Semi-Supervised Learning for Face Sketch Synthesis in the Wild

Chaofeng Chen[1], Wei Liu[1], Xiao Tan[2] and Kwan-Yee K. Wong[1]

[1]The University of Hong Kong, [2]Baidu Research
{cfchen, wliu, kykwong}@cs.hku.hk, tanxchong@gmail.com

Abstract. Face sketch synthesis has made great progress in the past few years. Recent methods based on deep neural networks are able to generate high quality sketches from face photos. However, due to the lack of training data (photo-sketch pairs), none of such deep learning based methods can be applied successfully to face photos in the wild. In this paper, we propose a semi-supervised deep learning architecture which extends face sketch synthesis to handle face photos in the wild by exploiting additional face photos in training. Instead of supervising the network with ground truth sketches, we first perform patch matching in feature space between the input photo and photos in a small reference set of photo-sketch pairs. We then compose a *pseudo sketch feature* representation using the corresponding sketch feature patches to supervise our network. With the proposed approach, we can train our networks using a small reference set of photo-sketch pairs together with a large face photo dataset without ground truth sketches. Experiments show that our method achieve state-of-the-art performance both on public benchmarks and face photos in the wild. Codes are available at https://github.com/chaofengc/Face-Sketch-Wild.

1 Introduction

Face sketch synthesis targets at generating a sketch from an input face photo. It has many useful applications. For instance, police officers often have to rely on face sketches to identify suspects, and face sketch synthesis makes it feasible for matching sketches against photos in a mugshot database automatically. Artists can also employ face sketch synthesis to simplify the animation production process [1]. Many people prefer using sketches as their profile pictures in social media networks [2], and face sketch synthesis allows them to produce sketches without the help of a professional artist.

Much effort has been devoted to face sketch synthesis. In particular, exemplar based methods dominated in the past two decades. These methods can achieve good performance without explicitly modeling the highly nonlinear mapping between face photos and sketches. They commonly subdivide a test photo into overlapping patches, and match these test patches with the photo patches in a reference set of photo-sketch pairs. They then compose an output sketch using the corresponding sketch patches in the reference set. Although promising results have been reported [1,3,4,5], these methods have several drawbacks. For example, sketches in Fig. 4(c)(d)(e)(f) are over-smoothed and fail to preserve subtle contents such as strands of hair on the forehead. Moreover, the patch matching and optimization processes are often very time-consuming. Recent methods exploited Convolutional Neural Networks (CNNs) to learn a direct mapping between photos and sketches, which is, however, a non-trivial task. The straight forward CNN based method produces blurry sketches (see Fig. 4(g)), and methods based on Generative Adversary Networks (GAN) [6] introduces undesirable artifacts (see Fig. 4(h),(i)). Besides, all these CNN based methods do not generalize well to face photos in the wild due to the lack of large training datasets of photo-sketch pairs. Although unpaired GAN based methods such as Cycle-GAN [7] can use unpaired data to transfer images between different domains, they fail to well preserve the facial content because of the weak content constraint (see fig. 8).

In this paper, we propose a semi-supervised learning framework for face sketch synthesis that takes advantages of the exemplar based approach, the perceptual loss and GAN. We design a residual net [8] with skip connections as our generator network. Suppose we have a small reference set of photo-sketch pairs and a large face photo dataset without ground truth sketches. Similar to the exemplar based approach, we subdivide the VGG-19 [9] feature maps of the input photo into overlapping patches, and match them with the photo patches (in feature space) in the reference set. We then compose a *pseudo sketch feature* representation using the corresponding sketch patches (in feature space) in the reference set. We can then supervise our generator network using a perceptual loss based on the mean squared error (MSE) between the feature maps of the generated sketch and the corresponding pseudo sketch feature of the input photo. An adversary loss is also utilized to make the generated sketches more realistic.

In summary, our main contributions are three folds: (1) A semi-supervised learning framework for face sketch synthesis. Our framework allows us to train our networks using a small reference set of photo-sketch pairs together with a large face photo dataset without ground truth sketches. This enables our networks to generalize well to face photos in the wild. (2) A perceptual loss based on pseudo sketch feature. We show that the proposed loss is critical in preserving both facial content and texture details in the generated sketches. Extensive experiments are conducted to verify the effectiveness of our model. Both qualitative and quantitative results illustrate the superiority of our method. (3) To the best of our knowledge, our method is the first work that can generate visually pleasant sketches for face photos in the wild.

2 Related Works

2.1 Exemplar Based Methods

Tang and Wang [10] first introduced the exemplar based method based on eigentransformation. They projected an input photo onto the eignspace of the training photos, and then reconstruct a sketch from the eignspace of the training sketches using the same projection. Liu *et al.* [11] observed that the linear model holds better locally, and therefore proposed a nonlinear model based on local linear embedding (LLE). They first subdivided an input photo into overlapping patches and reconstructed each photo patch as a linear combination of the training photo patches. They then obtained the sketch patches by applying the same linear combinations to the corresponding training sketch patches. Wang and Tang [5] employed a multi-scale markov random fields (MRF) model to improve the consistency between neighboring patches. By introducing shape priors and SIFT features, Zhang *et al.* [12] proposed an extended version of MRF which can handle face photos under different illuminations and poses. However, these MRF based methods are not capable of synthesizing new sketch patches since they only select the best candidate sketch patch for each photo patch. To tackle this problem, Zhou *et al.* [3] presented the markov weight fields (MWF) model which produces a target sketch patch as a linear combination of K best candidate sketch patches. Considering that patch matching based on traditional image features (e.g., PCA and SIFT) is not robust, a recent method [4] used CNN feature to represent the training patches and computed more accurate combination coefficients. To accelerate the synthesis procedure, Song *et al.* [1] formulated face sketch synthesis as a spatial sketch denoising (SSD) problem, and Wang *et al.* [13] presented an offline random sampling strategy for nearest neighbor selection of patches.

