Learning to Generate and Edit Hairstyles

Weidong Yin[1], Yanwei Fu[1,†], Yiqiang Ma[1]
Yu-Gang Jiang[2], Tao Xiang[3], Xiangyang Xue[1,2]
[1]School of Data Science, Fudan University; [2]School of Computer Science, Fudan University;
[3] Queen Mary University of London , † Corresponding author. Email: yanweifu@fudan.edu.cn

ABSTRACT

Modeling hairstyles for classification, synthesis and image editing has many practical applications. However, existing hairstyle datasets, such as the Beauty e-Expert dataset, are too small for developing and evaluating computer vision models, especially the recent deep generative models such as generative adversarial network (GAN). In this paper, we contribute a new large-scale hairstyle dataset called Hairstyle30k, which is composed of 30k images containing 64 different types of hairstyles. To enable automated generating and modifying hairstyles in images, we also propose a novel GAN model termed Hairstyle GAN (H-GAN) which can be learned efficiently. Extensive experiments on the new dataset as well as existing benchmark datasets demonstrate the effectiveness of proposed H-GAN model.

KEYWORDS

Hairstyle dataset, Hairstyle Classification, Generative Adversarial Networks

ACM Reference format:
Weidong Yin[1], Yanwei Fu[1,†], Yiqiang Ma[1] and Yu-Gang Jiang[2], Tao Xiang[3], Xiangyang Xue[1,2] . 1997. Learning to Generate and Edit Hairstyles. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 9 pages.
DOI: 10.475/123_4

1 INTRODUCTION

Hairstyle can express one's personalities, self-confidence, and attitudes. It is thus an important aspect of personal appearance. A computer vision model that enables recognition, synthesis, and modification of hairstyles in images is of great practical use. For example, with such as model, customer can take a photo of him/herself and then synthesize different hairstyles before going to the hairdresser's to make the most satisfactory one a reality. In addition, an automated hairstyle recognition model can be used for recognizing person's identity for security applications.

Existing efforts on hairstyle modeling have been focused on recommending the most suitable hairstyles [18], or interactively users' editing [7, 22, 32]. However, there is no attempt so far to systematically study hairstyles in images and no model available that can address various hairstyle modeling task in a comprehensive manner. One of the reasons is that there are large variations in hairstyles and in order to model these variations, large-scale datasets are needed. Unfortunately, such a large-scale hairstyle dataset does not exist. In Multimedia and computer vision communities, hairstyles are often labeled as attributes for face datasets. However, such annotation is often crude, focusing mostly hair length and color. On the other hand, existing specialized hairstyle datasets such as Beauty e-Expert dataset [18] are too small to represent the diversity of human hairstyles in the wild.

In this paper, we introduce the first large-scale hairstyle dataset – Hairstyle30K to the community and hope that this will greatly boost the research into hairstyle modeling. Images in the dataset (see Fig. 1 for examples) are collected from the Web via search engines using keywords corresponding a hairstyle ontology. This results in 64 different types of hairstyles in 30K images. On average, each hairstyle class has around 480 images. The newly proposed dataset is used to train the H-GAN model proposed in this paper. Importantly, with 64 hairstyle classes, this is a fine-grained dataset presenting a challenging recognition task, as verified by our experiments.

Apart from releasing a new dataset, we also present a Hairstyle Generative Adversarial Network (H-GAN) model for automatically generating or modifying/editing hairstyles given an input image. Our H-GAN has three components: an encoder-decoding sub-network, a GAN and a recognition subnetwork. Particularly, the encoder-decoding network is a variant of Variational Auto-Encoders (VAE) [12]; the recognition sub-network shares the same networks as the discriminator of GAN as in InfoGAN [5]. The model is unique in that once trained, it can be used to perform various tasks including recognition, synthesis and modification. Extensive experiments of our H-GAN algorithm on the proposed dataset and other general-purpose benchmark datasets validate the efficacy of our model.

Contributions. We make several contributions in this paper. Firstly, to study the hairstyle related problems, we contribute a new large-scale hairstyle dataset – Hairstyle30k to the community. To the best of our knowledge, this is the largest hairstyle dataset, especially in terms of the number of hairstyle classes. Secondly, we present a new deep generative model called – H-GAN which can effectively and efficiently generate and modify the hairstyles of person images. Extensive experiments demonstrate that our H-GAN is superior to a number of state-of-the-art alternative models.

2 RELATED WORK

2.1 Image Editing and Synthesis

Editing image with interaction. Recent advances in interactive image segmentation have significantly simplified the tasks of object

Conference'17, Washington, DC, USA

DOI: 10.475/123_4

Figure 1: Examples of our Hairstyle30K dataset with corresponding hairstyle class labels.

segmentations [3, 4, 16, 25, 38]. Existing interactive segmentation approaches such as lazy snapping [16] and grab cut [25] as well as recent Generative Adversarial Networks (GAN) related methods [44] enable the users to achieve good quality object cutout with a few of strokes. In comparison, the existing efforts on hairstyles and makeup editing are very primitive [7, 22, 32]. In theory, these interactive image editing works can be used for editing hairstyles. However, it can be tedious and time-consuming to manually modify hairstyles via user-interaction. Fully automated image editing thus becomes desirable.

Automated image editing. There are some recent efforts on fully automated images editing [10, 24, 28, 43]. In particular, two recent studies [24, 28] propose approaches to modify the attribute of facial images. The proposed H-GAN is an automated image editing model but focuses on hairstyle images.

