Generating Face Images with Attributes for Free

Yaoyao Liu, Qianru Sun, Xiangnan He, An-An Liu*, Yuting Su*, and Tat-Seng Chua

***Abstract*—With super human-level performance of face recognition, we are more concerned about the recognition of fine-grained attributes, such as emotion, age, and gender. However, given that the label space is extremely large and follows a long-tail distribution, it is quite expensive to collect sufficient samples for fine-grained attributes. This results in imbalanced training samples and inferior attribute recognition models. To this end, we propose the use of arbitrary attribute combinations, without human effort, to synthesize face images. In particular, to bridge the semantic gap between high-level attribute label space and low-level face image, we propose a novel neural-network-based approach that maps the target attribute labels to an embedding vector, which can be fed into a pretrained image decoder to synthesize a new face image. Furthermore, to regularize the attribute for image synthesis, we propose to use a perceptual loss to make the new image explicitly faithful to target attributes. Experimental results show that (1) our approach can generate photo-realistic face images from attribute labels, and (2) more importantly, by serving as augmented training samples, these images can significantly boost the performance of attribute recognition model. The code is open-sourced at this link.**

***Index Terms*—Face attribute recognition, image generation, data augmentation, learning systems, pattern recognition**

I. Introduction

THE recognition of facial attributes, including gender, age, and emotions, as well as wearable accessories, is beneficial for fine-grained face applications, such as face verification [1], reidentification [2], and retrieval [3]–[5]. However, the collection of training data for fine-grained facial attributes of a combinatorially large label space is expensive, leading to a long-tail data distribution. In Figure 1, we show the image statistics of a specific tri-attribute composition from a large-scale face dataset CelebA [6]. The composition of *Attractive + No Beard + Young* has $71k$ samples, while that of *Mouth Slightly Open + Oval Face + Receding Hairline* has only $2k$ samples, which is 35 times fewer than the former.

For an imbalanced dataset, there is a huge performance gap between attribute models trained with large data and those trained with little data, resulting in a non-robust face recognition [7]–[14]. On the other hand, fine-grained facial attributes usually include both global and local categories, e.g., gender and age are global, while hairstyles and wearing glasses are local. Meanwhile, in the act of facial attribute recognition, experts extract features from both local and global face images before intergrating them for the subsequent psot-processing. Therefore, it is somewhat inevitable for attribute models with rare training data to avoid biases agaunst attributes with frequent training data. For example, when the *wearing sunglasses* samples are all woman faces, then the recognition of this attribute is seriously biased against the female attribute.

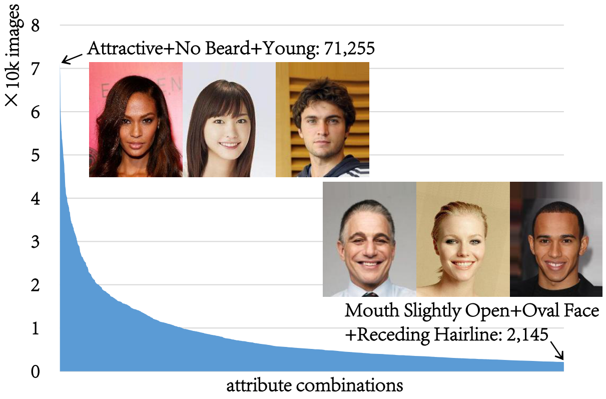


Figure 1. Statistics of image quantity with specific tri-attribute compositions in the CelebA dataset [6]. There are over $9k$ combinations in total, of which $2k$ items are shown for a better visualization. Examples of such combinations are given toward the end.

Traditional data augmentation techniques, such as up-sampling [15], image flipping [16], and random cropping [17], on minority classes have limited visual diversity since they cannot yield any novel visual cues. An alternative is to assign higher misclassification costs to compensate for those of the minority classes [18]. However, this does not resolve the inherent data imbalance in principle, but only offers an expedient solution. In this paper, we tackle the issue of synthesizing face images with the attributes on demand. In particular, we compensate for more rare classes data without any human effort on annotation. Besides generation from Gaussian noises [19], we use the 0/1 attribute labels as input. To solve any gap problem that may arise between the discrete label space and realistic attribute distribution, we propose a novel approach that utilizes the pretrained image reconstruction auto-encoder as a strong guidance. The mapping function from attribute labels to the embedding feature is determined by both the output feature of a pretrained encoder and the output image of a pretrained decoder. Then, we add a pretrained attribute recognizer to penalize the decoded images of wrong attributes, thereby making the whole neural-network-based generation framework sensible to attribute embedding.

During the attribute-to-image sampling, we feed the at-

Yaoyao Liu, An-An Liu, and Yuting Su are with the School of Electrical and Information Engineering, Tianjin University. Yaoyao Liu is secondly with Max Planck Institute for Informatics, Saarland Informatics Campus. Qianru Sun is with the School of Information Systems, Singapore Management University. **This work was done during Yaoyao's internship supervised by Qianru.** Xiangnan He is with the School of Information Science and Technology, University of Science and Technology of China. Tat-Seng Chua is with the School of Computing, National University of Singapore. Corresponding authors are An-An Liu and Yuting Su whose emails are {liuanan, ytsu}@tju.edu.cn.

This research is supported in part by the National Natural Science Foundation of China (61772359, 61572356), the grant of Tianjin New Generation Artificial Intelligence Major Program (19ZXZNGX00110, 18ZXZNGX00150), the Open Project Program of the State Key Lab of CAD and CG, Zhejiang University (Grant No. A2005, A2012). It is also part of NExT++ research which is supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@SG Funding Initiative, and the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant.

tribute labels for synthesizing novel face images. In the experimental section, we show that the generated images are sensible enough to contain the input attributes. For data augmentation, we mainly solve two problems: data lacking and data balancing, for which we propose random doubling and dynamic balancing strategies, and generate face images accordingly.

Our key contributions are three-fold: (1) we design a novel neural-network-based framework for generating images from attribute labels, (2) we develop an efficient data augmentation approach using the generated images, and (3) experimentally validate the generation of high-quality images and that our augmentation approach outperforms the state-of-the-art method for a large-scale face image dataset.

The rest of this paper is organized as follows. In Section II, we present related studies mainly on fine-grained attribute recognition, data augmentation, and face image generation. In Section III, we describe the proposed neural-network-based generation framework in detail. In Section IV, we analyze the class distribution of a large-scale face dataset and propose two data augmentation strategies. Finally, in Section V, we evaluate the proposed approach by comparing it with both the baselines and the state-of-the-art methods.

II. Related Work

A. Fine-grained facial attribute recognition

Super human-level performances include tradition recognition, face detection, and blooming of convlutional neural networks (CNNs) [20] Fine-grained attribute recognition and applications have been drawing increasing attention. Berg *et al.* [1] proposed the Part-Based One-vs-One Features (POOFs) for using fine-grained regional features for face verification. Manyam *et al.* [2] built a model based on conditional and joint probabilities for face reidentification. Yang *et al.* [3] used a deep CNN called Faceness-Net, for face detection based on facial attributes. Liu *et al.* [6] proposed a model for localizing human faces and recognizing attributes in wild face images using a combination of three CNNs. They built two facial attribute datasets, namely CelebA and LFWA, based on CelebFaces [21] and LFW [22]. CNN-based methods can predict many attributes together. However, the aforementioned methods perform better on balanced datasets, and may prefer to predict the attributes with more samples when trained on an imbalanced dataset.