2.2 Learning Based Methods

Recent works applied CNN to synthesize sketches and produced promising results. Zhang *et al.* [14] proposed a 7-layer fully convolutional network (FCN) to directly transfer an input photo to a sketch. Although their model can roughly estimate the outline of a face, it fails to capture texture details with the use of intensity based mean square error (MSE) loss. Zhang *et al.* [15] utilized a branched fully convolutional network (BFCN) consisting of a content branch and a texture branch. Because the face content and texture are predicted separately with different loss metrics, the final sketch looks disunited. Chen *et al.* [16] proposed the pyramid column feature and used it to compose a reference style for a test photo from the training sketches. They utilized a CNN to create a content image from the photo, and then transferred the reference style to introduce shadings and textures in the output sketch. Wang *et al.* [17] presented the multi-scale generative adversarial networks (GANs) to generate sketches from photos and vice versa. Multiple discriminators at different hidden layers are used to supervise the synthesis process. Gao *et al.* [18] took advantage of the facial parsing map and proposed a composition-aided stack GAN. All these deep learning based methods require ground truth photo-sketch pairs for training, and they do not generalize well to face photos in the wild due to the lack of training data.

3 Semi-Supervised Face Sketch Synthesis

3.1 Overview

Our framework is composed of three main parts, namely a generator network G, a pseudo sketch feature generator and a discriminator network D (see Fig. 1). The generator network is a deep residual network with skip connections. It is used to generate a synthesized sketch $\hat{\mathbf{y}}$ for each input photo $\mathbf{x}$. The pseudo sketch feature generator is the key to our semi-supervised learning approach. Instead of training the generator network directly with ground truth sketches, we construct a pseudo sketch feature for each input photo to supervise the synthesis of $\hat{\mathbf{y}}$. In this way, we can train our network on any face photo datasets, and generalize our model to face photos in the wild. We further adopt a discriminator network D to minimize the gap between generated sketches and real sketches drawn by artists.

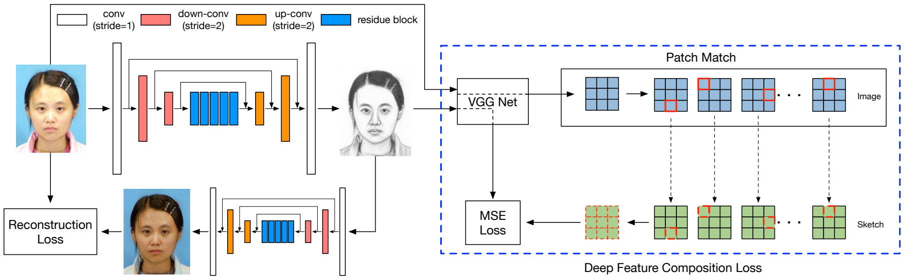


Fig. 1: Framework of the proposed method. The generator network is a deep residual network with skip connections. It generates a synthesized sketch from an input photo. The pseudo sketch feature generator utilizes patch matching in the deep feature space to generate a pseudo sketch feature for an input photo in training. The discriminator network tries to distinguish between generated sketches and sketches drawn by artists.

3.2 Pseudo Sketch Feature Generator

Given a reference set $\mathcal{R} = \{(\mathbf{x}_i^{\mathcal{R}}, \mathbf{y}_i^{\mathcal{R}})\}_{i=1}^{N}$, the pseudo sketch feature generator targets at constructing a pseudo sketch feature $\Phi'(\mathbf{x})$ for a test photo $\mathbf{x}$ which is used to supervise the synthesis of the sketch $\hat{\mathbf{y}}$. We follow MRF-CNN [19] to extract a local patch representation of an image. We first feed $\mathbf{x}$ into a pretrained VGG-19 network and extract the feature map $\Phi^l(\mathbf{x})$ at the l-th layer. Similarly, we obtain $\{\Phi^l(\mathbf{x}_i^{\mathcal{R}})\}_{i=1}^{N}$ and $\{\Phi^l(\mathbf{y}_i^{\mathcal{R}})\}_{i=1}^{N}$. Let us denote a $k \times k$ patch centered at a point j of $\Phi^l(\mathbf{x})$ as $\Psi_j\left(\Phi^l(\mathbf{x})\right)$, and the same definition applies to $\Psi_j\left(\Phi^l(\mathbf{x}_i^{\mathcal{R}})\right)$

and $\Psi_j(\Phi^l\left(\mathbf{y}_i^{\mathcal{R}}\right))$. Now for each patch $\Psi_j\left(\Phi^l(\mathbf{x})\right)$, where $j = 1, 2, \ldots, m$ and $m = (H^l - 2 \times \lfloor \frac{k}{2} \rfloor) \times (W^l - 2 \times \lfloor \frac{k}{2} \rfloor)$ with H^l and W^l being the height and width of $\Phi^l(\mathbf{x})$, we find its best match $\Psi_{j'}\left(\Phi^l(\mathbf{x}_{i'}^{\mathcal{R}})\right)$ in the reference set based on cosine distance, i.e.,

