Semi-supervised Cycle-GAN for face photo-sketch translation in the wild

Chaofeng Chen[a], Wei Liu[b], Xiao Tan[c], Kwan-Yee K. Wong[a]

[a]*The Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, China*
[b]*SenseTime Research, Shenzhen 518000, China*
[c]*Baidu Inc, Department of Computer Vision Technology, Beijing 100085, China.*

Abstract

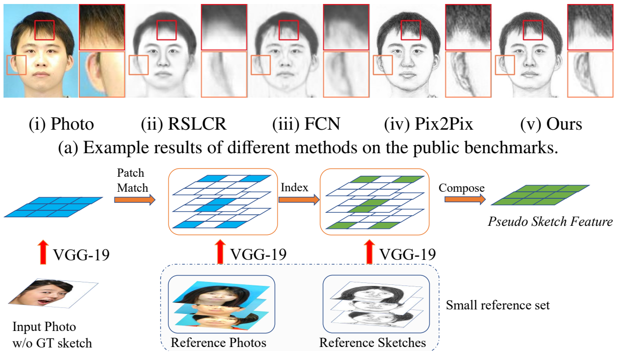
The performance of face photo-sketch translation has improved a lot thanks to deep neural networks. GAN based methods trained on paired images can produce high-quality results under laboratory settings. Such paired datasets are, however, often very small and lack diversity. Meanwhile, Cycle-GANs trained with unpaired photo-sketch datasets suffer from the *steganography* phenomenon, which makes them not effective to face photos in the wild. In this paper, we introduce a semi-supervised approach with a noise-injection strategy, named Semi-Cycle-GAN (SCG), to tackle these problems. For the first problem, we propose a *pseudo sketch feature* representation for each input photo composed from a small reference set of photo-sketch pairs, and use the resulting *pseudo pairs* to supervise a photo-to-sketch generator G_{p2s}. The outputs of G_{p2s} can in turn help to train a sketch-to-photo generator G_{s2p} in a self-supervised manner. This allows us to train G_{p2s} and G_{s2p} using a small reference set of photo-sketch pairs together with a large face photo dataset (without ground-truth sketches). For the second problem, we show that the simple noise-injection strategy works well to alleviate the *steganography* effect in SCG and helps to produce more reasonable sketch-to-photo results with less overfitting than fully supervised approaches. Experiments show that SCG achieves competitive performance on public benchmarks and superior results on photos in the wild.

1. Introduction

Face photo-sketch translation can be considered as a specific type of image translation between an input face photo and sketch. It has a wide range of applications. For example, police officers often have to identify criminals from sketch images, sketch images are also widely used in social media.

There are lots of works on face photo-sketch translation. Traditional methods are based on patch matching. They usually divide an input photo into small patches and find corresponding sketch patches in a reference dataset composed of well-aligned photo-sketch pairs. In this way, they (Song et al., 2014; Zhou et al., 2012; Zhu et al., 2017b; Wang and Tang, 2009) achieved pleasant results without explicitly modeling the mapping between photos and sketches, which is highly non-linear and difficult. However, sketches generated by these methods are often over-smoothed and lack subtle contents, such as ears in Fig. 1(a)(ii). Moreover, these methods are usually very slow due to the time-consuming patch matching and optimization process. Recent methods based on Convolutional Neural Networks (CNNs) try to directly learn the translation between photos and sketches. However, results produced by simple CNNs are usually blurry (see Fig. 1(a)(iii)), and Generative Adversary Networks (GAN) (Goodfellow et al., 2014) often generate unpleasant artifacts (see Fig. 1(a)(iv)). Finally, due to the lack of large training datasets, these learning-based approaches cannot generalize well to photos in the wild.

Latest works (Yu et al., 2020; Wang et al., 2017a; Fang et al., 2020) utilize Cycle-GAN (Zhu et al., 2017a) to learn the translation between photos and sketches. Cycle-GAN is designed for unpaired translation between different domains. Styles are translated with a discriminator loss and content consistency is guaranteed with a



(a) Example results of different methods on the public benchmarks.

(b) Pipeline for constructing the *pseudo sketch feature* (PSF)

Figure 1: Example results comparison and the proposed pseudo sketch feature.

cycle-consistency loss. However, the cycle-consistency loss used to constrain content is weak, and therefore these methods still require paired data to calculate an MSE (mean squared error) loss between the prediction and ground truth. In experiments, we observed that models directly using unpaired Cycle-GAN fail to preserve facial content (see Fig. 2). This is because Cycle-GAN learns to "hide" information of the input photos in the generated sketches as invisible high-frequency noise, also called *steganography* (Chu et al., 2017; Bashkirova et al., 2019). It makes it difficult to learn face photo-sketch translation with Cycle-GAN in an unpaired setting. Please refer to Sec. 3.1 for a detailed discussion.