Image Editing and Synthesis. Our work is also related to previous work on joint image editing and synthesis [10, 14, 24, 28, 39]. Shen *et al.* [28] manipulated the facial attributes by a GAN-based image transformation networks; nevertheless each trained model in [28] can only modify one special type of facial attribute images. In contrast, one trained model of our H-GAN can modify all hairstyles presented in the training data. In model proposed by [39] , conditional variational auto-encoder is used to generate facial images of different attributes; due to the lack of the adversarial loss, the generative images are often blurry. In VAEGAN[14], VAE is combined with GAN to generate more realistic image. However, compared to our H–GAN, it does not use attribute information and modification is achieved by calculating residual attribute vector. The main difference between VAEGAN and H_GAN is thus the fact that we add attribute information explicitly into the generator so that we can specify different attributes that we want to change. Also we introduce a recognition network to maximize the mutual information between attributes and generated images to generate images with specified attributes.

2.2 Attribute Analysis

Attribute-based people search. Hairstyles can be considered as a special type of person attributes. Such attributes can be used in the applications of surveillance environments, including but not limited in attribute-based people search [29, 34, 36] and person re-identification [15, 21, 31, 42]. Recent studies on person attributes are focused on clothing. These include clothes recognition [37], clothing parsing [17] as well as clothing retrieval [2, 41]. The study presented in this paper complements the existing clothing oriented attribute analysis.

Face attribute analysis. It is another important research topic related to our hairstyle analysis. Facial attribute analysis was first studied by Kumar et al. [13]. In terms of different visual features and distinctive learning paradigm, facial attribute analysis has been developed into three categories: (1) the methods using hand-crafted visual features [13], such as SIFT [20] and LBP [23]; (2) the methods utilizing the recent deep features [19, 36]; and (3) multi-task methods for learning facial attribute [1, 6, 26].

3 DATASET COLLECTION

Our hairstyle30*k* dataset is designed for studying the problem of hairstyle classification as well as other hairstyle related tasks including synthesis and editing. To construct the dataset, we had downloaded more than 1 million images using various web search engines (Google, Flicker, and Bing, etc) with hairstyle related search words, e.g. *Beehive hairstyle*. The full set of class names of hairstyles are listed in Fig. 1. The initial downloaded images were firstly filtered by face detection algorithm. We subsequently pruned some irrelevant or erroneous images which has neither faces nor hairstyles. We then manually filtered out the irrelevant images for some hairstyles

that come without faces, *e.g.*, *Ducktail*. We carefully annotated the pruned images and classified them into different hairstyle classes. Finally, we obtained $30k$ images with 41 types of male hairstyles and 42 types of female hairstyles. Among them, 19 kinds of hairstyle have both male and female versions. Thus totally, the dataset has 64 different types of hairstyles.

3.1 Statistics

The number of images of each hairstyle class are varied in term of how popular this hairstyle is. In general, similar to most object classification dataset [33], we also observe a long-tailed distribution of the number of hairstyle instances over classes as illustrated in Fig. 2. On average, each hairstyle has around 480 images.

3.2 Uniqueness

Existing publicly available datasets for academic research either have too few image (e.g., the Beauty e-Experts dataset); or too few hairstyle classes (e.g., the CelebA dataset). Specifically, The Beauty e-Expert dataset [18] has only 1505 female figures in distinct fashions; in contrast, our hairstyle dataset has around $30K$ instances. The general-purpose face dataset CelebA [19] has around $200K$ celebrity figures with 40 annotated attributes. Nevertheless, in CelebA, only very few and very simple hairstyles are labeled as the attributes, e.g., wavy hair. Generally, the targets of our dataset is also different from CelebA, since ours is a benchmark dataset for recognizing different hairstyles; and the images within each hairstyle class can cover large pose variations and background clutter. Importantly, The images of the same person with different hairstyles should be categorized into different hairstyle classes.

3.3 Applications

The datasets can be used to develop different applications. Specifically, the task of recognizing different hairstyles belongs to the category of fine-grained classification, which is an active and yet very challenging research topic in the multimedia community, e.g. [11, 35]. Potentially, this dataset can serve as the benchmark dataset for many real-world applications and tasks such as hairstyle retrieval and recommendation systems [18], and the research of recognizing fine-grained hairstyles, and automatically generating and changing the hairstyles. In this next section, we propose a framework that enables three tasks, *i.e.* recognition, generation and modification of hairstyles, by using a single model.

4 HAIRSTYLE GAN (H-GAN)

4.1 Background

GAN [8] targets at learning to discriminate real data samples from generated samples by training the generator network G to fool the discriminator network D. It is formulated to optimize the following objective functions,

$$\min_G \max_D \mathcal{L}_{GAN} = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_{prior}(z)}\left[\log\left(1 - D(G(z))\right)\right], \quad (1)$$

where $p_{data}(x)$ and $p_{prior}(z)$ are the distributions of real data and Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{1})$. The training process iteratively updates the parameters of G and D with the loss functions $\mathcal{L}_D = \mathcal{L}_{GAN}$ and $\mathcal{L}_G = -\mathcal{L}_{GAN}$ for generator and discriminator respectively. The generator can draw a sample $z \sim p_{prior}(z) = \mathcal{N}(\mathbf{0}, \mathbf{1})$ and utilized the generator network G, *i.e.*, $G(z)$ to generate an image.

InfoGAN [5] further models the noise variable z in Eq (1) by decomposing it into a latent representation y and incompressible noise z. To ensure no loss information of latent representation in the generation, InfoGAN maximizes the mutual information $I(y; G(z, y))$ as the recognition loss,

$$\mathcal{L}_{rg} = -\mathbb{E}_{x\sim G(z,y)}\left[\mathbb{E}_{y\sim p_{data}(y|x)}\left[\log Q(y \mid x)\right]\right] \quad (2)$$

where $Q(y \mid x)$ is an approximation of the posterior $p_{data}(y \mid x)$. InfoGAN can unsupervisedly learn disentangled, interpretable and meaningful representations with the loss function of the generator G as $\mathcal{L}_{G_{InfoGAN}} = \mathcal{L}_G - \mathcal{L}_{rg}$