$$(i', j') = \underset{\substack{i^*=1\sim N \\ j^*=1\sim m}}{\arg\max} \frac{\Psi_j\left(\Phi^l(\mathbf{x})\right) \cdot \Psi_{j^*}\left(\Phi^l(\mathbf{x}_{i^*}^{\mathcal{R}})\right)}{\left\|\Psi_j\left(\Phi^l(\mathbf{x})\right)\right\|_2 \left\|\Psi_{j^*}\left(\Phi^l(\mathbf{x}_{i^*}^{\mathcal{R}})\right)\right\|_2}. \tag{1}$$

Since the photos and the corresponding sketches in $\mathcal{R}$ are well aligned, we directly apply (i', j') to index the corresponding sketch feature patch $\Psi_{j'}\left(\Phi^l(\mathbf{y}_{i'}^{\mathcal{R}})\right)$ for $\Psi_{j'}\left(\Phi^l(\mathbf{x}_{i'}^{\mathcal{R}})\right)$, and use it as the pseudo sketch feature patch $\Psi'_j\left(\Phi^l(\mathbf{x})\right)$ for $\Psi_j\left(\Phi^l(\mathbf{x})\right)$. Finally, a pseudo sketch feature representation (at layer l) for $\mathbf{x}$ is given by $\{\Psi'_j\left(\Phi^l(\mathbf{x})\right)\}_{j=1}^m$. Fig. 2 visualizes an example of the pseudo sketch feature. It can be seen that the pseudo sketch feature provides a good approximation of the real sketch feature (see Fig. 2(a)). We also show a naïve reconstruction in Fig. 2(b) obtained by directly using the matching index to index the pixel values in the training sketches. We can see such a naïve reconstruction does roughly resemble the real sketch, which also justifies the effectiveness of the pseudo sketch feature. Note that we only need alignment between photos and sketches in $\mathcal{R}$. Since we perform a dense patch matching between the input photo and the reference photos, we can also generate reasonable pseudo sketch features for input faces under different poses (see Fig. 2(c)).

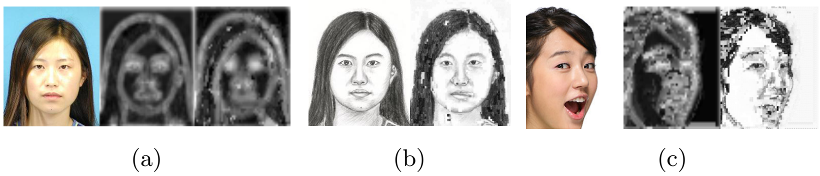

(a) (b) (c)

Fig. 2: (a) Ground truth sketch feature (middle) and pseudo sketch feature of the relu3_1 layer (right). (b) Ground truth sketch (left) and pixel level projection of the patch matching result (right). (c) Photos in the wild without ground truth sketches. (*Note that the pixel level results are only for visualization, and they are not used in training.*)

3.3 Loss Functions

Pseudo Sketch Feature Loss We define our pseudo sketch feature loss as

$$L_p(\mathbf{x}, \hat{\mathbf{y}}) = \sum_{l=3}^{5} \sum_{j=1}^{m} \left\|\Psi_j\left(\Phi^l(\hat{\mathbf{y}})\right) - \Psi'_j\left(\Phi^l(\mathbf{x})\right)\right\|_2^2, \tag{2}$$

where $l = 3, 4, 5$ refer to layers relu3_1, relu4_1, and relu5_1 respectively. High level features after relu3_1 are better representations of textures and more robust to appearance changes and geometric transforms [19]. Fig. 3 shows the results of using different layers in L_p. As expected, low level features (*e.g.*, relu1_1 and relu2_1) fail to generate sketch textures. While high level features (*e.g.*, relu5_1) can better preserve textures, they produce artifacts in terms of details (see the eyes of sketches in Fig. 3). To get better performance and reduce the computation cost of patch matching, we set $l = 3, 4, 5$.

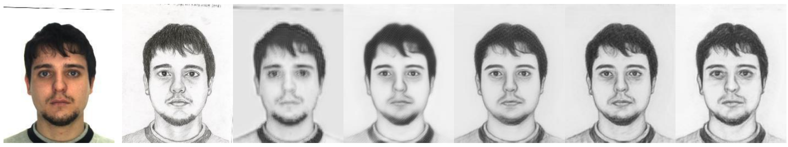


Fig. 3: Results of using different layers in pseudo sketch feature loss.

GAN Loss For easier convergence, we use the least square loss when training the GAN, known as LSGAN [20]. The objective functions of LSGAN are given by

$$L_{GAN\_D} = \frac{1}{2}\mathbb{E}_{y\sim p_{sketch}(y)}[(D(y)-1)^2] + \frac{1}{2}\mathbb{E}_{x\sim p_{photo}(x)}[(D(G(x)))^2] \quad (3)$$

$$L_{GAN\_G} = \mathbb{E}_{x\sim p_{photo}(x)}[(D(G(x))-1)^2] \quad (4)$$

Total Variation Loss Sketches generated by CNN may be unnatural and noisy. Following previous works [21,19,22], we adopt the *total variation loss* as a natural image prior to further improve the sketch quality,

$$L_{tv}(\hat{\mathbf{y}}) = \sum_{m,n}\left((\hat{\mathbf{y}}_{m+1,n} - \hat{\mathbf{y}}_{m,n})^2 + (\hat{\mathbf{y}}_{m,n+1} - \hat{\mathbf{y}}_{m,n})^2\right), \quad (5)$$

where $\hat{\mathbf{y}}_{m,n}$ denotes the intensity value at (m, n) of the synthesized sketch $\hat{\mathbf{y}}$.