B. Data augmentation

Although the deep model has achieved great success in image recognition tasks, it still suffers from data sparsity and imbalance problems. Therefore, data augmentation is high recommended. Traditional augmentation methods artificially enlarge the training dataset using basic translations, e.g., random flipping and random cropping [17], on the existing data. Most recently, random erasing on original images [23] has yielded consistent improvement on multiple tasks, such as image classification, object detection and person reidentification. However, these methods cannot augment novel data. Thus, the present paper proposes the synthesis of novel images for rare classes in a controllable manner. A similar idea has been applied to the augmentation task of Action Unit (AU) intensity estimation [24]. Specifically, the paper [24] proposed an AU synthesis framework that uses adversarial learning to generate expression parameters from AU labels using a 3D morphable model.

Some other authors have attempted to indirectly solve the data imbalance problem. To share the positive class information in each subclass, Shen *et al.* [18] designed the positive-sharing loss. Rudd *et al.* [25] introduced a novel mixed objective optimization network (MOON), and designed a loss function that merged multiple task objectives with domain-adaptive reweighting on propagated losses. The balanced network outperformed the unbalanced one. Instead of designing new algorithms, our approach to synthesizing new images fundamentally solves the data lacking problem.

C. Face image generation and manipulation

For face image generation and attribute manipulationn [26]–[38], Generative Adversarial Networks (GANs) [39] show remarkable performance when tasked with image generation. In a GAN model, a generator is designed to produce *fake* images with a distribution similar to that for the *real* ones, while the discriminator learns to distinguish between fake and real images. In general, GANs are notoriously difficult to train and often suffer from model collapse [40]. In our study, we solve these problems by proposing a novel framework that leverages pretrained models as learning guidance.

The boundary equilibrium generative adversarial networks (BEGANs) [40] employed an auto-encoder as a discriminator for calculate thing Wasserstein distances as adversarial losses. It proposes new, more stable equilibrium strategies that hasten convergence compared to the original GANs. BEGANs are employed in our method to train an auto-encoder whose components serve as pretrained parts in the framework. For a conditional generation task, deep convolutional GANs (DCGANs) [19] applied some constraints to make traditional GANs more stable for training. In our task, the condition is the facial attribute, and thus the discriminator of the DCGAN is utilized in our main generation architecture.

There has been significant progress in the field of attribute transferring and manipulation. Choi *et al.* proposed the StarGAN [26] and achieved the state-of-the-art image-to-image translation performance for multiple domains by a single model. However, this model treats multiple attributes as additional input channels, which makes the whole model suffering from heavy computations. In our method, we map the attribute labels to an embedding space to solve this problem. Other models such as IcGAN [41] and AttGAN [28] manipulated the face attributes by concatenating logic vectors (changing factors) with original image embedding features. However, directly concatenation may destroy the continuity of the image embedding space. Wan *et al.* [42] presented a multi-attribute GAN (FM-GAN), which was able to generate face images with specific attributes, e.g. a 30-year-old white man. However, their experiments were focused on the generation of three main human attributes as Age, Gender, and Ethnicities, but our work pays more attention to a more fine-grained level, e.g. 40 classes on the CelebA dataset.

Sun *et al.* [16] proposed to generate head images guided by predicted facial landmarks. They found that the quality of the generated images could be improved by using the pre-trained decoder of a landmark reconstruction network. In this paper, we adopt a similar idea while we also use the pre-trained encoder to guide the reconstruction of embedding features.

III. Framework of face image generation

This section presents our framework for generating face images from the target attribute labels. The task is challenging due to the semantic gap between the high-level attribute label space and low-level face images. We propose a novel generation framework that first maps the 0/1 labels to an embedding vector using a pretrained encoder, and then feeds the vector into a pretrained image decoder to synthesize a new image. In addition, we use an attribute perceptual loss to make the new image explicitly faithful to the target attributes.

A. Pretraining

In Figure 2, we introduce the framework of guidance networks at the pretraining stage.

Encoder **G_{en} and *Decoder* G_{de}.** The best quality of image generation is obtained from the strongest conditional information, *i.e.*, the original image. This paper proposes the use of the image reconstruction method, which is a tyical architecture for data reconstruction [16], [43], to train an auto-encoder comprising and *encoder* and a *decoder*. To make the reconstructed image sharp and real, we adopt an adversarial learning [44] using the discriminator architecture of BEGAN [40].

Figure 2(a) illustrates the global framework, which includes the auto-encoder components and the adversarial loss from the BEGAN discriminator. The original image x_0 goes to *Encoder* G_{en} and is compressed into an embedding vector z_0. Then, z_0 is decoded to $\hat{x}_0$, which is optimized towards the original image x_0, *i.e.*,

$$G_{en}(x_0) = z_0, \tag{1}$$

$$G_{de}(z_0) = \hat{x}_0. \tag{2}$$

According to [16], [44], the L1 loss between $\hat{x}_0$ and x_0 is the image reconstruction loss. The traditional auto-encoder framework has the problem of blurriness in image reconstruction [44], so we adopt the image adversarial loss to make the image clearer. The optimization function $\mathcal{L}_G$ for G_{en} and G_{de} is defined as follows,

$$\mathcal{L}_G = -k_t\mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{rec}}, \tag{3}$$

where $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{rec}}$ indicate the adversarial loss and reconstruction loss, respectively, and are calculated as follows:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{x_0,\hat{x}_0}[||x_0 - \hat{x}_0||_1], \tag{4}$$

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{z_0}[\log D(z_0)] + \mathbb{E}_{\hat{z}_0}[\log(1 - D(\hat{z}_0))]. \tag{5}$$

The proportional control theory is applied as in [40] to maintain the equilibrium $\mathbb{E}(\mathcal{L}_{\text{adv}}) = \gamma\mathbb{E}(\mathcal{L}_{\text{rec}})$. For the k-th training step, the variable $k_t \in [0, 1]$ is used to control how much emphasis is put on $\mathcal{L}_{\text{adv}}$ during the gradient descent. k_0 is initialized as 0 and its recurrence relation is given as:

$$k_{t+1} = k_t + \lambda_k(\gamma\mathcal{L}_{\text{rec}} - \mathcal{L}_{\text{adv}}), \tag{6}$$

where λ_k is the learning rate for the k-th training step and γ is a hyperparameter, which is set to 0.5. The optimization function of the discriminator is $\mathcal{L}_{\text{adv}}$, which is optimized along with G_{en} and G_{de}.

