original images \( x \) to target images \( G(x, v) \) by transferring its original attributes to those in vectors \( v \). And \( v' \) presents the transfer vectors which could transfer the target image to the original image and get the new image \( G(G(x, v), v') \). Ideally, we hope the new image is the same as the original image. Thus, we apply the L-1 norm as this loss function. Specifically, \( v_1 \) and \( v_2 \) mean the vector-only transfer one attribute. \( v_{12} \) present the corresponding vector for transferring both attributes, and \( v'_{12} \) present the opposite transfer vector. We constraint that the image generated by one transfer vector \( (v_{12}) \) should have the same result as the image generated by two transfer vectors \( (v_1 \text{ and } v_2) \). And also, it should be the same as the original image. Unlike CycleGAN \([6]\) translating images by two generators, we use one generator twice to generate the images which have target attributes and then translate them back to the original images.

D. DISCRIMINATOR

Our discriminator \( D \) consists of two parts to represent the probability of reality \( D_{src}(x) \) and to classify the attribute \( D_{cls}(x) \). The discriminator \( D_{src}(x) \) is introduced by a minimax two-player game in which generator \( G \) tries to capture the
underlying data density and fool $D_{src}(x)$ to make a mistake about whether the data is realistic or from G. While $D_{ori}(x)$ optimizes itself to distinguish them. Moreover, we need another discriminator $D_{cls}(x)$ for classifying multi attributes.

The loss function of $D_{src}(x)$ can be written as:

$$\mathcal{L}_{GAN-D} = \mathbb{E}_{x,v}[D_{src}(G(x,v))] - \mathbb{E}_{x}[D_{src}(x)] + \lambda_{gp}\mathbb{E}_{\tilde{x}}([\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1]^2),$$

(6)

where $\tilde{x}$ is sampled uniformly along a straight line between a pair of real and a generated images, when we input image $x$ and transfer vector $v$ into the generator, we want $G(x,v)$ to generate the image having target attributes. So we need a classifier $D_{cls}(x)$ to classify the attributes correctly and be a constraint condition to optimize G. The classifier $D_{cls}$ also need to optimize during the training. To achieve this, we apply two loss functions to the classifier to optimize G and D.

The classification loss of real image used to optimized G is defined as:

$$\mathcal{L}_{cls-r} = \mathbb{E}_{x,c}[-\log(D_{cls}(c|G(x,v)))]$$

(7)

where $x$, $v$, $c$ represent the input image, transfer vector and target attribute label, respectively. The input image $x$, original label $c$ and target attribute label $c'$ are given in training so that the attribute vector $v$ can be calculated by label $c$ and $c'$. And the classification loss of real image used to optimize $D_{cls}$ is defined as:

$$\mathcal{L}_{cls-f} = \mathbb{E}_{x,v,c'}[-\log(D_{cls}(c'|G(x,v)))]$$

(8)

Moreover, we imply a novel discriminator structure to enhance the image quality of multiple attributes transfers. Specifically, we use the pre-trained model VGG-Face as part of the initial parameters in our network. Then we adopt a feature pyramid network to extract features from different scales. As shown in Fig. 5, we extract features individually from several layers in the VGG-Face model, concat all feature maps at the end of each convolutional layer, and predict results by two vectors. One vector having size $1 \times 1$ represents the output of $D_{src}$ and another one having size $1 \times n_c$ means the result of $D_{cls}$ ($n_c$ is the number of label $c$). Practically, we use the feature maps of the 2nd, 4th, 7th and 10th convolution layers to be the branch for the pyramid architecture. The whole architecture of D is shown in Table 2. After pre-training to get the VGG-Face, we believe that the features of each block in VGG-Face should be fully utilized. So the pyramid structure is adopted, which can retain the features extracted by the original network and adapt to the new features brought by new requirements.

For training the model, we employ the $3 \times 3$ convolution kernels with the stride of 2. And all the convolution layers are followed by batch normalization (BN) and ReLU non-linearity activation except the last convolutional layer. Paddings are added to the layers to make the input and output have the same size. The architecture of D is shown in Table 2 (we do not show BN and ReLU activation for brevity).

E. FULL OBJECTIVE

Overall, our full loss function for G, D can be defined, respectively, as:

$$\mathcal{L}_G = -\mathcal{L}_{GAN-G} + \lambda_{spr}\mathcal{L}_{spr} + \lambda_{cls}\mathcal{L}_{cls-G} + \lambda_{cyc}\mathcal{L}_{cyc}$$

(9)

$$\mathcal{L}_D = \mathcal{L}_{GAN-D} + \lambda_{cls}\mathcal{L}_{cls-D}$$

(10)
TABLE 2. Discriminator network architecture.

| pre-trained part | conv3-1 | conv4-1 |
|------------------|---------|---------|
| 64 x 128 x 128   | 256 x 32 x 32 | 512 x 16 x 16 |

| training part | conv3-2 | conv4-2 |
|--------------|---------|---------|
| 64 x 128 x 128 | 256 x 32 x 32 | 512 x 16 x 16 |
| pool1        | pool2   | pool3   |
| 64 x 128 x 128 | 128 x 32 x 32 | 256 x 16 x 16 |
| pool4        |         |         |
| 64 x 128 x 128 | 128 x 32 x 32 | 512 x 8 x 8 |

TABLE 3. Statistics of CelebA for each mentioned attributes.

|                  | black hair | blond hair | brown hair | male | eyeglasses | overall |
|------------------|------------|------------|------------|------|------------|---------|
| train            | 47966      | 29619      | 41184      | 83583| 13052      | 2060600 |
| test             | 506        | 289        | 388        | 851  | 141        | 2000    |

where $\lambda_{spr}$, $\lambda_{cls}$ and $\lambda_{cyc}$ are constant weight that control the relative importance for each loss function.

