Hierarchical neural network for facial attribute transfer

You Guo¹, Yanlei Shang¹,* and Xiaofang Zhang²

¹Institute of network technology, Beijing University of Posts and Telecommunications, Beijing, China
²School of Electrical Engineering, Beijing Jiaotong University, Beijing, China

*Corresponding author e-mail: shangyl@bupt.edu.cn

Abstract. The facial attribute transfer refers to generating a face image with desired attributes while preserving other details. In the existing methods, some of the them consider the independence among attributes but neglect the integrity. It may result in information loss and lead to distorted generation. The others ensure the integrity at the cost of insufficient independence limits. The generated images will be partially blurred. In this paper, GAN and Variational Autoencoders (VAE) structure are incorporated to preserve the details while attribute transfer occurs. The concept of hierarchical latent representation is introduced to realize the attribute independence. These methods work with each other forming an effective network for facial attribute transfer, referred as Hi-GAN. Experiments on the CelebA dataset show that our model can output clearer and more realistic images on facial attribute transfer.

1. Introduction

Pixel2Pixel [1] implements style transfer with paired datasets. However, it is difficult to collect labeled images of a same person with various attribute combinations, so this method is not suitable for facial attribute transfer. Jun-Yan Zhu et al. [2] proposed the cycle structure for style transfer without paired datasets. But one well trained model can handle only one attribute.

By introducing the structure of Variational Autoencoders (VAE) [3], it is possible to realize facial attribute transfer by a single model. VAE/GAN [4] encodes a real face image as a latent representation, and each attribute is defined by the difference between the mean of multiple face latent representations with and without this attribute. The latent representation obtained by encoder is added to specified attribute vectors and input into the decoder, then VAE/GAN outputs a face image with the specified attributes of the same identity. However, this method may cause some unexpected changes of other attributes. In IcGAN [5], the independence among attributes is considered by ensuring that faces with different attributes and the same identity output the approximate latent representation. The encoder and decoder are two independent training processes. The latent representation is sampled from a random normal distribution and input into the decoder. This uncertainty may result in information loss and lead to distorted generation. The face image generated by IcGAN can realize the transfer of specified attributes, but the identity may change simultaneously. AttGAN [6] makes all the structures as a whole during the training process, and achieves the purpose of changing only what you want by introducing the attribute classification loss. However, only trained attributes will be limited by classification loss. When attribute transfer occurs, untrained attributes may also change.
This paper designs Hi-GAN as shown in Figure 1, the contribution lies in three folds:

1) Combine VAE with the structure of GAN to train the network. The result of the encoder instead of random sampling will be input into the generator. The reconstruction loss of image is introduced to keep the information integrity.

2) Introduce the concept of hierarchical latent representation for facial attribute transfer. The latent representation generated by the encoder will be hierarchically input into each layer of the generator. In this way, different parts of the latent representation will control different hierarchical features to realize the attribute independence.

3) A single model is sufficient to realize facial attribute transfer. Compared with other methods, our model can output clearer and more realistic face images.

2. Related work

There are two commonly used generative models. One is the Variational Autoencoders (VAE) [3]. It imposes a prior latent representation space, such as random normal distribution, in order to regularize and constrain the model to sample from it. However, the generated images are usually blurry. The other generative model is Generative Adversarial Net (GAN) [7]. The generator $G$ outputs a fake image with the random noise $z$ sampled from a latent representation distribution $P_z$. The discriminator $D$ is used to distinguish whether the input image is from the real data distribution $P_{data}$. The minimax game can be formulated as:

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

The main problem faced by GAN is its unstable training process and mode collapse. The instability means it is hard to converge. DCGAN [8] uses a full convolutional neural network to keep the stability of the training process. The mode collapse means the generator can only generate a single style or even some meaningless images. The subsequent proposed WGAN [9] solves this problem, it uses Wasserstein distance instead of cross entropy to calculate the loss between the learned distribution $P_g$ and the real data distribution $P_r$. A discriminator without activation function in the last layer can be seen as $f_\omega$. The loss function is calculated as follows:

$$L_g = -E_{x \sim P_g} [f_\omega(x)]$$

$$L_d = E_{x \sim P_g} [f_\omega(x)] - E_{x \sim P_c} [f_\omega(x)]$$

I. Gulrajani et al. [10] introduce the gradient penalty to improve the convergence speed of WGAN. In this paper, the WGAN-GP method is adopted when calculating the adversarial loss.