In this paper, we propose a semi-supervised learning framework based on Cycle-GAN, named Semi-Cycle-GAN (SCG), for face photo-sketch translation. To ensure content consistency, we introduce a novel *pseudo sketch feature* (PSF) to supervise the training of the photo-to-sketch generator G_{p2s}. Figure 1(b) shows the pipeline to construct PSF for an input photo without ground truth sketch. 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 first subdivide an input photo and its VGG-19 (Simonyan and Zisserman, 2014) feature maps into overlapping patches. We then match (in the feature space) these photo patches with the photo patches in the reference set and compose a PSF from the VGG-19 features of the corresponding sketch patches in the reference set. We next supervise the training of G_{p2s} using the MSE between the feature maps of the generated sketch and the PSF of the input photo. The motivation for PSF is that styles of sketches are consistent for facial components with similar shapes. To find corresponding sketch patches for an input photo, we only need to match the facial components with similar shapes in the reference set. Since the shapes of facial components are limited, a small reference set with a few hundreds of photo-sketch pairs is often sufficient for this purpose. However, the same approach cannot be used for training the sketch-to-photo generator G_{s2p} because sketch patches with the same shape may give rise to photo patches of many different styles. Instead, we follow Cycle-GAN and use sketches generated by G_{p2s} to train G_{s2p} in a self-supervised manner. Although the proposed PSF helps to constrain the contents of the output sketches from G_{p2s}, we find *steganography* still exists and is quite harmful to the training of G_{s2p} because it learns to cheat. To solve this problem, we employ a simple *noise-injection* strategy to disrupt the invisible steganography and force G_{s2p} to learn better translation from sketches to photos. Although the inputs of G_{s2p} are noisy during training, we observed that G_{s2p} can handle clean sketches quite well during testing due to the intrinsic image prior of CNNs (Ulyanov et al., 2017). Experiments demonstrated that the *noise-injection* strategy can largely benefit the training of G_{s2p}.

In summary, our main contributions are:

- We propose a semi-supervised learning framework based on Cycle-GAN, named Semi-Cycle-GAN, for face photo-sketch translation.
- The proposed *pseudo sketch feature* (PSF) allows us to train G_{p2s} 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.
- We introduce a self-supervised approach to train the sketch-to-photo generator G_{s2p} *without using real sketches* through cycle-consistency. In particular, we find that cycle-consistency loss suffers greatly from invisible steganography, and the simple *noise-injection* strategy helps a lot to improve it.

A preliminary version of this work appeared in Chen et al. (2018a). We extend it in five aspects: (1) we combine our previously proposed semi-supervised learning framework with cycle-consistency to conduct both photo-to-sketch and sketch-to-photo translations; (2) we find that cycle-consistency loss suffers greatly from invisible steganography, and the simple *noise-injection* strategy helps a lot to improve it; (3) we add a Gram matrix loss based on PSF which provides second-order style supervision; (4) we provide more comparisons with recently proposed methods such as PS2MAN (Wang et al., 2017a), SCA-GAN (Yu et al., 2020), Knowledge Transfer (Zhu et al., 2019) (denoted as KT), GENRE (Li et al., 2021) and PANet (Nie et al., 2022); (5) we adopt recent perceptual oriented metrics (*i.e.*, LPIPS (Zhang et al., 2018a), DISTS (Ding et al., 2020), and FID (Heusel et al., 2017)) for performance evaluation. In particular, our extended framework shows better performance than Chen et al. (2018a).

2. Related Works

Exemplar-Based Methods Since photos and sketches are in two different modalities, it is not straightforward to learn a direct mapping between them. Tang and Wang (2003) introduced eigentransformation to perform exemplar matching between photos and sketches by assuming a linear transformation between them. Liu et al. (2005) noticed that the linear assumption holds better locally, and proposed the patch-based local linear embedding (LLE). Wang and Tang (2009) introduced a multi-scale markov random fields (MRF) model to resolve inconsistency between adjacent patches. Zhang et al. (2010) extended MRF with shape priors and SIFT features. Zhou et al. (2012) proposed the markov weight fields (MWF) model to synthesize new sketch patches that are not present in the training dataset. Gao et al. (2012) proposed to adaptively determine the number of candidate patches by sparse representation. Wang et al. (2013) proposed a transductive model which optimizes the MRF-based photo-to-sketch and sketch-to-photo models simultaneously. A few works such as Song et al. (2014)and Wang et al. (2017b) tried to improve the efficiency of the sketch generation procedure. Recent methods Zhu et al. (2017b) and Chen et al. (2018b) used features from a pretrained CNN network as the patch feature to replace unrobust traditional features.