Based on the above loss terms, we can train our generator network G and discriminator network D using the following two loss functions respectively:

$$L_G = \lambda_p L_p + \lambda_{adv} L_{GAN\_G} + \lambda_{tv} L_{tv}, \quad (6)$$

$$L_D = L_{GAN\_D} \quad (7)$$

where L_G and L_D are minimized alternatively until converge. λ_p, λ_{adv} and λ_{tv} are trade-off weights for each loss term respectively.

4 Implementation Details

4.1 Datasets

Photo-Sketch Pairs We use two public datasets: the CUFS dataset(consist of the CUHK student dataset [10], the AR dataset [23], and the XM2VTS dataset [24]) and the CUFSF dataset [25], to evaluate our model[1]. The CUFSF dataset is more challenging than the CUFS dataset because (1) the photos were captured under different lighting conditions and (2) the sketches exhibit strong deformation in shape and cannot be aligned with the photos well. Details of these datasets are summarized in Table 1.

Table 1: Details of benchmark datasets. Align: whether the sketches are well aligned with photos. Var: whether the photos have lighting variations.

Dataset		Total Pairs	Train	Test	Align	Var
CUFS	CUHK	188	88	100	✓	✗
	AR	123	80	43	✓	✗
	XM2VTS	295	100	195	✗	✗
CUFSF		1194	250	944	✗	✓

Face Photos We use the VGG-Face dataset [26] to evaluate our model on photos in the wild. There are 2,622 persons in this dataset and each person has 1,000 photos. We randomly select 2,000 persons for training and the rest for testing. For each person in the training split, we randomly select $\mathcal{N}$ photos and named the resulting dataset VGG-Face$\mathcal{N}$[2], where $\mathcal{N} = 01, 02, \ldots, 10$. We also randomly select 2 photos for each person in the testing split to construct a VGG test set of 1,244 photos.

Preprocessing For photos/sketches which have already been aligned and have a size of 250×200, we leave them unchanged. For the rest, we first detect 68 face landmarks on the image using `dlib`[3], and calculate a similarity transform to warp the image into one with the two eyes located at $(75, 125)$ and $(125, 125)$ respectively. We then crop the resulting image to a size of 250×200. We simply drop those photos/sketches from which we fail to detect face landmarks.

4.2 Patch Matching

As in exemplar based methods, patch matching is a time-consuming process. We accelerate this process in three ways. First, we precompute and store the feature

[1] Data comes from `http://www.ihitworld.com/RSLCR.html`

[2] The dataset will be made available.

[3] `http://dlib.net/`

patches for the photos and sketches in the reference set (i.e., $\{\Psi_j\left(\Phi^l(\mathbf{x}_i^{\mathcal{R}})\right)\}$ and $\{\Psi_j(\Phi^l\left(\mathbf{y}_i^{\mathcal{R}})\right)\}$). Second, instead of searching the whole reference feature set, we first identify k best matched reference photos for each input photo based on the cosine distance of their relu5_1 feature maps. Patch matching is then restricted within these k reference photos (we set $k = 5$ in the whole training process). Third, Equ. 1 is implemented as a convolution operator which can be computed efficiently on GPU.

4.3 Training Details

We update the generator and discriminator alternatively at every iteration. The trade-off weights λ_p, λ_{adv} are set to 1 and 10^3, and λ_{tv} is set to 10^{-5} when use CUFS as reference style and 10^{-2} when use CUFSF. We implemented our model using `PyTorch`[4], and trained it on a Nvidia Titan X GPU. We used Adam [27] with learning rates from 10^{-3} to 10^{-5}, decreasing with a factor of 10^{-1}. Data augmentation was done online in the color space (brightness, contrast, saturation and sharpness). Each iteration took about 2s with a batch size of 6, and the model converged after about 5 hours of training.