VAEGAN [14] integrates the Variational Auto-Encoders (VAE) into GAN. It uses the feature-wise errors to replace the element-wise errors of original GAN in the data space. The VAE part encodes the data sample x to latent representation z: $z \sim \text{Enc}(x) = p_{enc}(z \mid x)$ and decodes the z back to data space: $\tilde{x} \sim \text{Dec}(x \mid z) = p_{dec}(x \mid z)$ by two loss functions: (1) the regularization of the latent space $\mathcal{L}_{prior} = KL\left(q_{enc}(z \mid x) \parallel p_{prior}(z)\right)$, where $q_{enc}(z \mid x)$ is the approximation to the true posterior $p_{dec}(z \mid x)$; (2) the reconstruction error

$$\mathcal{L}_{recon}^{D_l} = -\mathbb{E}_{q_{enc}(z|x)}\left[\log p_{dec}\left(D_l(x) \mid z\right)\right] \quad (3)$$

where $D_l(x)$ is hidden representation of $l-th$ layer of the discriminator. Thus the loss functions of VAEGAN are updated as $\mathcal{L}_D = \mathcal{L}_{GAN}$, $\mathcal{L}_{dec} = \mathcal{L}_G = -\mathcal{L}_{GAN} + \lambda \cdot \mathcal{L}_{recon}^{D_l}$ and the encoder $\mathcal{L}_{enc} = \mathcal{L}_{prior} + \mathcal{L}_{recon}^{D_l}$, where λ is the coefficient. However the latent representation z is unsupervised learned and not explicitly associated with any nameable attributes.

CVAE [30, 39] is short for the conditional VAE. CVAE introduces an independent attribute y to control the generating process of x by sampling from $p(x \mid y, z)$; where $p(y, z) = p(y)\,p(z)$.
The encoder and decoder networks are thus $z \sim Enc(x) = q_{enc}(z \mid x)$ and $\tilde{x} \sim Dec(z, y) = p_{dec}(x \mid z, y)$. The variable y is introduced to control the generate process of x by sampling from $p(x \mid y, z)$; where $p(y, z) = p(y)\,p(z)$. Nevertheless, y is still sampled from data, but not directly optimized and learned from the data.

4.2 Hairstyle GAN (H-GAN)

Hairstyle GAN model is formulated for generating and modifying hairstyles in a single framework. Particularly, besides the input noise z in GAN, we utilize the independent hairstyle variables y of each image x, *i.e.* $y \sim p_{data}(y \mid x)$. Mathematically, we have *i.e.*, $p(y, z) = p(y)\,p(z)$.

As overviewed in Fig. 3, the network of H-GAN has three components: an encoder-decoding sub-network, a GAN sub-network and a recognition sub-network. The network structure is explained in Tab. 1. The whole H-GAN network is trained together in an end-to-end fashion. Once trained, the encoder-decoding sub-network enables generating and changing hairstyles; the recognition sub-network can be used for hairstyle classification. Note that the recognition sub-network and discriminator of GAN share the same network structures except the last softmax classification layer, since both

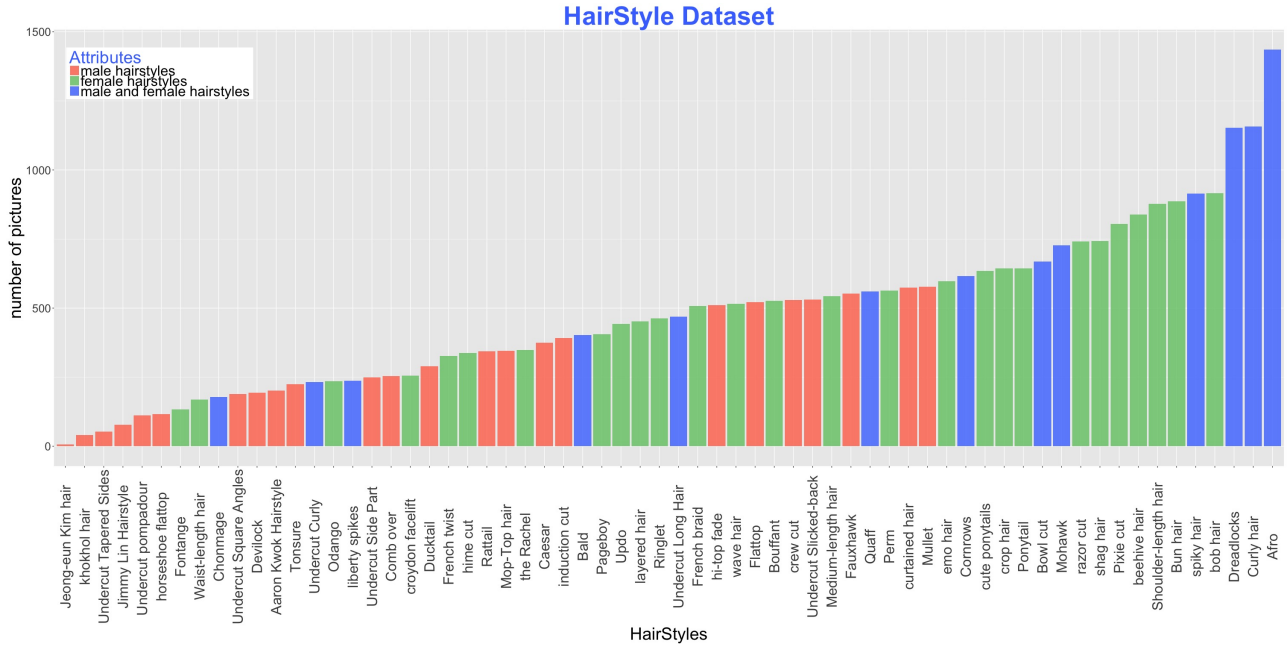

Figure 2: The statistics of the hair-style Dataset. In the X-axis, we list the hairstyles. The number of pictures of each hairstyle class are compared in Y-axis. We use three color attributes to annotate each hairstyle: red and green colors indicate the hairstyle for male only and female only respectively. The blue color means the hairstyle both for male and female.