Learning-Based Methods In recent years, CNN based methods have become the mainstream. Zhang et al. (2015) proposed to directly translate the input photo to sketch with a fully convolution network (FCN). Zhang et al. (2017) introduced a branched fully convolutional network (BFCN) which is composed of a content branch and a texture branch with different losses. Wang et al. (2017d) improved the vanilla GAN with multi-scale structure for face photo-sketch translation. Wang et al. (2017a) introduced multi-scale discriminators to Cycle-GAN. Zhang et al. (2018b) proposed multi-domain adversarial learning in the latent feature space of faces and sketches. Fang et al. (2020) introduced VGG-based feature identity loss to better preserve identity information. Yu et al. (2020) extended Cycle-GAN (Zhu et al., 2017a) with facial parsing map and proposed the SCA-GAN. Some recent popular works (Yi et al., 2019, 2020b,a; Huang et al., 2021; Li et al., 2020) consider a different kind of portrait style with simple thick lines and achieve pleasant results. However, it is out of the scope of this paper and hence we do not compare with them in this work.

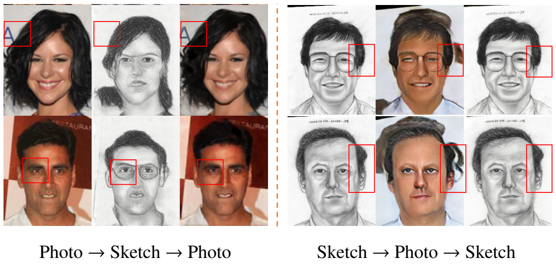


Figure 2: Illustration of steganography when training Cycle-GAN with unpaired data.

3. Semi-Cycle-GAN with noise-injection

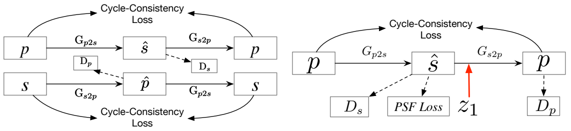
3.1. Steganography in Cycle-GAN

In this section, we first give a brief review of the unpaired Cycle-GAN for face photo-sketch translation. We then show how Cycle-GAN cheats with invisible steganography. Given a photo set P and a sketch set S, Cycle-GAN learns two generators: a photo-to-sketch generator G_{p2s} that maps photo $p \in P$ to sketch $s \in S$, and a symmetric sketch-to-photo generator G_{s2p} that maps sketch $s \in S$ to photo $p \in P$ (see Fig. 3(a)). Two discriminators D_s and D_p are used to minimize the style differences between the generated and real sketches (*i.e.*, $\hat{s}$ and s) and between generated and real photos (*i.e.*, $\hat{p}$ and p). Cycle-consistency losses are used to constrain content information in photo-sketch translation and are given by:

$$L_{cyc_p} = \mathbb{E}[||G_{s2p}(G_{p2s}(p)) - p||],$$
$$L_{cyc_s} = \mathbb{E}[||G_{p2s}(G_{s2p}(s)) - s||]. \quad (1)$$

Note that Eq. (1) does not impose a direct constraint over $G_{p2s}(p)$ and $G_{s2p}(s)$, and this leads to a large solution space. Chu et al. (2017) pointed out that Cycle-GAN tends to hide invisible steganography in the outputs to satisfy the cycle-consistency constraint when two domains have different complexity. Specifically, in face photo-sketch translation, the photo domain P is much more complex than the sketch domain S, which makes learning of G_{s2p} much more difficult than G_{p2s}. As a consequence, when we train G_{s2p} and G_{p2s} in an unpaired manner with cycle-consistency, the networks tend to learn a trivial solution by cheating with steganography rather than learning the desired translation networks. Figure 4 provides a theoretical illustration of steganography effect and how noise-injection helps to solve this problem. Given that the high-dimensional photo domain P contains a more extensive range of information in comparison to the low-dimensional sketch domain S, it poses a considerable challenge for the G_{s2p} network to reconstruct the missing information (*e.g.*, hair color) from grayscale input sketches. The networks tend to learn to conceal the extra information in a low-amplitude signal (*i.e.*, the red curve) to facilitate seamless reconstruction of the high-dimensional signal while retaining the appearance of the sketch signal. Since steganography needs to be low-amplitude signals, it is vulnerable to disruption through the application of random noise. In addition, G_{s2p} with random noise will act as a normal GAN to complement missing information in the low-dimensional sketch domain.