GAN is unsupervised, it generates an uncontrollable image by a random noise $z$. Conditional GAN (cGAN) [11] can control the label of the generated images by introducing the label information $y$. The optimization objectives are as follows:

$$\min_G \max_D V(D, G) = E_{x \sim P_{data}} \left[ \log D(x|y) \right] + E_{z \sim P_z} \left[ \log \left( 1 - D(G(z|y)) \right) \right]$$
This method has many defects, such as image edge blurring and low image resolution. It has been improved by mapping the generated images back to the conditional information recently.

3. Hi-GAN

As shown in Figure 1, Hi-GAN is composed of a generator, a discriminator, an encoder and an attribute classifier. A real image $A^a = [a_1, a_2, ..., a_n, x_{n+1}, x_{n+2}, ..., x_{n+m}]$ will be input into the encoder. The lowercase letter $a = [a_1, a_2, ..., a_n]$ represents attribute binary vector. The uppercase letter $A = [x_{n+1}, x_{n+2}, ..., x_{n+m}]$ can be regarded as the identity information of the face image. The output of the encoder and a specified attribute information $b = [b_1, b_2, ..., b_n]$ will be put into the generator $G$ to generate a face image $A^b$ with the specified attributes.

3.1 Forward propagation

A real image $A^a$ will be encoded into a latent representation, which is expressed as follows:

$$ Z = Ez(A^a) \tag{5} $$

The vector $Z$ will be divided into several parts, and respectively sent to the different layers of the generator. A transferred image $A^b$ and a reconstructed image $\bar{A}^a$ will be generated:

$$ A^b = G(Z, b) \tag{6} $$
$$ \bar{A}^a = G(Z, a) \tag{7} $$

3.2 Formulation

**Adversarial Loss.** The adversarial loss ensures that the generated face image can be more realistic. $A^a$ is sampled from the real data distribution $P_{data}$ and $b$ is sampled from attribute distribution $P_{attr}$. WGAN-GP [10] is used to optimize the adversarial loss:

$$ \min_D L_{adv_d} = -E_{A^a \sim P_{data}} D(A^a) + E_{A^a, b \sim P_{data}, b \sim P_{attr}} D(A^b) \tag{8} $$

$$ \min_{Ez, G} L_{adv_g} = -E_{A^a, b \sim P_{data}, b \sim P_{attr}} D(A^b) \tag{9} $$

**Attribute Classification Loss.** With the summation of binary cross entropy losses of all attributes, the classification loss $\ell_c$ and $\ell_g$ respectively optimize the classifier $C$ and the generator $G$.

Generator can be limited to change only the specified attributes, denoted as:

$$ \min_c L_{cls_c} = E_{A^a \sim P_{data}} \ell_c(A^a, a) \tag{10} $$

$$ \min_{Ez, G} L_{cls_g} = E_{A^a \sim P_{data}, b \sim P_{attr}} \ell_g(A^a, b) \tag{11} $$

**Reconstruction Loss of Image.** We use the reconstruction loss of image to minimize the information loss. In other words, the real image $A^a$ can perfectly be restored by the model:

$$ \min_{Ez, G} L_{rec,img} = E_{A^a \sim P_{data}} \| A^a - \bar{A}^a \|_1 \tag{12} $$

**Objective Function.** The final loss function is divided into the three parts. According to above Equation, the optimization goals can be expressed as:

$$ \min_{Ez, G} L_g = L_{adv_g} + \lambda_1 L_{cls_g} + \lambda_2 L_{rec,img} \tag{13} $$

$$ \min_{D, C} L_d = L_{adv_d} + L_{cls_c} \tag{14} $$

The hyperparameters $\lambda$ represent the weight of each loss functions to balance the value of loss.

| Operation         | Kernel | Stride | Filters | BN / IN | Activation |
|-------------------|--------|--------|---------|---------|------------|
| Split & Fully connected | 5x5    | 2      | 1024    | None    | Relu       |
| Concat & Deconv   | 5x5    | 2      | 1024    | None    | Relu       |
| Concat & Deconv   | 5x5    | 2      | 512     | BN      | Relu       |
| Concat & Deconv   | 5x5    | 2      | 256     | BN      | Relu       |
| Concat & Deconv   | 5x5    | 2      | 128     | BN      | Relu       |
| Deconv            | 5x5    | 2      | 3       | BN      | Tanh       |

**Table 1.** Detailed generator architecture.
3.3 Implementation

**Network Architecture.** Table 1 shows the detailed generator architecture of our model for 128×128 images. The encoder architecture is a mirror of the generator without the Concat operation. Kernels with size of 5×5 and stride 2 are used in the whole network. The discriminator $D$ and the classifier $C$ share convolutional layers, and use IN (Instance Normalization) instead of BN (Batch Normalization). The encoder $Ez$ uses Leaky ReLU as the activation function while the generator $G$ uses ReLU. The latent representation $Z$ is divided into four parts. Each part and the specified attributes will be sent to different layers of the generator. The generator finally maps the output to [-1, +1] with Tanh function.