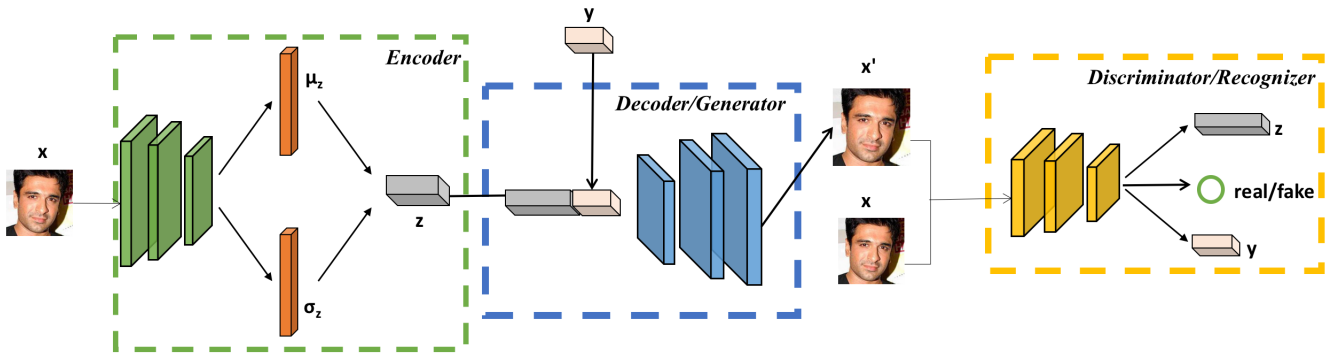

Figure 3: Overview of our H-GAN.

sub-networks are assigned the classification task with the same input. The loss functions of the generator G_{H-GAN} , discriminator D_{H-GAN}, encoder and decoder are thus updated as,

$$\mathcal{L}_{G_{H-GAN}} = \mathcal{L}_G + \lambda_1 \left(\mathcal{L}_{rg_z} + \mathcal{L}_{rg_y - G}\right) + \lambda_2 \mathcal{L}_{recon}^{D_l} \quad (4)$$

$$\mathcal{L}_{D_{H-GAN}} = \mathcal{L}_D + \left(\mathcal{L}_{rg_y - D} + \mathcal{L}_{rg_z}\right) \quad (5)$$

$$\mathcal{L}_{enc} = \mathcal{L}_{VAE} = \mathcal{L}_{prior} + \mathcal{L}_{recon}^{D_l} \quad (6)$$

$$\mathcal{L}_{dec} = \mathcal{L}_{G_{H-GAN}} \quad (7)$$

where λ_1 and λ_2 are the weight of corresponding term; The reconstruction loss of Eq (3) is updated as

$$\mathcal{L}_{recon}^{D_l} = -\mathbb{E}_{z\sim p_{enc}(z|x), y\sim p_{data}(y|x)} \left[\log p_{dec}\left(D_l\left(x\right) \mid z, y\right)\right] \quad (8)$$

where $D_l(x)$ is the hidden representation of $l-th$ layer of the discriminator. It measures the loss of reconstructing generated images by sampling the z. $\mathcal{L}_{rg_z}$ is the recognition loss on z, which is defined as

$$\begin{aligned}\mathcal{L}_{rg_z} = &-\mathbb{E}_{x\sim p_{data}(x), z\sim p_{enc}(z|x)} \left[\log\left(Q\left(z \mid x\right)\right)\right] \\ &-\mathbb{E}_{x\sim p_{dec}(x|z,y), y\sim p_{data}(y|x), z\sim p_{enc}(z|x)} \left[\log\left(Q\left(z \mid x\right)\right)\right] \\ &-\mathbb{E}_{x\sim p_{dec}(x|y,z), y\sim p_{data}(y), z\sim p_{prior}(z)} \left[\log\left(Q\left(z \mid x\right)\right)\right] \\ &-\mathbb{E}_{x\sim p_{dec}(x|z,y), y\sim p_{data}(y), z\sim p_{enc}(z|x)} \left[\log\left(Q\left(z \mid x\right)\right)\right],\end{aligned} \quad (9)$$

where the first term measures the loss of predicting errors on real data; and rest three terms are the loss functions on generated data. $Q\left(\cdot\right)$ is an approximation of the corresponding posterior data distribution. $p_{enc}\left(z_{la} \mid x\right)$ is the distribution of z given x parameterized

Encoder $(128 \times 128)^{\dagger}$	Decoder $(128 \times 128)^{\dagger}$	Discrim/Recog$^{\curlywedge}$
64@C (6×6)	$\mid y \mid$ +256 FC	64@C(6×6)
128@C(4×4)	8064FC -> $4 \times 4 \times 7 \times 72$ FC	128@C(4×4)
256@C(4×4)	288@DeC(3×3)	128@C(4×4)
256@C(4×4)	216@DeC(3×3)	256@C(4×4)
256×2 FC $^{\nearrow}$	144@DeC(5×5)	1 FC/256+ $\mid y \mid$ FC
	72@DeC(5×5)	
	3@DeC(6×6)	

Table 1: The details of networks. Stride is 2 for all layers. C and Dec indicate the convolutional and deconvolutional filter individually. 64@C (4×4) means 64 convolutional filters with size 4×4. FC means the fully connected layer. The Discriminator and Recognition networks share the same four layers (the first four layers in the third column). The last layer of Discriminator and Recognition networks are 1 and $256+ \mid y \mid$ neurons individually. Note that: (1) † and $^{\curlywedge}$ indicate the activation of ReLU and Leaky ReLU (ratio: 0.2) for the corresponding networks. (2)$^{\nearrow}$ indicates the reparameterization trick [12]; specifically, we take $z \sim \mathcal{N}(\mu_z, \sigma_z)$; and 256 and 256 neurons to regress μ and σ respectively. On decoding part, $z = \mu + \epsilon\sigma, \epsilon \sim \mathcal{N}(0, 1)$. (3) $\mid y \mid$ means the number of hairstyles. (4) -> denotes the reshape operation.

by the encoder network; $p_{data}(y)$ is the data distribution of y on real data; $p_{data}(y \mid x)$ is the data distribution of y given x on the real data; $p_{data}(x)$ is the data distribution of x on the real data; $p_{prior}(z)$ is the prior distribution of z and we use the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$; $p_{dec}(x \mid z, y)$ is the distribution of x given z and y, and the distribution is parameterized by the decoder network; $p_{enc}(z \mid x)$ is the distribution of z given x and the distribution parameterized by the encoder network [40].