Figure 2 shows some example results when training Cycle-GAN with unpaired dataset. We can observe from the left half of Fig. 2 (*photo→sketch→photo*) that the lost letter in the generated sketch was recovered in the reconstructed photo, and extra glasses in the sketch were removed. A similar phenomenon also appears in the right half (*sketch→photo→sketch*). Closely related works including Chu et al. (2017) and Bashkirova et al. (2019) focus on how to avoid adversarial attack that is usually invisible in the images. We, on the other hand, are the first to study the visual effects brought by such steganography in face photo-sketch translation, which have been ignored by previous works based on Cycle-GAN (Yu et al., 2020; Wang et al., 2017a).



(a) Unpaired CycleGAN architecture (b) Our Semi-Cycle-GAN architecture

Figure 3: Framework of unpaired Cycle-GAN and our Semi-Cycle-GAN for face-sketch translation.

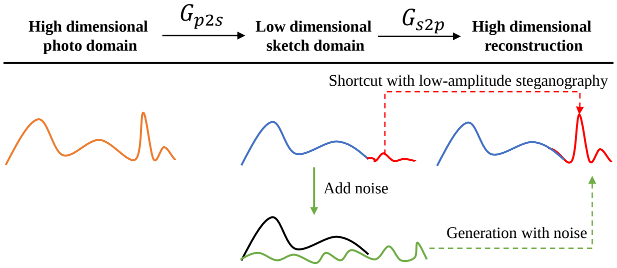

High dimensional photo domain
G_{p2s}
Low dimensional sketch domain
G_{s2p}
High dimensional reconstruction
Shortcut with low-amplitude steganography
Add noise
Generation with noise

Figure 4: Theoretical illustration of how noise-injection works.

To solve this problem, we propose the Semi-Cycle-GAN framework for face photo-sketch translation. As shown in Fig. 3(b), our framework is composed of four networks, namely G_{p2s}, G_{s2p}, D_s, and D_p. Unlike Cycle-GAN, we do not use the bidirectional cycle-consistency loss as a content constraint. We use PSF loss (see Sec. 3.2 for details) to supervise the training of G_{p2s}, and cycle-consistency loss with *noise-injection* to supervise the training of G_{s2p}. In this manner, we can train our Semi-Cycle-GAN using a small paired photo-sketch dataset together with a large face dataset.

3.2. *Pseudo Sketch Feature*

Given a test photo p, our target is to construct a pseudo sketch feature $\Phi'(p)$ as the supervision using the reference set $\mathcal{R}\{(p_i^{\mathcal{R}}, s_i^{\mathcal{R}})\}_{i=1}^N$, where $p_i^{\mathcal{R}}$ and $p_i^{\mathcal{R}}$ are photos and sketches in $\mathcal{R}$. We first use a pretrained VGG-19 network to extract a feature map for p at the l-th layer, denoted as $\Phi^l(p)$. Similarly, we can get the feature maps for photos and sketches in the reference dataset, *i.e.*, $\{\Phi^l(p_i^{\mathcal{R}})\}_{i=1}^N$ and $\{\Phi^l(s_i^{\mathcal{R}})\}_{i=1}^N$. The feature maps are then subdivided into $k \times k$ patches for the following feature patch matching process. For simplicity, we denote a vectorized representation of a $k \times k$ patch centered at a point j of

$\Phi^l(p)$ as $\Psi_j\left(\Phi^l(p)\right)$, and the same definition applies to $\Psi_j\left(\Phi^l(p_i^{\mathcal{R}})\right)$ and $\Psi_j(\Phi^l\left(s_i^{\mathcal{R}})\right)$. For each patch $\Psi_j\left(\Phi^l(p)\right)$, where $j = 1, 2, \ldots, m^l$ and $m^l = (H^l - k + 1) \times (W^l - k + 1)$ with H^l and W^l being the height and width of $\Phi^l(p)$, we find its best match $\Psi_{j'}\left(\Phi^l(p_{i'}^{\mathcal{R}})\right)$ in the reference set based on cosine distance, *i.e.*,

$$(i', j') = \underset{\substack{i^*=1\sim N \\ j^*=1\sim m^l}}{\arg\max} \frac{\Psi_j\left(\Phi^l(p)\right) \cdot \Psi_{j^*}\left(\Phi^l(p_{i^*}^{\mathcal{R}})\right)}{\left\|\Psi_j\left(\Phi^l(p)\right)\right\|_2 \left\|\Psi_{j^*}\left(\Phi^l(p_{i^*}^{\mathcal{R}})\right)\right\|_2}. \quad (2)$$