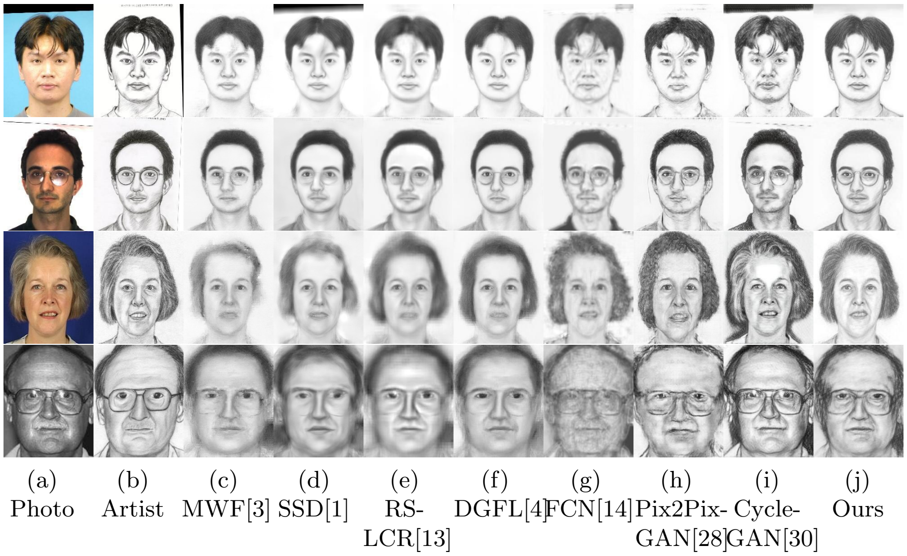
5 Evaluation on Public Benchmarks

In this section, we evaluate our model using two public benchmarks, namely CUFS and CUFSF, which were captured under laboratory conditions. We use the training photos from CUFS$\cup$CUFSF to train our networks. When evaluating on CUFS, the reference photo-sketch pairs only comes from CUFS, and the same applies to CUFSF. To demonstrate the effectiveness of our model, we compare our results both qualitatively and quantitatively with seven other methods, namely MWF [3], SSD [1], RSLCR [13], DGFL [4], FCN [14], Pix2Pix-GAN [28], and Cycle-GAN [7]. We also compare our results quantitatively with the latest GAN based sketch synthesis methods, i.e., PS^2-MAN [29] and stack-CA-GAN [18]. Since the models of their work are not available, we can only compare with the results that are directly taken from their published papers.

5.1 Qualitative Comparison

As we can observe in Fig. 4, exemplar based methods (see Fig. 4(c),(d),(e) in general perform worse than learning based methods (see Fig. 4(g),(h),(i),(j)), especially in preserving contents of the input photos. Using deep features in exemplar based methods helps to alleviate the problem, but the results are oversmoothed (see Fig. 4(f)). Due to the lack of training data, FCN produces bad results when the photos are taken under very different lighting conditions (see last two rows of Fig. 4(g)). Although the two GANs can produce much better results than FCN, they also introduce many artifacts and noise. Thanks to the

[4] `http://pytorch.org/`



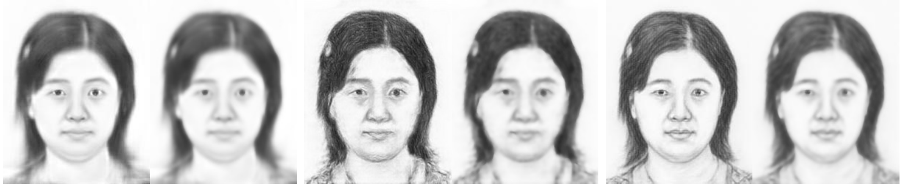
(a) Photo (b) Artist (c) MWF[3] (d) SSD[1] (e) RS-LCR[13] (f) DGFL[4] (g) FCN[14] (h) Pix2Pix-GAN[28] (i) Cycle-GAN[30] (j) Ours

Fig. 4: Sketches generated using different methods. First 3 rows: test photos from CUFS. Last row: test photo from CUFSF.

pseudo sketch feature loss, our method does not suffer from the above problems. In particular, our semi-supervised strategy allows us to incorporate more training photos without ground truth in training, which helps to improve the generalization ability.

5.2 Quantitative Comparison

Image Quality Assessment For datasets with ground truth sketches (e.g., CUFS and CUFSF), previous work [13,18,4] typically used structural similarity (SSIM) [31] as an image quality assessment metric to measure the similarity between a generated sketch and the ground truth sketch. However, many researchers (e.g., in super resolution [32] and face sketch synthesis [30,29]) pointed out that SSIM is not always consistent with the perceptual quality. One main reason is that SSIM favors slightly blurry images when the images contain rich textures. To demonstrate this, we show some sketches generated using different methods together with their SSIM scores in Fig. 5. It can be seen that sketch generated by RSLCR is smoother than those by Pix2Pix-GAN and our model, but have higher SSIM scores. We applied a bilateral filter to smooth all the sketches. It can be observed that the SSIM scores of the sketch generated by RSLCR remain roughly the same after smoothing, whereas those of the sketches generated by Pix2Pix-GAN and our model improve by more than 1.5%. In Fig. 6(a), we show the averaged SSIM scores of the sketches generated by different methods on CUFS, together with the averaged SSIM scores of their smoothed counterparts. As expected, the averaged SSIM scores of most of the



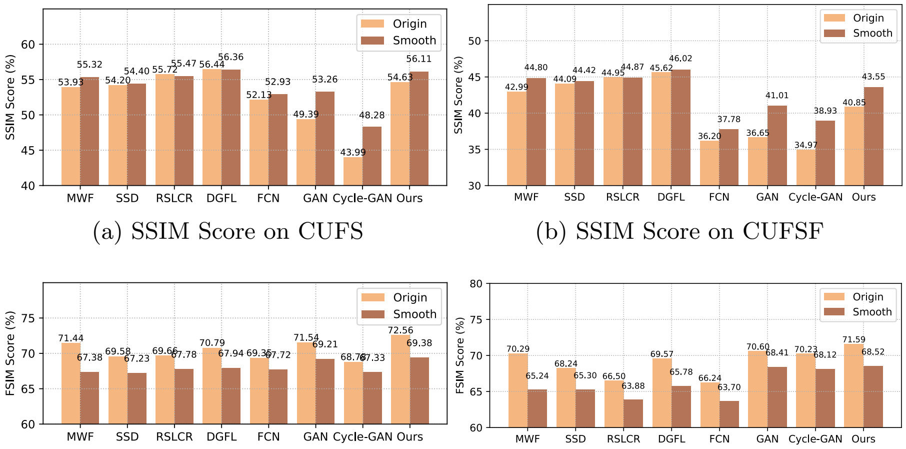
(b) RSLCR
SSIM: 0.5970/0.5903.
FSIM: 0.7488/0.7362.