For the recognition loss on y, the loss functions for the discriminator and generator are defined as,

$$\mathcal{L}_{rg_y-D} = -\mathbb{E}_{x\sim p_{data}(x), y\sim p_{data}(y|x)}\left[\log\left(Q\left(y \mid x\right)\right)\right] \tag{10}$$

$$\begin{aligned}\mathcal{L}_{rg_y-G} = &-\mathbb{E}_{x\sim p_{dec}(x|y,z), z\sim p_{enc}(z|x), y\sim p_{data}(y)}\left[\log\left(Q\left(y \mid x\right)\right)\right] \\ &-\mathbb{E}_{x\sim p_{dec}(x|y,z), z\sim p_{enc}(z|x), y\sim p_{data}(y|x)}\left[\log\left(Q\left(y \mid x\right)\right)\right] \\ &-\mathbb{E}_{x\sim p_{dec}(x|y,z), z\sim p_{prior}(z), y\sim p_{data}(y|x)}\left[\log\left(Q\left(y \mid x\right)\right)\right]\end{aligned} \tag{11}$$

where for the discriminator$\mathcal{L}_{rg_y-D}$, only the real data is used to train the model since the quality of generated data in the training process is unreliable.

Training algorithms. The training of H-GAN is optimized by many epochs; each epoch is divided into three stages, namely,

(1) *Learning image reconstruction*: we update the encoder-decoder subnetwork and learn to reconstruct the image given the desired hairstyle. Specifically, we sample a batch of images $x \sim p_{data}(x)$ and hairstyle $y \sim p_{data}(y|x), z \sim p_{enc}(z \mid x)$ to update the encoder, and decoder, by minimizing $\mathcal{L}_{enc}$ and $\mathcal{L}_{dec}$ individually.

(2) *Learning image modification*: Given an image x and desired hairstyle y, this stage learns of modifying image x with the y hairstyle. In particular, we firstly sample a batch of images $x \sim p_{data}(x)$ and the hairstyle $y \sim p_{data}(y), z \sim p_{enc}(z \mid x)$, to update the decoder and discriminator by minimizing $\mathcal{L}_{dec}$ and $\mathcal{L}_{D_{SL-GAN}}$ respectively.

(3) *Learning image generation*: We sample a batch of latent vectors $z \sim p_{prior}(z)$ and the hairstyle $y \sim p_{data}(y)$; to minimize the decoder $\mathcal{L}_{dec}$ and discriminator with $\mathcal{L}_{D_{SL-GAN}}$ iteratively.

4.3 Generation and modification of hairstyles

Our H-GAN can be used to perform both tasks.

Generation of hairstyles. To generate a new hairstyle image, we can sample z from $p_{prior}(z)$ and setting y to any desired hairstyle. The image can be generated as $x' \sim G(z, y)$.

Modification of hairstyles. For efficient image editing, we proposed to utilize the residual differences between the desired hairstyle and all hairstyles. The resultant hairstyle modification algorithm takes two steps. (1) Given an image x and the desired hairstyle $y_{desired}$, we first sample $z \sim p_{enc}(z \mid x)$. (2) We employ the encoder to compute the corresponding z of all images. We compute the

$$\bar{z}_{y_{desired}} = \mathbb{E}_{z\sim p_{enc}(z|x), x\sim p(x|y_{desired})}\left[z\right] \tag{12}$$

$$\bar{z} = \mathbb{E}_{z\sim p_{enc}(z|x), x\sim p_{data}(x)}\left[z\right] \tag{13}$$

where $\bar{z}_{y_{desired}}$ is the mean vector of images with the desired hairstyle; and $\bar{z}$ is the mean vector for all the images. We then compute the $\triangle = \bar{z}_{y_{desired}} - \bar{z}$; and the modified image can be generated by $x' \sim p_{dec}(x \mid z + \triangle, y_{desired})$.

5 EXPERIMENTS

5.1 Experimental setup

Dataset. We conduct the experiments on three datasets.

(1) Hairstyle30K is the newly proposed dataset in this paper. This dataset has totally 30911 images of 64 different fine-grained male and female hairstyles. On average, each hairstyle has 480 images.

(2) CelebA is a facial attribute dataset of approximately 200k images of 10k identities [19]. Each image is annotated with 5 landmarks (two eyes, the nose tips, the mouth corners) and binary labels of 40 attributes. Among all these attributes, 8 attributes are related to hairstyle and thus can be used to evaluate our H-GAN algorithm.

(3) Male hairstyle dataset combines the male hairstyles from both Hairstyle and CelebA dataset. Particularly we divide the male styles into 6 different male hairstyle classes, namely, *bald, bang, curly, long, undercut-short,* and *undercut-long*. We re-annotate the images from Hairstyle and CelebA dataset into these 6 categories. Totally, this dataset has 38293 images, and each class on average contains 6382 images. Compare to the original Hairstyle30K dataset, this dataset has much few classes but each class has much more samples and suffers less the class imbalance problem.