IV. EXPERIMENTS

A. DATA COLLECTION

Large-scale CelebFaces Attributes (CelebA) Dataset [21] contains more than 200K facial images of celebrities, each annotated with 40 binary attributes. Since CelebA includes large scale facial poses and complicated backgrounds, it can be easily applied in a realistic context. We cropped the images and resized them to 128 x 128 pixels. The algorithm automatically finishes the cropping process in the pre-processing data stage. We use the centre cropping method to ensure that the cropped image can ultimately preserve the face. We select five classes of annotated attributes in our training model as the attribute labels, including gender, eyeglasses and three kinds of hair colour (black, blond, brown). Hence, there are seven elements as input transfer values (male and female, wear eyeglasses, have no eyeglasses, black hair colour, blond hair colour, and brown hair colour). Moreover, we randomly selected 2000 samples as testing data. The statistics are shown in Table 3.

B. IMPLEMENTATION DETAIL

For all the experiments, we set $\lambda_{spr} = 0.1$, $\lambda_{cls} = 1$, $\lambda_{gp} = 10$, and $\lambda_{cyc} = 10$ in Equation 3. For parameters $\lambda_{cls}$, $\lambda_{gp}$, and $\lambda_{cyc}$, we learned it from other papers [18], [48]. For parameter $\lambda_{spr}$, it’s the result of our experiment. If $\lambda_{spr}$ is too high, there are few left in the residual network. If the value is too low, this loss function will no longer affect. So it is a trade-off to choose this value. We use Adam [49] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ to train the model. We (i) update the generator once after five discriminator updates as in [44], (ii) use the cycle consistency at every generator iteration, and (iii) the sparse constraint for residual images is applied at every generator iterations. The parameters of the proposed model are 64.7 million. Compared with the AttGAN (89.1 million) and the StarGAN (53.2 million), the calculation speed of the proposed model is faster. Hence, a more effective model did not increase the amount of calculation.

The networks are trained with a batch size of 16 for 40000 iterations, which takes around two days. This project is developed under Python 3.6 with Pytorch 1.4.0 framework, and the hardware contains NVIDIA GeForce RTX-1080 Ti GPU, 32GB memory, and Ubuntu 16.04 OS system.

C. PERFORMANCE COMPARISON

1) EXPERIMENT I: SINGLE ATTRIBUTE TRANSLATION

We first compare our proposed method with the StarGAN v2 on a single transfer task. We train the multi facial attributes translation model with five attributes (black hair colour, blond hair colour, brown hair colour, gender and eyeglasses). This experiment is conducted by using the source code given in the paper [19]. Fig.7 shows the facial attribute transfer results of StarGAN v2 and our method on CelebA. We observed that our method better retains other image details such as background, skin colour, and hair colour. That is due to the encoder-decoder process during the generator, which results in StarGAN v2 must lose some points. But our generator aims at generating a residual image with a sparse
constraint loss. Thus our result could avoid lost other details to the most extent.

Moreover, in some different situations (fig.8), we find our method outperforms the current method. Such as when the image has an unusual filter like pink, our method could keep the artistic effect well. Accordingly, applying our method to video translation yields remarkable differences to the video scene. Another situation is when the image already wears a pair of eyeglasses. The StarGAN still put new glasses on it while our method keeps it unchanged. This improvement is feeding the generator with recoding input attribute values, which indicates that the generator should be transferred.

2) EXPERIMENT II: MULTI ATTRIBUTES TRANSLATION
For multi attributes translation, we noticed some inaccuracy. Fig.9 shows some examples. In some cases, when using StarGAN for multi-domain translation, partial attribute transfer may occur (i.e. part of the hair colour translated, half of the eyeglass added). We infer one reason is the structure of the discriminator. When we use a pre-trained pyramid architecture as the discriminator, the result performs much better. We use both trained discriminators to classify facial attributes of 2000 samples randomly divided from CelebA (as described in Section 4.1). The comparison of classification accuracy is shown in Table 4. The result presents that our discriminator performs better.

3) EXPERIMENT III: EVALUATION FOR IDENTITY MAINTAINING
As for facial attribute transfer, we are keeping one identity detail unchanged is one of the most important indicators. This means after the transfer, and we can still identify it. We use the VGG-Face model to evaluate it. We put paired original images and generated images into the model and get the features of the 5th convolution layer. We calculated the mean square error of these features between paired images to get the evaluation value. The lower values mean better identity maintenance. Note that we have not used the pre-trained VGG-Face model in our method during training for this evaluation. The result is shown in Table 5 and Table 6. Almost all the data show that our target image has a more similar face to the original image.