(c) Pix2Pix-GAN
SSIM: 0.5648/0.5953.
FSIM: 0.7559/0.7506.

(d) Ours.
SSIM: 0.5814/0.6055.
FSIM: 0.7692/0.7557.

Fig. 5: SSIM and FSIM scores of some generated sketches (left) and their smoothed counterparts (right).

methods improve after smoothing, same for a few exemplar based methods which produce over-smoothed sketches. The averaged SSIM score of our smoothed results is comparable to that of the state-of-the-art method. In Fig. 6(b), we show the corresponding results on CUFSF. Similar conclusions can be drawn.



(a) SSIM Score on CUFS

(b) SSIM Score on CUFSF

(c) FSIM Score on CUFS

(d) FSIM Score on CUFSF

Fig. 6: Averaged SSIM and FSIM scores of the sketches generated by different methods on CUFS and CUFSF. The proposed method achieves state-of-the-art FSIM score on both datasets.

Due to the drawback of SSIM, we use feature similarity (FSIM) [33] as our image quality assessment metric. FSIM is better at evaluating detailed textures compared with SSIM. It can be observed from Fig. 5 that the FSIM scores of the sketches decrease after smoothing. The average FSIM scores of the sketches generated by different methods on CUFS and CUFSF are shown in Fig. 6(c) and Fig. 6(d) respectively. It can be seen that our method achieves the state-of-the-art in terms of FSIM score on both CUFS and CUFSF.

Face Sketch Recognition Sketch recognition is an important application of face sketch synthesis. We follow the same practice of Wang *et al.* [13] and employ the null-space linear discriminant analysis (NLDA) [34] to perform the recognition experiments. Fig. 7 shows the recognition accuracy of different methods on the two datasets. Our method achieves the best result when the dimension of the reduced eigenspace is less than 100, and achieves a competitive result to the state-of-the-art method [4] when the dimension is above 100.

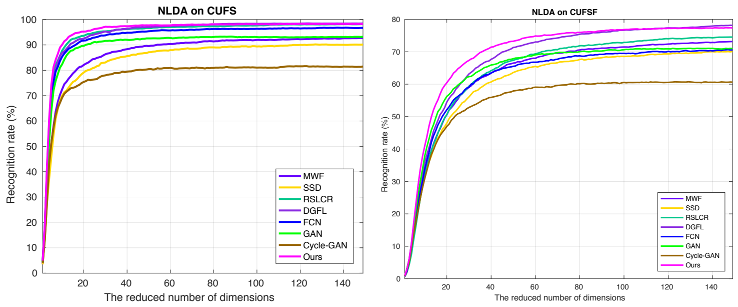
Comparison with PS^2-MAN and stack-CA-GAN To further demonstrate the effectiveness of the proposed method, we compare it with two latest GAN methods, namely PS^2-MAN [29] and stack-CA-GAN [18], which are specially designed for sketch synthesis. As shown in Table 2, our method achieves the best performance on almost all datasets, except for the SSIM score in CUFSF. However, we obtain a better performance on NLDA which indicates that our model can better preserve the identify information. Note that both of these GAN methods use extra information to train their network, i.e., multi-scale supervision (PS^2-MAN) and parsing map (stack-CA-GAN). Compared with them, our perceptual loss can not only avoid producing artifacts but also help to improve the generalization of the network.

Table 2: Quantitative comparison with PS^2-MAN and stack-CA-GAN. Results are taken from their original papers.

	CUHK		CUFS		CUFSF	
	SSIM	FSIM	SSIM	NLDA	SSIM	NLDA
PS^2-MAN	0.6156	0.7361	—		—	
stack-CA-GAN	—		0.5266	96.04	**0.4106**	77.31
Ours	**0.6328**	**0.7423**	**0.5463**	**98.22**	0.4085	**78.04**

6 Sketch Synthesis in the Wild

There are two challenges for sketch synthesis in the wild. The first challenge is how to deal with real photos captured under uncontrolled environments with varying pose and lighting, and cluttered backgrounds. The second is the computation time. Our method tackles the first challenge by introducing more training photos through the construction of pseudo sketch features. Regarding computation time, our CNN based model can generate a sketch in a single feed forward pass which takes about 7ms on a GPU for a 250 × 200 photo. We compare our



(a) NLDA score on CUFS (b) NLDA score on CUFSF

Fig. 7: Face recognition rate against feature dimensions on CUFS and CUFSF.

method with five other methods , including SSD[5], FCN, Pix2Pix-GAN[6], Cycle-GAN[7], and Fast-RSLCR[8]. In this experiment, we train our model using CUFS as the reference set and all the training photos from the CUFS, CUFSF and VGG-Face10 as the training set. Since there are no ground truth sketches for the test photos, we carry out a mean opinion score (MOS) test to quantitatively evaluate the results.