Since photos and their corresponding sketches in $\mathcal{R}$ are well aligned, the indices of the best matching result (i', j') can be used directly to find the corresponding sketch feature patch, *i.e.*, $\Psi_{j'}\left(\Phi^l(s_{i'}^{\mathcal{R}})\right)$ which serves as the pseudo sketch feature patch $\Psi'_j\left(\Phi^l(p)\right)$. Finally, we obtain the pseudo sketch feature representation (at layer l) for p as $\{\Psi'_j\left(\Phi^l(p)\right)\}_{j=1}^{m^l}$. We provide an intuitive visualization of PSF in supplementary material.

3.3. Loss Functions

We train generators (G_{p2s}, G_{s2p}) and discriminators (D_s, D_p) alternatively with the following loss functions

$$L_G^{total} = \lambda_p L_p + \lambda_{sty} L_{sty} + \lambda_{cyc} L_{cyc} + \lambda_{adv}(L_{G_{p2s}} + L_{G_{s2p}}), \quad (3)$$

$$L_D^{total} = L_{D_{p2s}} + L_{D_{s2p}} \quad (4)$$

where λ_p, λ_{sty}, λ_{cyc}, and λ_{adv} are trade-off weights for each loss term respectively. We describe details of each term as below.

Pseudo Sketch Feature Loss The pseudo sketch feature loss is formulated as

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

where $l = 3, 4, 5$ are relu3_1, relu4_1, and relu5_1 in VGG-19, and $\hat{s}$ is the predicted sketch from G_{p2s}.

Style Loss Inspired by recent style transfer methods, we include Gram Matrix loss (Gatys et al., 2016) as a second-order feature loss to provide better style supervision. We first average pool features in each $k \times k$ patch for both $\Psi_j\left(\Phi^l(\hat{s})\right)$ and $\Psi'_j\left(\Phi^l(p)\right)$, resulting in features ψ_l and ψ'_l of size $m^l \times c^l$, where c^l is the channel number in l-th layer. We then calculate the Gram Matrix loss as

$$L_{sty}(p, \hat{s}) = \sum_{l=3}^{5} \frac{1}{(c^l m^l)^2} \|\psi_l^T \psi_l - \psi_l'^T \psi'_l\|_2^2, \quad (6)$$

Cycle-Consistency with Noise-injection We use the cycle-consistency loss with *noise-injection* as supervision, which is formulated as

$$L_{cyc}(p) = \|G_{s2p}\left(G_{p2s}(p) + \sigma z_1\right) - p\|_2^2, \quad (7)$$

where z_1 is randomly sampled from a normal distribution with the same dimensions as $G_{p2s}(p)$, and σ is a hyperparameter that controls the noise level.

GAN Loss We use the hinge loss to make the training process more stable. The objective functions of hinge loss are given by

$$L_G = -\mathbb{E}[D(G(x))], \quad (8)$$

$$L_D = \mathbb{E}[\max(0, 1 - D(y))] + \mathbb{E}[\max(0, 1 + D(G(x)))], \quad (9)$$

where x, y, D refer to p, s, D_s when G is G_{p2s}, and s, p, D_p when G is G_{s2p}.

4. Experiments

4.1. Datasets and Metrics

Datasets To compare with previous works, we evaluate our model on two public benchmark datasets, namely the CUFS dataset (combination of CUHK (Tang and Wang, 2003), AR (Martinez and benavente., 1998) and XM2VTS (Messer et al., 1999)), and the CUFSF dataset (Zhang et al., 2011b). For semi-supervised learning, we use extra face photos from VGG-Face dataset (Parkhi et al., 2015). We randomly select 1,244 photos from VGG-Face to test model performance on natural images. More details are provided in supplementary material.

Training Details We set all the trade-off weights λ_p, λ_{sty}, λ_{cyc}, and λ_{adv} to 1 for simplicity. We use Adam (Kingma and Ba, 2014) with learning rates 0.001 for generators and 0.004 for discriminators, and set $\beta_1 = 0.9, \beta_2 = 0.999$. The learning rates are linearly decayed to 0 after the first