Classifier **C.** The attribute *Classifier* C is trained on the original images x_0 and labels a_0. The objective is to recognize the facial attributes a_0, *i.e.*

$$C(x_0) = \hat{a}_0. \tag{7}$$

The optimization function is defined as,

$$\mathcal{L}_C = \mathbb{E}_{x_0,a_0}[-\log C(a_0|x_0)]. \tag{8}$$

Later in this paper, we will show that the above pretrained models lead to easier and faster conditional generation (on attribute labels) compared to the baseline method trained from scratch. Moreover, this method can applied to training other conditional generation frameworks that are difficult to train from scratch.

B. Generation with attributes

This section presents our generation framework that is conditioned on the target attribute labels. As shown in Figure 3, pretrained components (the gray background) are fixed to serve as a learning guide for the *Mapping function*. The learning process is optimized from the following three aspects: (1) the embedding vector from attribute labels updates toward the image embedding feature compressed by the *Encoder*; (2) the generated image from the *Decoder* updates towards the original image; and (3) the predicted labels on the generated images update towards the given attribute labels.

Attribute label input. Following the traditional multi-label settings, each attribute is represented by 1/0 code for with/without it. Therefore, the input attribute labels are represented by a 1/0 sequence, denoted by "a".

Attribute *Mapping function* Φ. In our framework, we study six deep network components: *Mapping function*, *Encoder*, *Decoder*, attribute *Classifier*, convolutional layer discriminator, and fully-connected layer discriminator[1]. When fixing the pretrained *Encoder*, *Decoder* and *Classifier*[2], the model aims to learn the *Mapping function* Φ that embeds the attribute labels a to features $\hat{z}$, i.e.,

$$\Phi(a) = \hat{z}. \tag{9}$$

Φ is realized by a fully-connected neural network (see Section III-D for architectural details). Thus we assume that the image embedding feature z contains the key information for

[1] Discriminators denotations are not explicitly given in Figure 3. The adversarial loss between images is due to convolutional layer discriminator and that between features results from the fully-connected layer discriminator. Architecture details are given in Section III-D.

[2] Discriminators also get optimized during model training while their function is only refined due to the dominating reconstruction of L1/L2 losses.

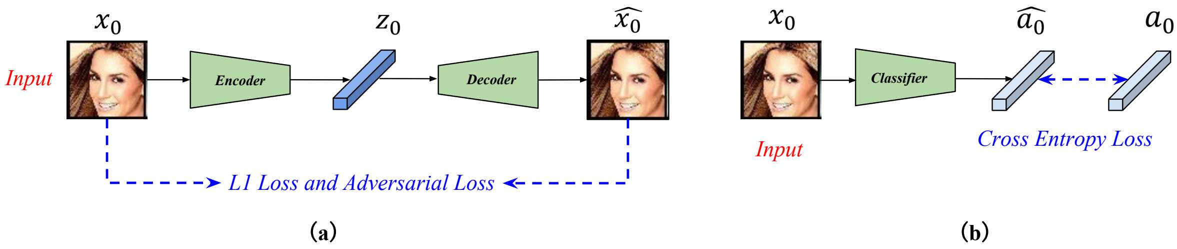


Figure 2. (a) Framework of image reconstruction auto-encoder. An *Encoder* G_{en} encodes an original image x_0 to the embedding feature z_0, and then a *Decoder* G_{de} learns to reconstruct the image $\hat{x}_0$ from z_0. L1 loss and adversarial loss are applied to optimize the framework. (b) The framework of attribute *Classifier* C.

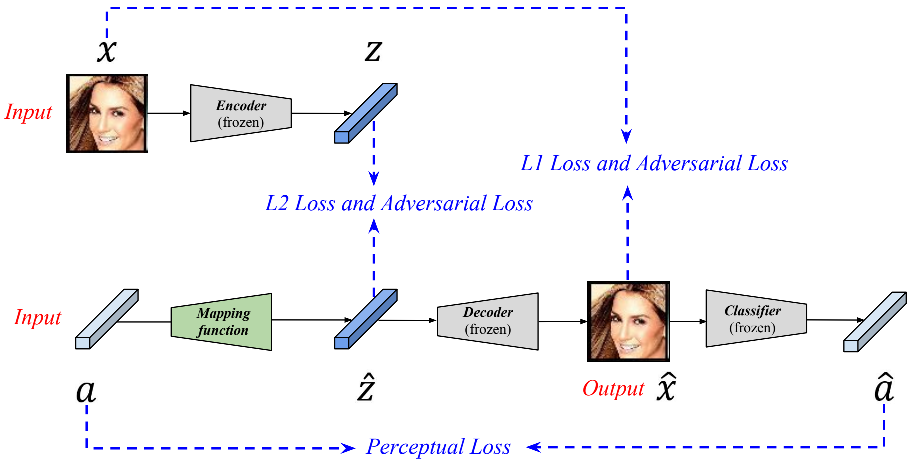


Figure 3. Overview of our training framework for image generation, comprising six main parts: a *Mapping function* Φ, an *Encoder* G_{en}, a *Decoder* G_{de}, an attribute *Classifier* C, and two discriminators. Note that discriminators are implicitly shown by adversarial losses instead. The components in gray are pretrained as in Figure 2, while the *Mapping function* Φ is optimized with reference to this framework. During testing, we feed forward the target attribute labels a through the lower branch to generate an image $\hat{x}$ as the final output.

reconstructing the image x. If $\hat{z}$ can be approximated to z, then the generation result $\hat{x}$ from $\hat{z}$ will be approximated to x through the same *Decoder*.

The optimization of Φ depends on the global back-propagation from all network components and it has five losses in total, including the perceptual loss introduced in the following paragraph. Full optimization functions are discussed in Section III-C.

Attribute perceptual loss. The result of image decoding $\hat{x}$ is expected to embed the target attributes. Here, we use the attribute perceptual loss [45] explicitly to regularize the attribute-embedding process. The attribute recognition model is pretrained, as introduced in Section III-A, and then it is loaded in the generation framework to compute the perceptual loss.

Testing phase. The testing of image generation is a sub-branch of the training phase. Given an arbitrary attribute label sequence a, we generate a face image $\hat{x}$ by feeding a through the *Mapping function* Φ and the *Decoder* G_{de}.

C. Optimization strategy

We introduce the optimization of our framework. As shown in Figure 3, we deploy five losses.