6.1 Qualitative Comparison

As photos in the wild are captured under uncontrolled environments, their appearance may vary largely. Fig. 8 shows some photos sampled from our VGG-Face test dataset and the sketches generated by different methods. It can be observed that these photos may show very different lightings, poses, image resolutions, and hair styles. Besides, some photos may be incomplete and people may also use a cartoon as their photos for entertainment (see the last row of Fig. 8). It is therefore very difficult, if not impossible, for a method which only learns from a small set of photo-sketch pairs to generate sketches for photos in the wild. Among the results of other methods, exemplar based methods (see Fig. 8(b)(c)) fail to deal with pose changes and different hair styles. FCN produces sketches (see Fig. 8(d)) that can roughly preserve the contour of the face but lose important facial components (e.g., nose and eyes). Although GANs can generate some sketch like textures, none of them can well preserve the contents. The face shapes are distorted and the key facial parts are lost. It can be seen from Fig. 8(g) that our model can handle photos in the wild well and generate pleasant results.

[5] `http://www.cs.cityu.edu.hk/~yibisong/eccv14/index.html`
[6] `https://github.com/phillipi/pix2pix`
[7] `https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix`
[8] `http://www.ihitworld.com/RSLCR.html`

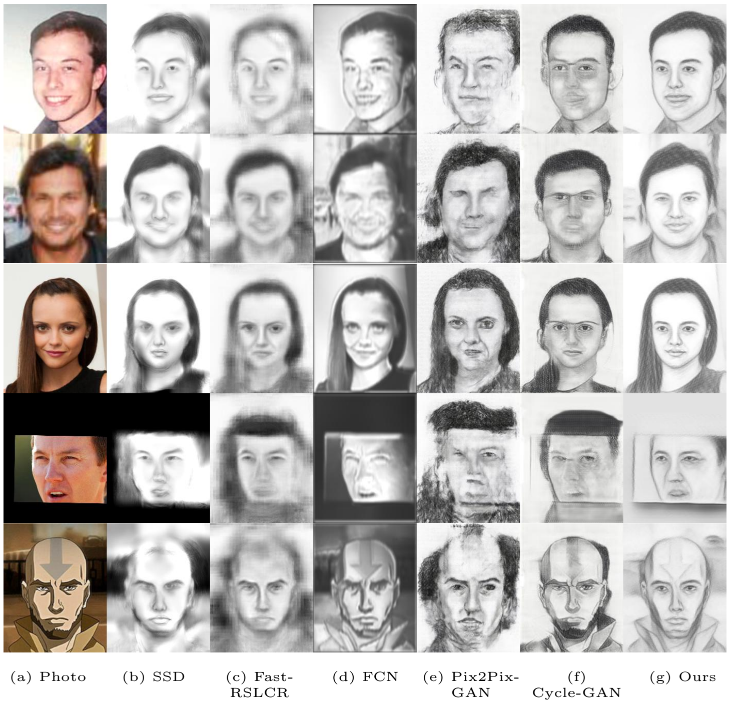


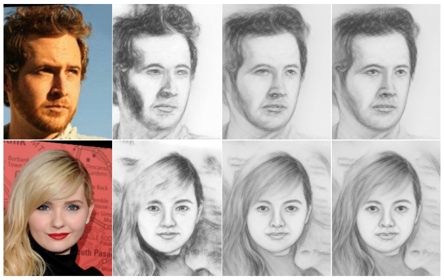
Fig. 8: Qualitative comparison of different methods for images in the wild. Benefit from the additional training photos, the proposed methods can deal with various photos.

6.2 Effectiveness of Additional Training Photos

Introducing more training photos from VGG-Face dataset is the key to improve the generalization ability of our model. As demonstrated in Fig. 9, the model trained without additional photos from VGG-Face has difficulty in handling uncontrolled lightings and different hair colors (see Fig. 9(b)). As we add more photos to the training set, the results improve significantly (see the eyes and hair in Fig. 9).

6.3 Mean Opinion Score Test

Since there are no ground truth sketches for the photos in the wild, we performed a MOS test to assess the perceptual quality of the sketches generated by different methods. Specifically, we randomly selected 30 photos from the VGG test set, and then generated the sketches for these photos using SSD, FCN, Fast-RSLCR, Pix2Pix-GAN and our method respectively. Given the example photo-sketch pairs from public benchmarks as reference, 108 raters were asked to rank 10 groups of randomly selected sketches synthesized by the five different methods.



(a) Photo (b) VGGFace00 (c) VGGFace05 (d) VGGFace10

Fig. 9: Effectiveness of additional training photos. The results improve a lot when more and more photos are added to the training set.

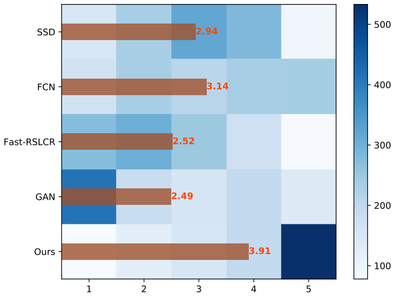


Fig. 10: Results of MOS test on the quality of sketches generated by SSD, FCN, Fast-RSLCR, Pix2Pix-GAN and our model on photos in the wild.