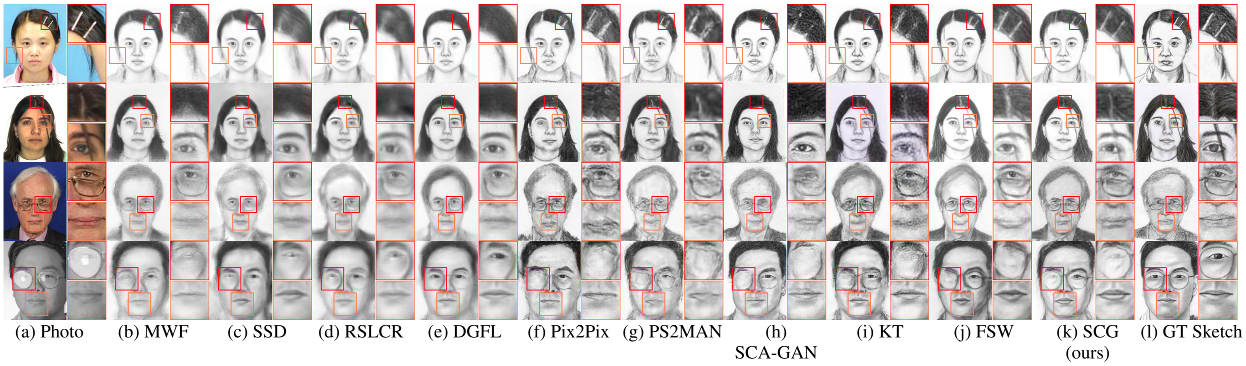


Figure 5: Examples of synthesized face sketches on the CUFS dataset and the CUFSF dataset. See more examples in supplementary material.

10 epochs. The training batch size is 2, and models are trained on Nvidia 1080Ti GPUs.

Metrics For test sets with ground truth, we use FSIM (Zhang et al., 2011a), LPIPS (Zhang et al., 2018a) and DISTS Ding et al. (2020) to measure the texture quality, and NLDA score to measure the identity similarity following Wang et al. (2017b). For the evaluation of face-sketch translation in the wild, there are no ground truth sketches to calculate FSIM, LPIPS, and DISTS. We therefore exploit FID (Heusel et al., 2017) to measure the feature statistic distance between the generated sketch datasets and real sketch datasets. We explain details of these metrics in supplementary material.

4.2. Comparison on Public Benchmarks

We evaluate our model on both photo-to-sketch and sketch-to-photo translations on CUFS and CUFSF, which were captured under laboratory settings. We compare our results both qualitatively and quantitatively with four exemplar-based methods, namely MWF (Zhou et al., 2012), SSD (Song et al., 2014), RSLCR (Wang et al., 2017b), and DGFL (Zhu et al., 2017b), and five GAN-based methods, namely Pix2Pix-GAN (Isola et al., 2017), PS2MAN (Wang et al., 2017a), MDAL (Zhang et al., 2018b), KT (Zhu et al., 2019) and SCA-GAN (Yu et al., 2020). We obtain the results of MWF, SSD, RSLCR, and DGFL from Wang et al. (2017b), the results of SCA-GAN and KT from the respective authors, and use the public codes of Cycle-GAN and PS2MAN to produce the results. We also compare the photo-to-sketch translation results with our previous work FSW (Chen et al., 2018a). All the models are trained on the CUFS and CUFSF datasets with the same train/test partition.

4.2.1. Photo-to-Sketch Translation

Figure 5 shows some photo-to-sketch results on CUFS and CUFSF. Exemplar-based methods (Fig. 5(b,c,d,e)) in general perform worse than learning-based methods (Fig. 5(f,g,h,i,j,k)). Their results are over-smoothed and do not show hair textures. They also fail to preserve contents well, such as hairpins in the first row and glasses in the last row. GAN-based methods can generate better textures, but they usually produce artifacts because of the unstable training. For example, Pix2Pix produces lots of artifacts in the hair and eyes (Fig. 5(f)), and PS2MAN generates lots of artifacts when the facial parts of inputs are not clear or with a strong reflection of light (see the last two rows of Fig. 5(g)). Although the results of SCA-GAN look great, it suffers from incorrect parsing map guidance, such as hairpins in the first row, hairlines in the second row of Fig. 5(h). Referring to Fig. 5(j,k), we have improved our previous results of FSW by introducing L_{sty} and the photo reconstruction branch.

The quantitative results with different metrics in Tab. 1 support our observations. It can be observed that exemplar-based methods perform much worse in terms of all metrics including FSIM, LPIPS, DISTS and NLDA. KT shows the best FSIM score but poor perceptual scores compared with SCG. We can see from Fig. 5(i) that the textures, especially hair textures, generated by KT are much worse than SCG. SCA-GAN generates better textures but the generated images might be different from

Table 1: Quantitative results for photo-to-sketch translation. SCA-GAN* needs a parsing map as guidance.