Image reconstruction loss. We use the L1 loss to dominate the recovery of the image pixels towards the ground-truth image x.

$$\mathcal{L}_{\text{rec}_{\text{img}}} = \mathbb{E}_{x,\hat{x}}[||x - \hat{x}||_1]. \tag{10}$$

Feature reconstruction loss. To ensure that the attribute embedding feature $\hat{z}$ to preserves the content of the image embedding feature z, we apply the L2 reconstruction loss defined as:

$$\mathcal{L}_{\text{rec}_{\text{emb}}} = \mathbb{E}_{z,\hat{z}}[||z - \hat{z}||_2^2]. \tag{11}$$

Convolution layer adversarial loss. We regard the generated image $\hat{x}$ as fake and the ground-truth image x as real. To distinguish $\hat{x}$ from x, we use the image adversarial loss defined as follows:

$$\mathcal{L}_{\text{adv}_{\text{conv}}} = \mathbb{E}_x[\log D_{conv}(x)] + \mathbb{E}_{\hat{x},a}[\log(1 - D_{conv}(\hat{x}|a))]. \tag{12}$$

Fully-connected layer adversarial loss. We regard the attribute embedding feature $\hat{z}$ as fake and the image embedding feature z as real. Then, we define the feature adversarial loss as:

$$\mathcal{L}_{\text{adv}_{\text{fc}}} = \mathbb{E}_z[\log D_{fc}(z)] + \mathbb{E}_{a,\hat{z}}[\log(1 - D_{fc}(\hat{z}|a))]. \tag{13}$$

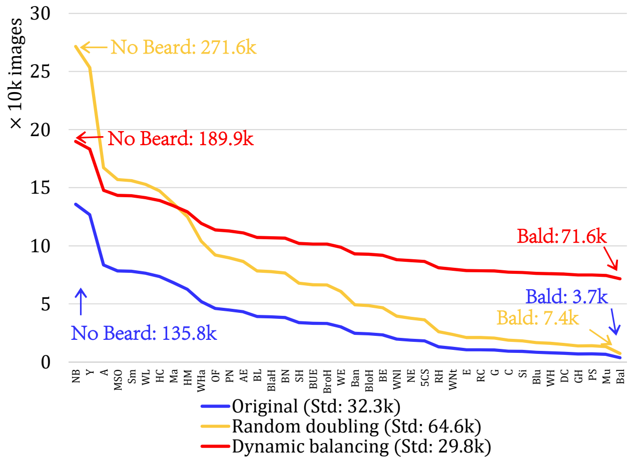


Figure 4. Dataset bias. This figure shows the sample numbers of the attribute labels. Attribute names are abbreviated.

Perceptual loss. The attribute *Classifier* training is a multi-label classification problem. We use the cross-entropy loss as follows:

$$\mathcal{L}_p = \mathbb{E}_{\hat{x},a}[-\log C(a|\hat{x})], \tag{14}$$

For training the generation framework, this loss is called attribute perceptual loss.

Full objective. The objective functions for optimizing D_{fc}, D_{conv} and Φ are, respectively, given as

$$\mathcal{L}_{D_{fc}} = -\mathcal{L}_{\text{adv}_{\text{fc}}}, \tag{15}$$

$$\mathcal{L}_{D_{conv}} = -\mathcal{L}_{\text{adv}_{\text{conv}}}, \tag{16}$$

$$\mathcal{L}_\Phi = \mathcal{L}_{\text{adv}_{\text{fc}}} + \lambda_1 \mathcal{L}_{\text{adv}_{\text{conv}}} + \lambda_2 \mathcal{L}_{\text{rec}_{\text{img}}} + \lambda_3 \mathcal{L}_{\text{rec}_{\text{emb}}} + \lambda_4 \mathcal{L}_p, \tag{17}$$

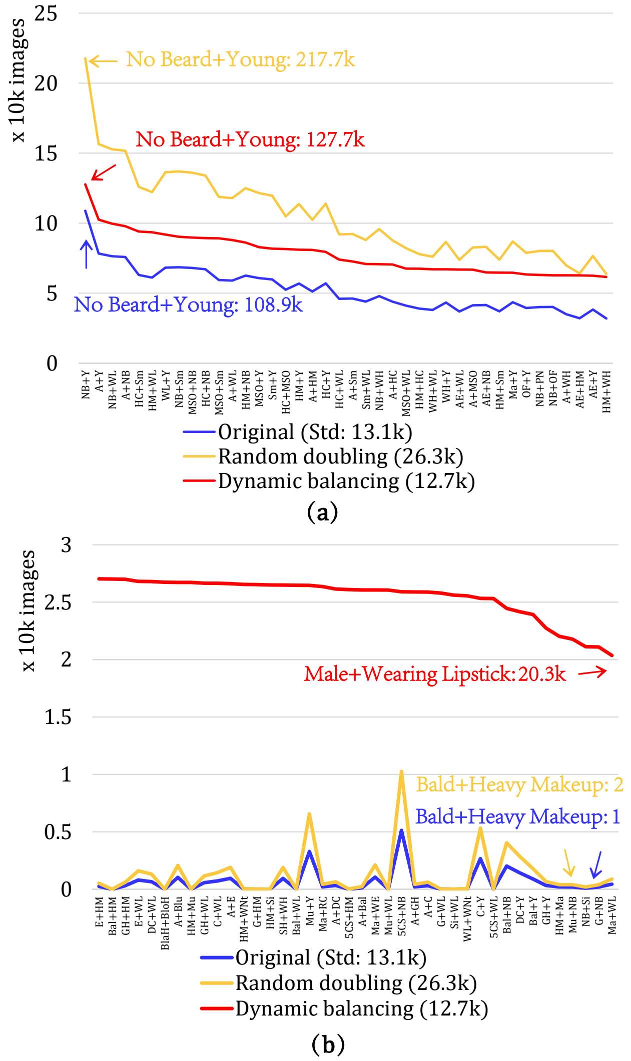
where λ_1, λ_2, λ_3 and λ_4 are the weight parameters for balancing different terms.

D. Network architecture

Here, we give details of our network architecture.

***Mapping function*.** The *Mapping function* Φ comprises six fully-connected layers. The input dimension of the first layer, *i.e.*, the number of facial attribute labels, is 40. The output dimensions of the layers are respectively 128, 128, 256, 256, 128, and 64. The activation functions are exponential linear units [46].

***Encoder* and *Decoder*.** The *Encoder* G_{en} and *Decoder* G_{de} follow the framework of BEGAN [40]. We use 3×3 convolution kernels with exponential linear units [46]. Each layer is repeated for four times. The convolution filters are increased linearly with each down-sampling. Down-sampling is performed as a subsampling with stride 2, and convolution filters are increased linearly with each down-sampling. In the event of up-sampling, we apply the nearest neighbor. we apply nearest neighbor for up-sampling. We use a fully-connected layer without any activation functions to map the data tensor between the *Encoder* and the *Decoder*. The final output of the fully-connected layers is the embedding feature z. Additionally, residual blocks are employed in both convolutional and fully-connected layers, which has been proven to be effective for image generation [16], [44].



(a)

(b)

Figure 5. Number of images with specific double-attribute compositions. Note that there are hundreds of double-attribute compositions. For clear visualization, we only present the top (a) and bottom (b) 40 classes. Attribute names are abbreviated.

Fully-connected layer discriminator. In the fully-connected layer discriminator D_{fc}, we use four fully-connected layers with output dimensions: 512, 512, 512, and 1. We apply leaky rectified linear units [47] to each layer, except for the output layer.

Convolutional layer discriminator. In the convolutional layer discriminator D_{conv}, we use five convolutional layers. The kernel size of the filters is 4 and the stride is 2. The number of filters increases exponentially with each block. The output numbers increases from 64 to 1024. We apply rectified linear units (ReLUs) [48] to each layer, except for the output layer.