4) EXPERIMENT IV: QUANTITATIVE EVALUATION
For quantitative evaluation, we adopted three evaluation methods to evaluate the quality of the generated image. The Fréchet Inception Distance (FID) [50] can represent...
the diversity and quality of generated images. The smaller the FID, the better the image diversity and quality. Multi-Scale Structural SIMilarity (MS-SSIM) [51] can measure the variety of generated images. The lower the MS-SSIM value, the higher the diversity of generated images. The peak signal-to-noise ratio (PSNR) [52] can evaluate the quality of the generated image compared with the original image. The higher the PSNR, the smaller the distortion of the generated image. Table 7 shows the comparison result of our method and others. The results of the StarGAN [18], the AttGAN [43], the ST-GAN [53], and the FEGAN [54] are utilized for a fair comparison.

5) EXPERIMENT V: ABLATION STUDY
Following the popular evaluation scheme proposed in [55] and [56], we utilized three evaluation indexes to test the proposed model, including AMS, FID, and MMD. Three indexes are measured on 2000 test images from the CelebA dataset. AMS is used to calculate the KL divergence between the training dataset’s class label distribution and the generated dataset’s class label distribution. The lower the AMS score, the better the model’s generation performance. The smaller the value of FID, the closer the two Gaussian distributions are, and the better the performance of GAN. MMD can measure the distance between the training dataset distribution and the generated dataset and then use this distance as the evaluation index of GAN. The smaller the MMD distance, the better the performance of the model. As shown in Table 8, we conduct the following experiments: 1) The attribute encoding process was removed; 2) The residual calculation was removed; 3) Our discriminator was replaced by the one in StarGAN. The results show that each contribution is effective in our methods.

V. CONCLUSION
This paper proposed a novel GAN-based method for multi facial attributes translation in a single model. The proposed method uses the residual image, a transfer vector and a more complicated cycle loss to optimize the generator. And it also uses a discriminator with a pyramid structure. Combining all of them made the multi-attribute translation looks closer to reality and complete. Compared with other StarGAN-based state-of-the-art works, our method performs better in multi attributes translation.

In our algorithm, the L1 loss function is used to constrain the residual image, which can reduce the content of the residual image but does not consider the integrity of the retained features from the perspective of space. This is a crucial direction we need to improve in the future. In the future study, the model optimization scheme will be further improved.

REFERENCES
[1] Z. Wang, Q. She, and T. E. Ward, “Generative adversarial networks in computer vision: A survey and taxonomy,” ACM Comput. Surv., vol. 54, no. 2, pp. 1–38, 2021.
[2] J. Zheng, Y. Wu, W. Song, R. Xu, and F. Liu, “Multi-scale feature channel attention generative adversarial network for face sketch synthesis,” IEEE Access, vol. 8, pp. 146754–146769, 2020.
[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Advances in Neural Information Processing Systems, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
[50] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two-time-scale update rule converge to a local Nash equilibrium,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 1–12.

[51] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in Proc. 37th Asilomar Conf. Signals, Syst. Comput., vol. 2, Nov. 2003, pp. 1398–1402.

[52] A. Hore and D. Ziou, “Image quality metrics: PSNR vs. SSIM,” in Proc. 20th Int. Conf. Pattern Recognit., Aug. 2010, pp. 2366–2369.

[53] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, “ST-GAN: Spatial transformer generative adversarial networks for image compositing,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 9455–9464.

[54] X. Ning, S. Xu, W. Li, and S. Nie, “PEGAN: Flexible and efficient face editing with pre-trained generator,” IEEE Access, vol. 8, pp. 65340–65350, 2020.

[55] Z. Zhou, H. Cai, S. Rong, Y. Song, K. Ren, W. Zhang, Y. Yu, and J. Wang, “Activation maximization generative adversarial nets,” 2017, arXiv:1703.02000. [Online]. Available: http://arxiv.org/abs/1703.02000.

[56] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos, “MMD GAN: Towards deeper understanding of moment matching network,” 2017, arXiv:1705.08584. [Online]. Available: http://arxiv.org/abs/1705.08584.

JUN LIU received the B.E. and M.E. degrees from Shenyang University of Technology, in 1995 and 2000, respectively, and the Ph.D. degree from the Graduate University of Chinese Academy of Sciences and Shenyang Institute of Automation, Chinese Academy of Sciences, in 2010. He is currently an Associate Professor with Shenyang Ligong University. His research interests include intelligent sensors and detection technology, image and signal processing, and intelligent robots.

XIAOYANG LIU received the B.E. degree from Shenyang Institute of Technology, in 2002, and the M.E. degree from Shenyang Ligong University, in 2019. She is currently a Lecturer with Shenyang Ligong University. Her research interests include image processing and object recognition.

YANJUN FENG received the B.E. degree from Liaoning University of Technology, in 1997, and the M.E. degree from Shenyang University of Technology, in 2000. She is currently a Lecturer with Shenyang Ligong University. Her research interests include the IoT technology and intelligent information processing.