Method	FSIM ↑		LPIPS ↓		DISTS ↓		NLDA↑	
	CUFS	CUFSF	CUFS	CUFSF	CUFS	CUFSF	CUFS	CUFSF
MWF	0.7144	0.7029	0.3671	0.4090	0.2533	0.2825	92.3	73.8
SSD	0.6957	0.6824	0.4033	0.4283	0.2536	0.2608	91.1	70.6
RSLCR	0.6965	0.6650	0.4042	0.4521	0.2556	0.2896	98.0	75.9
DGFL	0.7078	0.6957	0.3655	0.3972	0.2410	0.2480	98.2	78.8
Pix2Pix-GAN	0.7153	0.7060	0.3600	0.3868	0.2151	0.2025	93.8	71.7
PS2MAN	0.7157	0.7219	0.3794	0.4155	0.2430	0.2471	97.6	77.0
SCA-GAN*	0.7160	0.7268	0.3608	0.4169	0.2005	0.2168	—	—
MDAL	0.7275	0.7076	0.3319	0.3841	0.2037	0.2096	96.6	66.7
KT	**0.7369**	**0.7311**	0.3485	0.3743	0.2116	0.2039	98.0	**80.4**
FSW	0.7274	0.7103	0.3262	0.3787	0.2063	0.2111	98.0	78.04
SCG (ours)	0.7343	0.7261	**0.3232**	**0.3489**	**0.1967**	**0.184**	**98.6**	78.1

Table 2: Quantitative results for sketch-to-photo translation. SCA-GAN* needs a parsing map as guidance.

Method	FSIM ↑		LPIPS ↓		DISTS ↓		NLDA↑	
	CUFS	CUFSF	CUFS	CUFSF	CUFS	CUFSF	CUFS	CUFSF
Pix2Pix-GAN	0.7598	0.7877	0.3977	0.4025	0.2421	0.2481	87.1	51.4
PS2MAN	0.7645	0.7807	0.3668	0.4267	0.2254	0.2706	84.7	42.2
SCA-GAN*	0.7633	**0.8304**	0.3251	**0.3198**	0.1794	**0.1829**	—	—
KT	**0.7794**	0.7932	**0.3233**	0.3758	0.1821	0.2379	**93.8**	**65.9**
SCG (ours)	0.7652	0.7777	0.3374	0.3527	**0.1710**	0.2082	90.0	49.7

the original images (*e.g.*, missing components) due to incorrect parsing map, which also leads to poor LPIPS and DISTS scores. In contrast, our SCG presents the second best results in terms of FSIM and the best results in terms of LPIPS and DISTS. As for sketch recognition, SCG also demonstrates best NLDA score on CUFS and competitive results on CUFSF, which clearly demonstrate its superiority.

4.3. Sketch-to-Photo Translation

Figure 6 shows some example sketch-to-photo results. Same as photo-to-sketch translation, the results of Pix2Pix and PS2MAN contain many undesired artifacts. SCA-GAN produces results with the best visual quality, which is consistent with the quantitative results shown in Tab. 2. However, it still generates results with missing components under incorrect parsing map predictions, such as the missing eyes and glasses in the last row of Fig. 6(d). Without any GAN losses, KT suffers from unrealistic textures. For instance, results in Fig. 6(e) are grainy. Although SCG is trained in a self-supervised manner without seeing any real input sketches, it still shows competitive performance. Referring to Tab. 2, SCG shows the best or second results in 5 out of 8 columns. The biggest problem of SCG is that the synthesized colors are quite different from the ground truth. This is legitimate because the model is not suppose to recover exact color as ground truth unless overfitting.

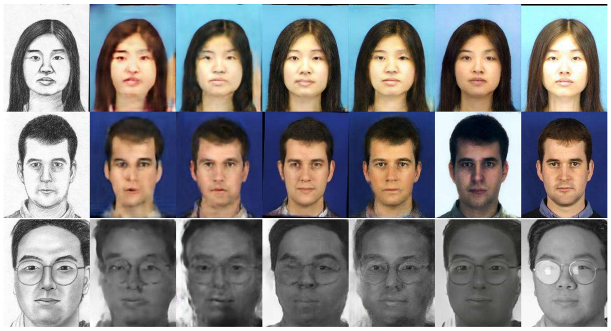


Figure 6: Examples of synthesized face photos on the CUFS dataset and the CUFSF dataset.