Attribute *Classifier*. The architecture of the *Classifier* has the same convolutional layers as that possessed by our convolutional layer discriminator. However, there are two differences: (1) the final output layer has 40 dimensions and (2) the activation function is leaky-rectified linear units [47].

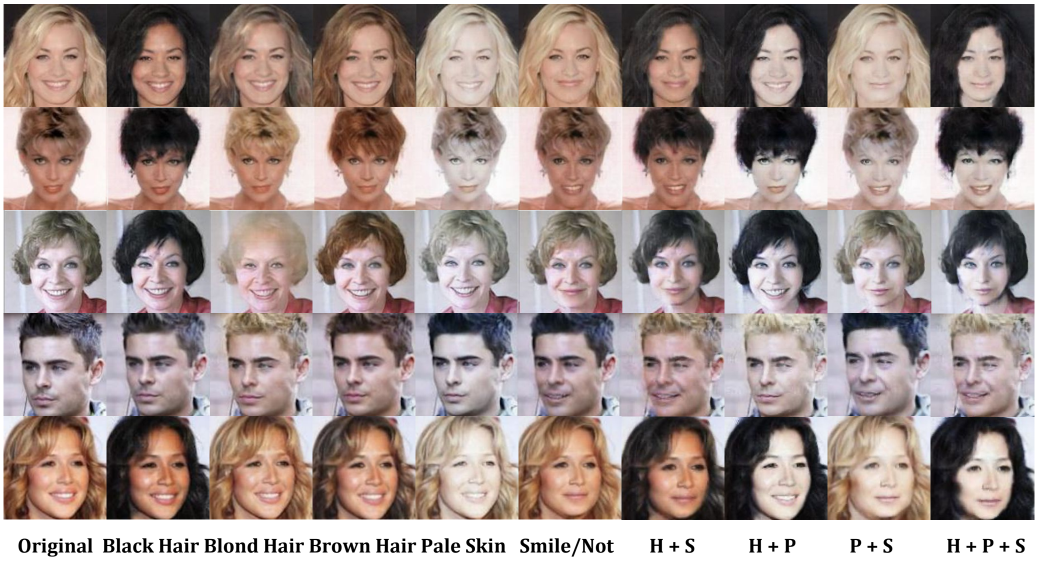


Figure 6. Visualization of manipulation of facial attributes. In particular, *Simle/Not* indicates whether the original image is smiling or not. We exchange the smiling with not smiling, and vice versa. The rightmost columns show the multi-attribute transfer results. H: *Hair color*, P: *Pale skin*, S: *Smile/Not*.

IV. Data augmentation strategy

Given arbitrary attribute labels, our approach can generate face images accordingly. Using these images as augmented data, we propose a random doubling strategy and a dynamic balancing strategy to tackle data sparsity and data imbalance, respectively.

A. Random doubling

Considering the training image, we randomly modify one of its 40 attribute labels of the training image from 0 to 1 or from 1 to 0. In the process of going through all training images, we *exactly* double the sample number of the dataset, but *almost* double the sample number for each attribute due to the randomness. This is a simple way to augment multi-labeled data with a minor change in the original data distribution. However, using this strategy, the imbalance problem still exists.

B. Dynamic balancing

An ideal approach to solve data imbalance is to provide samples for minority classes. In our task, we have multiple labels in each face image, e.g., 40 attributes in CelebA dataset. It is quite challenge to augment data independently for a single class. For example, if we add a sample of a man, then we should add a sample of each of his attributes. To resolve this problem, we propose a dynamic data balancing strategy. When composing the input attribute labels for generation, we restrict these attributes with several samples. Then, we can witness an increase in the probability of minority attribute classes.

The implementation steps of this approach are as follows. (1) Calculate the sample proportions for all attributes, and then rank them. (2) Pick up a training image with its attribute labels a. According to the rank, we set the top 8 attributes to a *lower priority* for augmentation and the bottom 8 to a *higher priority*. (3) Modify the attribute labels a to a_1 by setting its *higher priority* attributes to 1 and its *lower priority* attributes to 0. (4) Feed a_1 through the *Mapping function* Φ and *Decoder* G_{de} to generate a face image $\hat{x_1}$. (5) Assign the target attribute labels a_1 to $\hat{x_1}$ and put this new sample into the augmented dataset. (6) Re-execute the above steps(1)-(5) until the total number of samples is augmented to double.

C. Data distribution analysis

Figure 4 and Figure 5 show the data statistics for single attributes and double-attribute compositions, respectively. Attribute names are replaced with abbreviations[3]. Note that there are over hundreds of double-attribute compositions but we only show the top and bottom 40 classes in Figure 5(a) and (b) for clear visualization.

From the blur curve in Figure 4, we observe that the original CelebA classes have a long-tail distribution problem that cannot be solved by random sampling (yellow curve) but can be largely reduced by dynamic balancing. Specifically, compared to the random doubling, dynamic balancing increases the minimum sample number by $64.2k$, decreases the maximum number by $62.7k$, and drops the standard deviation from $64.9k$ to $29.8k$.

In Figure 5(a), we observe that the use of dynamic balancing decreases the amounts of top attribute compositions to the average of the original and randomly doubled. As shown in Figure 5(b), the balancing impact tends to be more significant among rare classes; for example, the standard deviation gets over halved.

[3] Attribute names are replaced with abbreviations. For example, the attribute *Attractive* is abbreviated to *A*. All abbreviations are given in Section V-A.

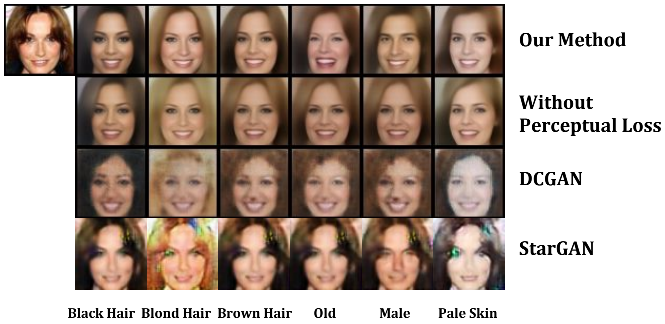


Figure 7. Comparisons of facial attribute editing results among our methods, DCGAN and StarGAN. The first two lines are our method with/without using perceptual loss, respectively.

V. Experiments

We evaluate the proposed pipeline on a large-scale face dataset. Comparisons with baseline and state-of-the-art methods are for both the quantitative and qualitative results of the face image generation as well as the data augmentation performances for the attribute recognition task.

A. Dataset

We conduct experiments on the CelebFaces Attributes (CelebA) dataset [6], which comprises $202,599$ face images of $10,177$ identities, with each image having 40 binary attribute labels. The image quantities for training, validation, and test are $162,770$, $19,962$, and $19,867$ respectively. Following the settings in BEGAN [40], we crop the initial images from 178×218 to 128×128, and resize them as 64×64. We merge the training and validation sets to train our generation models.