**Training details.** We train the model with the Adam optimizer ($\beta_1 = 0.5, \beta_2 = 0.999$). The generator and the encoder are trained once after the discriminator and classifier are trained 5 times. The hyperparameters in the final objective loss function are set as: $\lambda_1 = 10, \lambda_2 = 100$. Batch size is set to 32 or 16, and the initial value of the learning rate is 0.0002.

4. Experiment

CelebFaces Attribute Dataset (CelebA) is a large-scale face attributes dataset containing 202,599 face colored images of 10,177 celebrities and 40 attribute binary vectors.

Several of the original 40 attributes with strong visual impact are chosen for the experiment. We preprocess the face images by cropping the images to 128×128, scaling the pixel value to [-1, +1] and so on. Finally, we evaluate the quality of generated images in single facial attribute

![Figure 2](image2.jpg)

*Figure 2.* The result of applying Hi-GAN to face images for single attribute inversion.

![Figure 3](image3.jpg)

*Figure 3.* Comparisons with AttGAN and StarGAN on inversing specified attributes.

![Figure 4](image4.jpg)

*Figure 4.* The result of applying Hi-GAN to face images for multiple attributes inversion.

and multi-attribute inversion. We mainly compare our model with AttGAN [6] and StarGAN [12] which are all trained by their official code.
Single facial attribute transfer. Figure 2 shows the performance of our model in single facial attribute inversion. We also compare the result with AttGAN and StarGAN in Figure 3. The results of StarGAN contain some artifacts while the results of our work and AttGAN look more natural and realistic. When some attributes are inversed, it can be found in Figure 3 that AttGAN caused unexpected changes in the images than just the specified attributes. Specifically, if we turn the gender attribute of a short-haired male image into a female, short hair will also become long hair in AttGAN. The reason for the above problem is that AttGAN does not consider attribute independence. There are some logical correlation between attributes. For example, in the dataset, most of the males are short-haired while most of the females are long-haired, so the hair characteristics usually change when the gender attribute changes. In this paper, the concept of hierarchical latent representation is introduced. Different parts of the latent representation will control the different hierarchical features on the generated image, thus eliminating the possible correlation among the attributes.

Multi-attribute transfer. As shown in Figure 4, a face image is generated by inversing several specified attributes in one time. There may be some more abstract relationships among these attributes, so it has higher requirements for attribute independence. It can be seen from the results of AttGAN, if there is a high correlation among these simultaneously changed attributes, the generated images will be partially blurry. Because the pixel values generated in some regions will be affected by multiple attributes. Visual blurring will be caused as the result of these mutual interferences. Hi-GAN limits this correlation, so when we inverse multiple attributes at the same time, the generated face image has better visual performance.

5. Conclusion
This paper mainly proposes an effective network for facial attribute transfer. It realizes the attribute independence while maintaining the integrity of other information. The hierarchical latent representation is respectively input into each layer of the generator to realize the attribute independence. Combining VAE with GAN structure, the network can restore the information of the input image as much as possible. Experiments on the CelebA dataset show that our design has achieved good results. Compared with other existing methods, Hi-GAN can output clearer and more realistic face images.

6. Acknowledgments
The work presented in this study is supported by the National Natural Science Foundation of China "Research on Web-based Dynamic Service Delivery Mechanism and Method in Information Center Network"(61671081) and the Natural Science Foundation of Beijing “Research on Dynamic Service Provision Mechanism of Information Center Network Based on Edge Computing” (4172042).

References
[1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” in CVPR, 2017.
[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” in ICCV, 2017.
[3] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in ICLR, 2014.
[4] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, “Autoencoding beyond pixels using a learned similarity metric,” in ICML, 2016.
[5] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Alvarez, “Invertible conditional gans for image editing,” in Advances in NIPS Workshops, 2016.
[6] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen, “AttGAN: Facial Attribute Editing by Only Changing What You Want,” in TIP, 2019.
[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S.Ozair, A.Courville, and Y.Bengio, “Generative adversarial networks,” in NIPS, 2014.
[8] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in ICLR, 2016.
[9] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein gan,” in ICML, 2017.
[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of wasserstein gans,” in NIPS, 2017.

[11] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” arXiv preprint arXiv:1411.1784, 2014.

[12] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” in CVPR, 2018.