Evaluation. We employ several different evaluation metrics to evaluate our proposed dataset and H-GAN algorithm. (1) *Classification of hairstyles.* We evaluate the tasks of the classification tasks of the Hairstyle30K dataset. Particularly, in each hairstyle class, the images are equally sliced into two halves. 50% images are used for training while the rest images are saved as the testing data. The mean accuracy (i.e., the mean of the diagonal of the confusion matrix) is employed as the metrics for evaluation due to the data

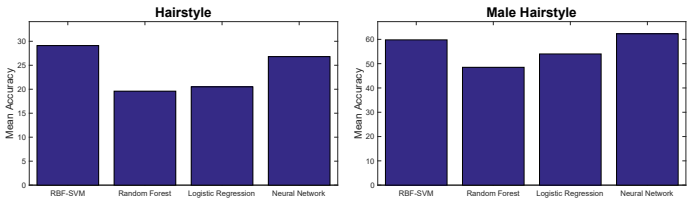

Figure 4: The results of classification on two hairstyle datasets. The chance-level are 1.56% and 16.7% respectively for the hairstyle30k and male hairstyle datasets respectively.

Figure 5: Qualitative results on the hairstyle generation task using the Male Hairstyle dataset.

Dataset	H-GAN	VAEGAN	Attrb2img
Male hairstyle	2.28	2.26	2.31
CelebA	2.29	2.08	2.32

Table 2: Inception Scores on CelebA and Male hairstyle datasets. The higher values, the better results.

imbalance between classes. The experiments are repeated for 10 round to reduce the variance. (2) *Hairstyle generation.* we employ to evaluate the hairstyle generation, the inception scores [27] which aims at measuring whether the varied images are generated and whether the generated images contains meaningful objects. (3) *Modifying hairstyles.* A user-study is carried out to evaluate the effects of changing hairstyles by different methods.

Implementation details. Our model is implemented based on the tensor flow platform and our model get converged in 4-5 hours on Male hairstyle dataset on GeForce GTX 1080; and our model needs around 8GB GPU memory. We use the Adam optimizer to train the generator and the discriminator is trained by the Rmsprop optimizer with a mini-batch size of 64. The learning rate is set as $1e-4$ and $2e-4$ for generator/encoder and discriminator respectively. The input size of images is 128×128. Our H-GAN is trained by 10 epochs, and each epoch has 5000 iterations.

5.2 Classification of Hairstyles

To investigate the classification performance of some baselines, we extract the deep features of all the images by Resnet-50 net [9]. On the Hairstyle30k and Male hairstyle datasets, we compare several baselines: (1) SVM: a SVM with RBF kernel is trained for the classification. (2) Random Forest: 100 estimators are used and the minimum number of samples required to split an internal node is set as 10. (3) Logistic Regression (LR): a LR classifier is also learned for classification; (4) Deep neural network (DNN): we use two fully connected layers with the 1024 and 512 neurons respectively. We use 10-fold cross validation to estimate the key parameters of each method.

The results are compared in Fig. 4. The mean accuracy of each method is reported. We observe that on Hairstyle30k, SVM can beat the other three methods and achieves the accuracy of 29.1%. This reveals the challenging nature of our dataset. Note that the second best result 26.8% is obtained by DNN method, partly due to the insufficient training instances for some rare types of hairstyles as shown in Fig. 2. In contrast, the male hairstyle dataset has 6 types of hairstyles, and each hairstyle averagely has more than 3000 images. Thus on this dataset, the DNN achieves the best results on Male hairstyle dataset and outperforms the SVM results by 2.5%.

5.3 Hairstyle generation

Competitors. We compare various open-source methods on this task, including VAEGAN [14], and Attrb2img [39]. Attrb2img is an advanced version of CVAE. To make the results more comparable, all the methods are trained with the same experimental settings. We conduct the generation of hairstyles on both the CelebA and male hairstyle datasets.

The results of inception scores on CelebA and Male hairstyle dataset are compared and shown in Fig. 2. Both the inception scores of the generated and reconstructed images are compared. Totally 3000 images are generated for each method. Salimans *et al.* [27] proposed the inception score for evaluating image generation quality. Higher inception scores indicate better visual quality of samples generated. To make a fair comparison, we make all three methods to generate the same hairstyle.

As we can see from Table 2, our H-GAN achieves a 2.28 inception score, outperforming the VAEGAN on both datasets. This validates that images generated using our H-GAN have better visual quality than those of VAEGAN. Table 2 also shows that the inception score of Attrb2img is marginally higher than ours. After a close inspection of the qualitative results generated on CelebA (see Fig. 7) and Male dataset (see Fig. 5), we conclude that our H-GAN's results are still better than those of Attrb2img in term of the resolution, and clarity of generated hairs. Particularly, the images generated by Attrb2img has very blurred hairstyles but sharp human faces (which have contributed to the high inception scores).

Qualitative results. Some qualitative examples of the generated images of VAEGAN [14], Attrb2img [39] and H-GAN are illustrated in Fig. 7 and Fig. 5 for CelebA and Male hairstyle respectively. The generation results of Attrb2img show again sharp human faces and yet blurred hairstyles. The VAEGAN can generated hairstyles with fine details. Nevertheless, the overall quality of generated images of VAEGAN is worse than our H-GAN. For example, the second image of the VAEGAN results contains a very distorted face.