We now list the abbreviations of the attribute labels used in some figures. 5'o Clock Shadow: 5CS; Arched Eyebrows: AE; Attractive: A; Bags Under Eyes: BUE; Bald: Bal; Bangs: Ban; Big Lips: BL; Big Nose: BN; Black Hair: BlaH; Blond Hair: BloH; Blurry: Blu; Brown Hair: BroH; Bushy Eyebrows: BE; Chubby: C; Double Chin: DC; Eyeglasses: E; Goatee: G; Gray Hair: GH; Heavy Makeup: HM; High Cheekbones: HC; Male: Ma; Mouth Slightly Open: MSO; Mustache: Mu; Narrow Eyes: NE; No Beard: NB; Oval Face: OF; Pale Skin: PS; Pointy Nose: PN; Receding Hairline: RH; Rosy Cheeks: RC; Sideburns: Si; Smiling: Sm; Straight Hair: SH; Wavy Hair: WHa; Wearing Earrings: WE; Wearing Hat: WH; Wearing Lipstick: WL; Wearing Necklace: WNl; Wearing Necktie: WNt; Young: Y.

B. Training

All the models are trained by Adam optimizers [49] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, which are the same as those for BEGAN [40].

At the pretraining stage, the *Encoder* and *Decoder* are trained using similar hyper parameters with original BEGAN [40]. For *Classifier* C, we train 10 epochs with the learning rate of 0.0001.

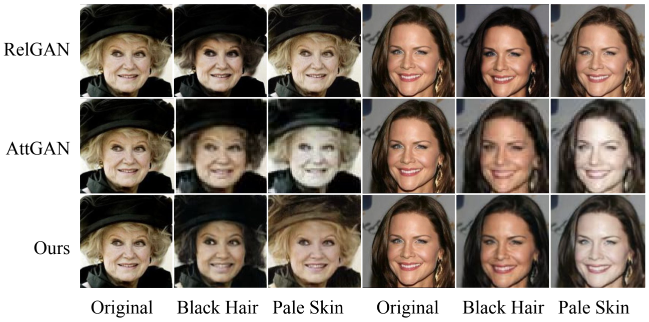


Figure 8. Comparing ours to related face image generation methods AttGAN [28] and RelGAN [50].

At the generation with the attributes stage, for the *Mapping function* Φ, we train 8 epochs. At first, the learning rate is 0.0001, then it decays by 50% after every 200k steps. The fully-connected layer discriminator D_{fc}, convolution layer discriminator D_{conv}, and *Mapping function* Φ are trained together. The hyper parameters in Eq. (17) are set as follows: $\lambda_1 = 1$, $\lambda_2 = 10$, $\lambda_3 = 1$, and $\lambda_4 = 10$.

C. Face image generation

Using our image generation approach, we can manipulate the target attribute labels to synthesize what we need. In Figure 6, we show the visualization results on seven instances with the 0/1 exchanges on both single and double labels. From a global viewpoint, we observe that the generated images are sensible to new labels. For example, from smiling to not smiling (and vice versa), we observe a clear change in facial emotion. When adding the change of face or hair colors, i.e. *H+S* and *P+S*, the results also make sense visually.

Qualitative comparison. In Figure 7, we compare our results (with perceptual loss) and those of the following approaches: without perceptual loss, DCGAN [19][4] and StarGAN [26]. In Figure 8, we compare our results with some state-of-the-art methods, e.g., AttGAN [28] and RelGAN [50]. Our results are clearly superior to both image quality and attribute changes.

The detailed comparisons include the following: (1) without perceptual loss, the model fails to generate sensible images for age and genders; (2) DCGAN is not able to generate high-resolution images from attribute labels; (3) StarGAN can generate face images with target attributes while the image quality is clearly inferior to ours. Besides, with the same computation source, our model takes 10 hrs, while StarGAN needs more than three days.

Method	DCGAN	StarGAN	Without Perceptual Loss	Our Method
FID	122.514	186.921	4.812	**1.319**

Table 1. Quantitative evaluation. For FID, lower scores are better.

[4] It is a variation of the original DCGAN with the same architecture of our framework. It is composed of *Mapping function*, *Decoder* and the original DCGAN discriminator. All components are trained from scratch.

	Method	Data Amount	Accuracy (%)
	Without Augmentation	original	80.9
Tra.	Random Flipping	double	83.9
	Random Flipping	4 times	83.2
	Random Cropping	double	85.1
	Random Cropping	4 times	84.2
	Random Erasing	double	85.4
	Random Erasing	4 times	83.4
By Gen.	StarGAN (random doubling)	double	83.7
	StarGAN (dynamic balancing)	double	82.9
	DCGAN (random doubling)	double	84.5
	DCGAN (dynamic balancing)	double	85.2
Ours	W/o Perceptual Loss (random doubling)	double	86.8
	W/o Perceptual Loss (dynamic balancing)	double	88.1
	Our Method (random doubling)	double	89.5
	Our Method (dynamic balancing)	1.5 times	87.1
	Our Method (dynamic balancing)	triple	88.7
	Our Method (dynamic balancing)	double	**91.1**

Table 2. Data augmentation results in 40 classes. *Tra.* is *Traditional* and *By Gen.* denotes the augmentation methods by image generation.

Quantitative comparison. The Fréchet Inception Distance (FID) is regarded as the best evaluation metric of generation quality [51], and the lower, the better. Table 1 shows the FID scores of comparable approaches. The performance of our generation approach is significantly superior over others.

D. Data augmentation

Rather than sensible image visualization, we are more interested in validating the data augmentation performance using the synthesized images. First, through the data doubling and balancing strategies proposed in Section IV, we augment the original training data in CelebA. Then, we train attribute recognition models using augmented datasets and report the recognition results on the original test set. We perform a comparison with traditional augmentation methods as well as the state-of-the-art methods with the same idea of user-generated images.

In the multi-label recognition task, we use the following definition of recognition accuracy Acc [28]. For each sample x, we compute the proportion of correctly recognized attributes, and then, take the average of the proportion over all test samples, *i.e.*

$$Acc_{\hat{x}} = \frac{\text{correct number}}{\text{attribute labels number}} \times 100\%,$$
$$Acc = \mathbb{E}[Acc_{\hat{x}}] \times 100\%.$$

Overview of data augmentation performance. Tables 3 and 2 show the attribute recognition accuracies on 13 classes (following AttGAN [28]) and 40classes (all classes), respectively. For both tasks, our method with a dynamic balancing strategy gives the best performance, i.e., 95.9% for 13-class recognition and 91.1% for 40-class recognition. Also, in Table 2, we obtain a significant improvement of 10.2% over the baseline of *Without Augmentation*.

Comparing to generation-based data augmentation methods. To have a fair comparison with AttGAN [28], the recognition models in Table 3 are trained on the following 13 chosen classes as follows: Bald, Brown Hair, Gender, No Bread, Bangs, Bushy Eyebrows, Mouth Open, Pale Skin, Black Hair, Eyeglasses, Mustache, Age and Blond Hair, following [28]. Moreover, the same CNN architecture with AttGAN [28] is applied to our methods, as shown in Table 3. Our dynamic balancing records the highest improvement, i.e., 2.9% higher than that obtained *Without Augmentation*.