We assign a score of 1-5 to the sketches based on their rankings (5 being the best). The results are presented in Fig. 10. It can be observed that the MOS of our results significantly outperforms that of the other methods. This demonstrates the superiority of our method on photos in the wild.

7 Conclusion

In this paper, we propose a semi-supervised learning framework for face sketch synthesis in the wild. We design a residual network with skip connections to transfer photos to sketches. Instead of supervising our network using ground truth sketches, we construct a novel pseudo sketch feature representation for each input photo based on feature space patch matching with a small reference set of photo-sketch pairs. This allows us to train our model using a large face photo dataset (without ground truth sketches) with the help of a small reference set of photo-sketch pairs. Training with a large face photo dataset enables our model to generalize better to photos in the wild. Experiments show that our method can produce sketches comparable to those produced by other state-of-the-art methods on four public benchmarks (in terms of SSIM and FSIM), and outperforms them on photos in the wild.

8 Acknowledgment

We thank Nannan Wang, Hao Zhou and Yibing Song for providing their codes and data. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

1. Song, Y., Bao, L., Yang, Q., Yang, M.H.: Real-time exemplar-based face sketch synthesis. In: European Conference on Computer Vision. (2014) 800–813
2. Berger, I., Shamir, A., Mahler, M., Carter, E., Hodgins, J.: Style and abstraction in portrait sketching. ACM Transactions on Graphics (TOG) **32** (2013) 55
3. Zhou, H., Kuang, Z., Wong, K.Y.K.: Markov weight fields for face sketch synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition. (2012) 1091–1097
4. Zhu, M., Wang, N., Gao, X., Li, J.: Deep graphical feature learning for face sketch synthesis. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. (2017) 3574–3580
5. Wang, X., Tang, X.: Face photo-sketch synthesis and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009) 1955–1967
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680
7. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networkss. In: Computer Vision (ICCV), 2017 IEEE International Conference on. (2017)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv:1512.03385 (2015)
9. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
10. Tang, X., Wang, X.: Face sketch synthesis and recognition. In: IEEE International Conference on Computer Vision. (2003) 687–694
11. Liu, Q., Tang, X., Jin, H., Lu, H., Ma, S.: A nonlinear approach for face sketch synthesis and recognition. In: IEEE Conference on Computer Vision and Pattern recognition. Volume 1. (2005) 1005–1010
12. Zhang, W., Wang, X., Tang, X.: Lighting and pose robust face sketch synthesis. In: European Conference on Computer Vision. (2010) 420–433
13. Wang, N., Gao, X., Li, J.: Random sampling for fast face sketch synthesis. arXiv:1701.01911 (2017)
14. Zhang, L., Lin, L., Wu, X., Ding, S., Zhang, L.: End-to-end photo-sketch generation via fully convolutional representation learning. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (ICMR). (2015) 627–634
15. Zhang, D., Lin, L., Chen, T., Wu, X., Tan, W., Izquierdo, E.: Content-adaptive sketch portrait generation by decompositional representation learning. IEEE Transactions on Image Processing (TIP) **26** (2017) 328–339
16. Chen, C., Tan, X., , Wong, K.Y.K.: Face sketch synthesis with style transfer using pyramid column feature. IEEE Winter Conference on Applications of Computer Vision (2018)
17. Wang, N., Zhu, M., Li, J., Song, B., Li, Z.: Data-driven vs. model-driven: Fast face sketch synthesis. Neurocomputing (2017)
18. Gao, F., Shi, S., Yu, J., Huang, Q.: Composition-aided sketch-realistic portrait generation. arXiv:1712.00899 (2017)
19. Li, C., Wand, M.: Combining markov random fields and convolutional neural networks for image synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2479–2486

20. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE (2017) 2813–2821
21. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, Springer (2016) 694–711
22. Kaur, P., Zhang, H., Dana, K.J.: Photo-realistic facial texture transfer. arXiv:1706.04306 (2017)
23. Martinez, A., benavente., R.: The AR face database. Technical report, CVC Tech. Report (1998)
24. Messer, K., Matas, J., Kittler, J., Jonsson, K.: Xm2vtsdb: The extended m2vts database. In: In Second International Conference on Audio and Video-based Biometric Person Authentication. (1999) 72–77
25. Zhang, W., Wang, X., Tang, X.: Coupled information-theoretic encoding for face photo-sketch recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2011) 513–520
26. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Conference. (2015)
27. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
28. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. CVPR (2017)
29. Wang, L., Sindagi, V.A., Patel, V.M.: High-quality facial photo-sketch synthesis using multi-adversarial networks. arXiv:1710.10182 (2017)
30. Wang, N., Zha, W., Li, J., Gao, X.: Back projection: An effective postprocessing method for gan-based face sketch synthesis. Pattern Recognition Letters (2017)
31. Karacan, L., Erdem, E., Erdem, A.: Structure-preserving image smoothing via region covariances. ACM Transactions on Graphics **32** (2013) 176
32. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. arXiv:1609.04802 (2016)
33. Zhang, L., Zhang, L., Mou, X., Zhang, D.: Fsim: A feature similarity index for image quality assessment. IEEE transactions on Image Processing **20** (2011) 2378–2386
34. Chen, L.F., Liao, H.Y.M., Ko, M.T., Lin, J.C., Yu, G.J.: A new lda-based face recognition system which can solve the small sample size problem. Pattern recognition **33** (2000) 1713–1726