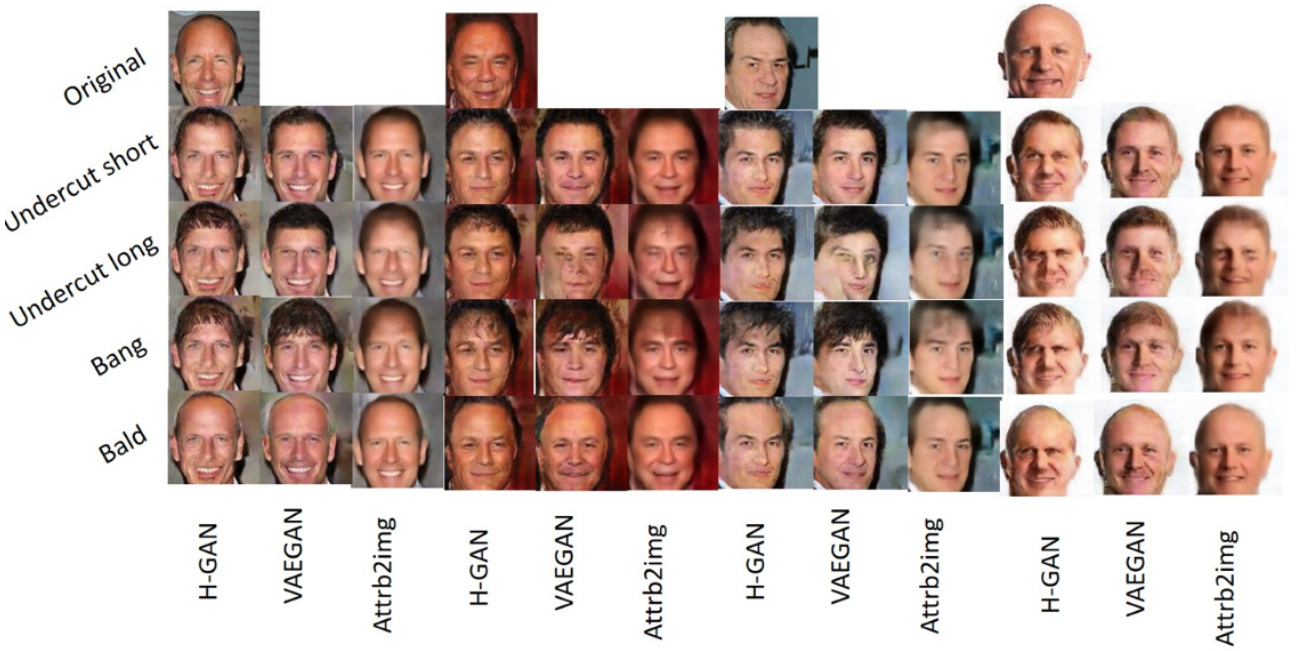

Figure 6: Modification of hairstyles on Male hairstyle dataset.

Figure 7: Qualitative results of the generation on CelebA dataset.

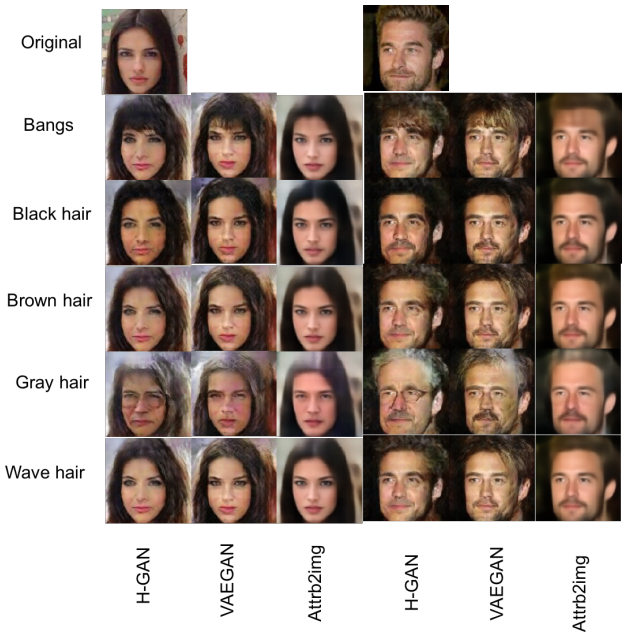

Figure 8: Results of modifying hairstyles on the CelebA dataset. Each column indicates modifying the hairstyles for one method.

5.4 Modifying hairstyles

Competitors. We compare various open source methods on editing hairstyles, including Attrib2img [39], and VAEGAN [14]. Though Attrb2img is not designed for modifying the images, we can actually employ the "attribute-conditioned image progression" strategy suggested in [39] to interpolate one type of hairstyle gradually changing the values along the dimension of hairstyle. The same experimental settings are used for all the experiments.

User-study experiments. Since the task of modifying hairstyles is essentially only modifying some parts of images, we notice that the modified images often have almost similar visual quality and thus the inception score used in generation task is not suitable as the evaluation metrics any more in this task. Instead, a user study as suggested in [40] is employed to compare these methods. Particularly, ten participants unaware of the project goals are invited for the user study. Given each image and one target hairstyle, these three methods are utilized to modify the hairstyle of images and we can obtain two images. Totally 100 images are randomly sampled from the CelebA and Male hairstyle dataset respectively, and we

Metric	Saliency	Quality	Similarity	Guess
Attrb2img	3.40	4.40	4.50	40.0%
VAEGAN	4.15	4.23	4.12	65.0%
H-GAN	4.40	4.40	4.65	70.0%

Table 3: The user-study of modification of user-defined attributes. The "Guess" results are reported as the accuracy of guessing.

change the hairstyles of sampled images. The participants have been required to compare the modified images with the original image and give their judgment on a five-point scale system from the less to the most likely by answering the following questions? (1) *Saliency*: how salient is the hairstyle has been changed in the image? (2) *Quality*: How is the overall quality of image modified? (3) *Identity*: how much degree do you think the modified image and original image is of the same person? (4) *Guess*: Additionally, a guessing game is introduced as the fourth metrics. Given a pair of modified and original images, the participants will be asked to guess which hairstyle has been modified from the four candidate choices, which are randomly sampled from the hairstyle names and of course, the correct hairstyle name should be included.

Quantitative results. We list the user-study results in Tab. 3. On all metrics, our H-GAN beats the other compared methods. Thus we can draw the conclusion that our H-GAN can more effectively modify the hairstyles whilst keeping the person's identity. Interestingly, even though Attrb2img has relative good visual quality, the strategy of modifying hairstyle employed by Attrb2img is relative less efficient and the scores "Guess" is significantly lower than the other two methods.