Method	Data Amount	Accuracy (%)
Without Augmentation	original	92.6
AttGAN [28]	13 times	94.6
Our Method (randomly doubling)	double	94.6
Our Method (dynamic balancing)	double	**95.5**

Table 3. Data augmentation results in 13 classes.

The comparisons with other image generation methods[5] are given in Table 2. Using the same augmentation strategy, our generation model outperforms others. For instance, using dynamic balancing, we achieve 4.9% and 8.2% more than DCGAN and StarGAN, respectively. Our model is faster and easier to learn than DCGAN since we use pretrained *Encoder* and *Decoder*, which offer reliable guidance through self-supervised learning. Due to the perceptual loss from pretrained *Classifier*, we obtain 2.0% improvements. Compared with DCGAN architecture and StarGAN, by using the pre-trained *Encoder*, *Decoder* and *Classifier*, our model is faster and easier to learn. Thus our method's visualization quality is better under the same experimental settings, which leads to better performance. Therefore, the accuracy is 5.9% higher.

Comparing to traditional data augmentation methods. Table 2 also presents the recognition results of using traditional image augmentation methods: random flipping, random cropping [17] and random erasing [23]. The best result (85.4%) is achieved by random erasing (double), and is 5.7% lower than our best (double). These traditional methods affect the original images in many physical ways, but do not create any *new image*. In contrast, our method can generate novel images from target attribute labels.

Evaluating different augmentation sizes. We evaluate the effect of augmentation amount using the method of dynamic balancing. When we augment the dataset to 1.5 times, the accuracy increases from 80.9% to 87.1%; when we double the dataset, it further improves to 91.9%. However, when we augment the data to its triple size, the recognition rate drops to 88.7%. This is because the generated face images contain blurriness and artifacts, which have a negative effect on the classification.

Method	Recursive Time(s)	Accuracy (%)
Ours + CNN (dynamic balancing)	1	91.1
Ours + CNN (dynamic balancing)	2	**91.7**
Ours + CNN (dynamic balancing)	3	91.3

Table 4. The recognition accuracies for different recursive times of the pretrained classifier. The original dataset is augmented to double.

[5] In original papers of DCGAN and StarGAN, there is not data augmentation experiment.

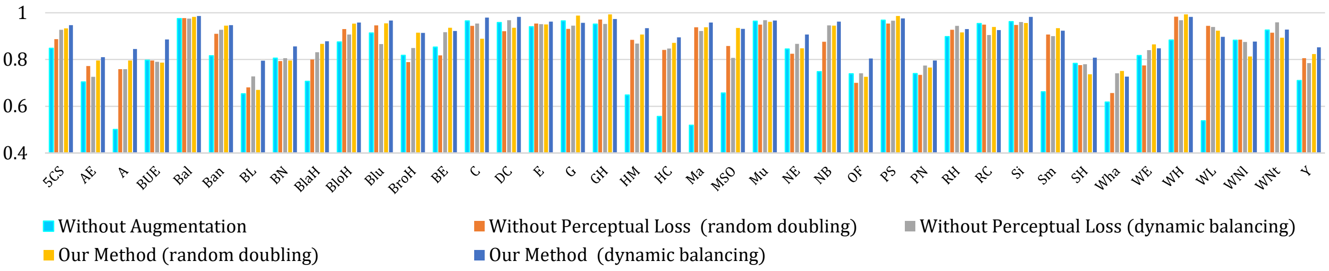


Figure 9. Recognition rates using different methods for each specific attribute class.

Recursive pretrained classifier. When training the *Mapping Function*, we use a pretrained classifier to calculate the perceptual loss. However, this classifier tends to suffers from an imbalance problem. To tackle the inherent bias for the pretraining, we additional designed a recursive framework for the prtrained classifier. If the number of recursive time is set to 1, the framework is exactly the same as the original method. If the number of recursive times is set to 2, the pretrained classifier is then replaced by a new one, with the dataset augmented by the framework when the number of recursive times is 1, and so on.

Table 4 shows the performance of different baselines for different recursive times. We observe that using the recursive strategy boosts the performance of "Ours + CNN (dynamic balancing)" improves by 0.6% when the number of recursive times is set to 2.

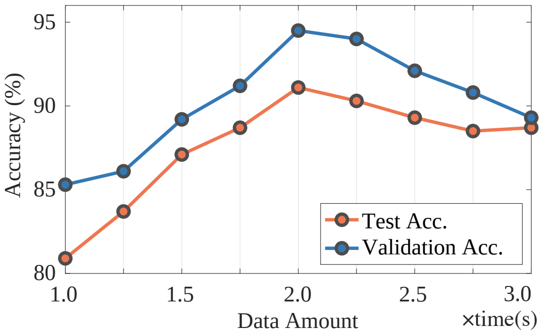


Figure 10. Validation and test accuracies with different augmentation sizes.

Augmentation size criterion. Due to the negative effect on recognition accuracy when over-augmenting the dataset, we propose the augmentation size criterion. For testing, we select the augmentation size according to the average accuracy of the validation set, i.e., the size that achieves the highest validation accuracy. Figure 10 shows the plots of validation and test accuracies for different augmentation sizes. We observe that the best augmentation size (2.0 times) for the test set is the same as that for the validation set on CelebA. This proves the efficiency of choosing the augmentation size according to the average accuracy of the validation set.

Recognition of specific attributes. Figure 9 shows the recognition results for 40 attribute classes. Our methods, with or without perceptual loss, perform clearly better than *Without Augmentation* for most classes. Obviously, higher scores are achieved for recognizing *Attractive* (AE), *Heavy Makeup* (HM), *High Cheekbones* (HC), *Male* (Ma) and *Wearing Lipstick* (WL). Note that these attributes are mainly related to gender.

Using the same augmentation strategy with perceptual loss improves the recognition of most attribute categories. Applying the same dynamic balancing, the following considerable improvements are achieved: 6.7% for *Young* (Y), 6.3% for *Oval Face* (OF), 8.6% for *Attractive* (A) and 12.4% for *Mouth Slightly Open* (MSO).

However, our method fails to improve the performance for some attributes such as *Rosy Cheeks* (RC), *Wearing Lipstick* (WL) and *Wearing Necktie* (WNl). It can be explained that the imbalance problem for these attributes does not pose a threat, and the performance for the baselines is already fairly high. Further data augmentation leads to additional noise for these classes, degrading the performance.

VI. Conclusions

We proposed a new data augmentation method by generating novel face images given arbitrary attribute labels. To narrow the gap between the target discrete attribute labels and the continuous image space, we leveraged image reconstruction as a guidance for attribute-conditional generation. Considering generated images, we used attribute perceptual losses to penalize wrong attribute recognition while regularizing attribute-conditional face image generation. In our experiments, we validated the generation performance of our method, as well as the efficiency of using generated images in two augmentation strategies. We found that (1) FID scores indicated our superiority over both baseline DCGAN and state-of-the-art StarGAN; (2) augmentation with random doubling tackled the small-data problem to some extent; and (3) using too many generated data for augmentation involved too much image noise and thus reduce the model performance. For future work, we will work on improving the quality and diversity of generated face images.

References

[1] T. Berg and P. N. Belhumeur, "POOF: part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," in *CVPR*, 2013, pp. 955–962. 1, 2

[2] O. K. Manyam, N. Kumar, P. N. Belhumeur, and D. J. Kriegman, "Two faces are better than one: Face recognition in group photographs," in *IJCB*, 2011, pp. 1–8. 1, 2

[3] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Faceness-net: Face detection through deep facial part responses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 8, pp. 1845–1859, 2017. 1, 2

[4] Y. Zhen, Y. Gao, D.-Y. Yeung, H. Zha, and X. Li, "Spectral multimodal hashing and its application to multimedia retrieval," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 27–38, 2015. 1

[5] X. Zhao, N. Wang, Y. Zhang, S. Du, Y. Gao, and J. Sun, "Beyond pairwise matching: Person reidentification via high-order relevance learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3701–3714, 2017. 1

[6] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015, pp. 3730–3738. 1, 2, 7

[7] M. Elhoseiny and M. Elfeki, "Creativity inspired zero-shot learning," in *ICCV*, 2019, pp. 5784–5793. 1

[8] H. Huang, C. Wang, P. S. Yu, and C.-D. Wang, "Generative dual adversarial network for generalized zero-shot learning," in *CVPR*, 2019, pp. 801–810. 1

[9] M. B. Sariyildiz and R. G. Cinbis, "Gradient matching generative networks for zero-shot learning," in *CVPR*, 2019, pp. 2168–2178. 1

[10] J. Li, M. Jing, K. Lu, Z. Ding, L. Zhu, and Z. Huang, "Leveraging the invariant side of generative zero-shot learning," in *CVPR*, 2019, pp. 7402–7411. 1

[11] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *CVPR*, 2019, pp. 403–412. 1

[12] X. Li, Q. Sun, Y. Liu, Q. Zhou, S. Zheng, T.-S. Chua, and B. Schiele, "Learning to self-train for semi-supervised few-shot classification," in *NeurIPS*, 2019, pp. 10 276–10 286. 1

[13] Y. Liu, B. Schiele, and Q. Sun, "An ensemble of epoch-wise empirical bayes for few-shot learning," *arXiv*, vol. 1904.08479, 2020. 1

[14] Y. Liu, Y. Su, A. Liu, B. Schiele, and Q. Sun, "Mnemonics training: Multi-class incremental learning without forgetting," in *CVPR*, 2020. 1

[15] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *CVPR*, 2014, pp. 1717–1724. 1

[16] Q. Sun, L. Ma, S. J. Oh, L. V. Gool, B. Schiele, and M. Fritz, "Natural and effective obfuscation by head inpainting," in *CVPR*, 2018, pp. 5050–5059. 1, 3, 5

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1106–1114. 1, 2, 8

[18] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *CVPR*, 2015, pp. 3982–3991. 1, 2

[19] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016. 1, 2, 7

[20] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, 2017. 2

[21] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *NIPS*, 2014, pp. 1988–1996. 2

[22] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Technical Report 07-49, University of Massachusetts, Amherst, Tech. Rep., 2007. 2

[23] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv*, vol. 1708.04896, 2017. 2, 8

[24] Z. Liu, G. Song, J. Cai, T. Cham, and J. Zhang, "Conditional adversarial synthesis of 3d facial action units," *Neurocomputing*, vol. 355, pp. 200–208, 2019. 2

[25] E. M. Rudd, M. Günther, and T. E. Boult, "MOON: A mixed objective optimization network for the recognition of facial attributes," in *ECCV*, 2016, pp. 19–35. 2

[26] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *CVPR*, 2018, pp. 8789–8797. 2, 7

[27] W. Shen and R. Liu, "Learning residual images for face attribute manipulation," in *CVPR*, 2017, pp. 1225–1233. 2

[28] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "Attgan: Facial attribute editing by only changing what you want," *IEEE Transactions on Image Processing*, 2019. 2, 7, 8

[29] M. Li, W. Zuo, and D. Zhang, "Deep identity-aware transfer of facial attributes," *arXiv*, vol. 1610.05586, 2016. 2

[30] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *CVPR*, 2018, pp. 8798–8807. 2

[31] F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, and W. T. Freeman, "Synthesizing normalized faces from facial identity features," in *CVPR*, 2017, pp. 3386–3395. 2

[32] B. Cao, N. Wang, J. Li, and X. Gao, "Data augmentation-based joint learning for heterogeneous face recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 6, pp. 1731–1743, 2019. 2

[33] A. Creswell and A. A. Bharath, "Inverting the generator of a generative adversarial network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 1967–1974, 2019. 2

[34] J. Yu, X. Gao, D. Tao, X. Li, and K. Zhang, "A unified learning framework for single image super-resolution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 4, pp. 780–792, 2014. 2

[35] L. Ma, Q. Sun, B. Schiele, and L. Van Gool, "A novel bilevel paradigm for image-to-image translation," *arXiv*, vol. 1904.09028, 2019. 2

[36] Q. Sun, A. Tewari, W. Xu, M. Fritz, C. Theobalt, and B. Schiele, "A hybrid model for identity obfuscation by face replacement," in *ECCV*, 2018. 2

[37] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "f-vaegan-d2: A feature generating framework for any-shot learning," in *CVPR*, 2019, pp. 10 275–10 284. 2

[38] A. Paul, N. C. Krishnan, and P. Munjal, "Semantically aligned bias reducing zero shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7056–7065. 2

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014. 2

[40] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: boundary equilibrium generative adversarial networks," *arXiv*, vol. 1703.10717, 2017. 2, 3, 5, 7

[41] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez, "Invertible conditional gans for image editing," *arXiv*, vol. 1611.06355, 2016. 2

[42] L. Wan, J. Wan, Y. Jin, Z. Tan, and S. Z. Li, "Fine-grained multi-attribute adversarial learning for face generation of age, gender and ethnicity," in *ICB*, 2018, pp. 98–103. 2

[43] L. Ma, Q. Sun, S. Georgoulis, L. V. Gool, B. Schiele, and M. Fritz, "Disentangled person image generation," in *CVPR*, 2018, pp. 99–108. 3

[44] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. V. Gool, "Pose guided person image generation," in *NIPS*, 2017, pp. 406–416. 3, 5

[45] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*, 2016, pp. 694–711. 4

[46] B. Schölkopf, J. Platt, and T. Hofmann, "Learning to traverse image manifolds," in *NIPS*, 2007, pp. 361–368. 5

[47] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, 2013. 5

[48] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010, pp. 807–814. 5

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv*, vol. 1412.6980, 2014. 7

[50] P.-W. Wu, Y.-J. Lin, C.-H. Chang, E. Y. Chang, and S.-W. Liao, "Relgan: Multi-domain image-to-image translation via relative attributes," in *CVPR*, 2019, pp. 5914–5922. 7

[51] 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 *NIPS*, 2017, pp. 6626–6637. 8