Qualitative results. Some visualization results are compared in Fig. 8 and Fig. 6. Each row is corresponding to one type of hairstyle. Each column indicates the results of one method. We highlight that in general, the results of our H-GAN are always better than or at least comparable to those of the other two methods. The results of Attrb2img still suffer from the problem of very blurred hair. We also notice that in Fig. 8, the hairstyle of Gray hair is highly correlated with the "age" and "eyeglasses" attribute in CelebA, since usually the senior person may wear glasses and have gray hair.

6 CONCLUSION

In this paper, we aim to present a comprehensive study on various hairstyle-related problems including classification, generation and modification of hairstyles. To promote the study of this topical issue, we introduce a new large-scale hairstyle dataset – Hairstyle30k with extensive hairstyle annotation. To automatically generate and change the hairstyle, we also propose a new – H-GAN model. Extensive experiments on several benchmark datasets had validated the effectiveness of the proposed H-GAN over the existing methods.

7 ACKNOWLEDGMENT

This work was supported in part by two NSFC projects (#61572138 and#61572134), one project from STCSM (#16JC1420401) and Shanghai Sailing Program (#17YF1427500)

REFERENCES

[1] Abrar H Abdulnabi, Gang Wang, Jiwen Lu, and Kui Jia. Multi-task cnn model for attribute prediction. *IEEE TMM*, 2015.
[2] D. Anguelov, K.-C. Lee, S. B. Gokturk, and B. Sumengen. Contextual identity recognition in personal photo albums. In *CVPR*, 2007.
[3] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. Video snapcut: Robust video object cutout using localized classifiers. In *ACM SIGGRAPH*, 2009.
[4] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jeibo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *ICML*, 2016.
[6] Ankur Datta, Rogerio Feris, and Daniel Vaquero. Hierarchical ranking of facial attributes. In *IEEE FG*, 2011.
[7] D.Guo and T.Sim. Digital face makeup by example. In *CVPR*, 2009.
[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, DavidWarde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.
[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.
[10] I. Kemelmacher-Shlizerman, S. Suwajanakorn, and S. M. Seitz. Illumination-aware age progression. In *CVPR*, 2014.
[11] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.
[12] Diederik P. Kingma and MaxWelling. Auto-encoding variational bayes. In *ICLR*, 2014.
[13] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.
[14] Anders Boesen Lindbo Larsen, Soren Kaae Sonderby, Hugo Larochelle, and OleWinther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016.
[15] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014.
[16] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. In *ACM SIGGRAPH*, 2004.
[17] Xiaodan Liang, Liang Lin, Wei Yang, Ping Luo, Junshi Huang, and Shuicheng Yan. Clothes co-parsing via joint image segmentation and labelingwith application to clothing retrieval. *IEEE TMM*, 2016.
[18] L. Liu, J. Xing, S. Liu, H. Xu, X. Zhou, and S. Yan. Wow! you are so beautiful today! *ACM TMCCA*, 2014.
[19] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015.
[20] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 2004.
[21] L. Ma, X. Yang, and D. Tao. Person re-identification over camera networks using multi-task distance metric learning. In *IEEE TIP*, 2014.
[22] Yuki Nagai, Kuniko Ushiro, Yoshiro Matsunami, Tsuyoshi Hashimoto, Yuusuke Kojima, and Weniger. Hairstyle suggesting system, hairstyle suggesting method, and computer program product. In *US Patent US20050251463*, 2005.
[23] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI*, 2002.
[24] Guim Perarnau, Joost van deWeijer, Bogdan Raducanu, and Jose M. Álvarez. Invertible conditional gans for image editing. In *NIPS Workshop on Adversarial Training*, 2016.
[25] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics*, 2004.
[26] Ethan M. Rudd, Manuel Gunther, and Terrance E. Boult. Moon:a mixed objective optimization network for the recognition of facial attributes. In *ECCV*, 2016.
[27] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *nips*, 2016.
[28] Wei Shen and Rujie Liu. Learning residual images for face attribute manipulation. In *arxiv*, 2017.
[29] Behjat Siddiquie, Rogerio Feris, and Larry Davis. Image ranking and retrieval based on multi-attribute queries. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
[30] Kihyuk Sohn, Xinchen Yan, and Honglak Lee. Learning structured output representation using deep conditional generative models. In *NIPS*, 2016.
[31] Xiao T, W.Ouyang, H. Li, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016.
[32] Wai-Shun Tong, Chi-Keung Tang, Michael S. Brown, and Ying-Qing Xu. example-based cosmetic transfer. In *FG*, 2007.
[33] A. Torralba, R. Fergus, and W.T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE TPAMI*, 2008.

[34] D.A. Vaquero, R.S. Feris, D. Tran, L. Brown, A. Hampapur, and M. Turk. Attribute-based people search in surveillance environments. In *IEEE WACV*, 2009.
[35] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
[36] Jing Wang, Yu Cheng, and Rogerio Feris. Walk and learn: Facial attribute representation learning from egocentric video and contextual data. *CVPR*, 2016.
[37] Xianwang Wang and Tong Zhang. Clothes search in consumer photos via color matching and attribute learning. In *ACM International Conference on Multimedia*, 2011.
[38] Chih-Yu Yan, Ming-Chun Tien, and Ja-Ling Wu. Interactive background blurring. In *ACM MM*, 2009.
[39] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. attribute2image: conditional image generation from visual attributes. In *ECCV*, 2016.
[40] Weidong Yin, Yanwei Fu, Leonid Sigal, and Xiangyang Xue. Semi-latent gan: Learning to generate and modify facial images from attributes. In *arxiv*, 2017.
[41] Bo Zhao, Xiao Wu, Qiang Peng, and Shuicheng Yan. Clothing cosegmentation for shopping images with cluttered background. *IEEE TMM*, 2016.
[42] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, and Qi Tian. Person re-identification in the wild. *arXiv preprint arXiv:1604.02531*, 2016.
[43] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. View synthesis by appearance flow. In *ECCV*, 2016.
[44] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016.