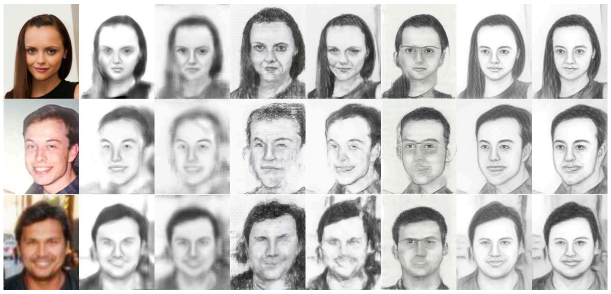


Figure 7: Comparison for images in the wild. Benefiting from the additional training data, SCG can deal with various photos.

4.4. Photo-to-Sketch Translation in the Wild

In this section, we will focus on photo-to-sketch translation in the wild. Since there are too many sketch styles in the wild, sketch-to-photo translation in the wild is beyond the scope of this paper, and we will leave it for future work. We compare SCG with other methods which provide codes, including SSD, RSLCR, Pix2Pix-GAN, PS2MAN, Cycle-GAN. Figure 7 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 and poses *etc*. Among the results of other methods, exemplar-based methods (see Fig. 7(b,c)) fail to deal with pose changes

Table 3: Quantitative results and user study for photo-to-sketch translation in the wild.

Method	FID↓
SSD	94.6
Fast-RSLCR	144.0
Pix2Pix-GAN	86.7
PS2MAN	90.8
Cycle-GAN	87.8
FSW	<u>81.3</u>
SCG (ours)	**67.9**

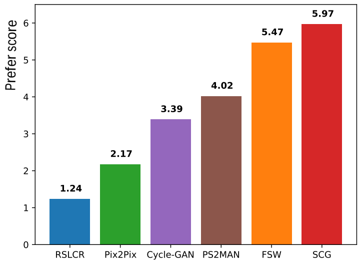


Table 4: Quantitative comparison on WildSketch dataset.

Method	Cycle-GAN	GENRE	CA-GAN	PANet	Ours
FSIM↑	0.6654	0.6902	<u>0.6960</u>	0.6950	**0.70**

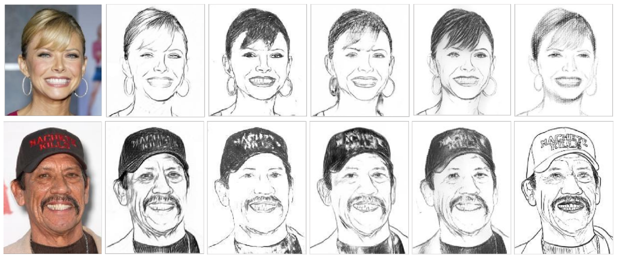

(a) Photo (b) Cycle-GAN (c) (S)CA-GAN (d) PANet (e) Ours (f) GT

Figure 8: Example comparison with WildSketch dataset.

and different hairstyles. 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. 7(g,h) that only FSW and SCG can handle photos in the wild well and generate pleasant results. Compared with FSW, SCG can generate more realistic shadows and textures. The same conclusion can also be drawn from the quantitative results shown in Tab. 3. We also conduct user study to better evaluate their subjective performance, as shown in Tab. 3 right part. We notice that our methods (FSW and SCG) are much preferred over previous methods. By introducing the G_{s2p} branch and cycle-consistency, SCG further improves the performance of our previous work FSW. Details of user study are in supplementary material.

We have also included comparison on the latest in-the-wild benchmark WildSketch Nie et al. (2022), as shown in Tab. 4 and Fig. 8. Our findings indicate that when incorporating additional, diverse photos from the VGG face dataset, our method achieves SoTA performance. Figure 8 also supports our claim that the inclusion of extra training photos improves generalization abilities of our model. For instance, ours demonstrates greater robustness towards hair color variations in the first row and shows better result with the presence of a hat in the second row. These results underscore the effectiveness of our proposed semi-supervised approach.

Table 5: Ablation study of Semi-Cycle-GAN. σ: noise level, k: feature patch size, L_{sty}: use second-order style loss or not.

Configuration	A	B	C	D	E	F	G
σ	0	10	20	30	20	20	20
k	1	1	1	1	3	5	3
L_{sty}	✗	✗	✗	✗	✗	✗	✓
LPIPS↓(P2S)	0.3260	0.3273	0.3277	0.3287	<u>0.3257</u>	0.3273	**0.3235**
LPIPS↓(S2P)	0.4273	0.3454	0.3435	0.3461	<u>0.3433</u>	0.3447	**0.3374**

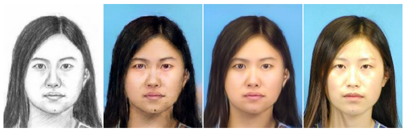

(a) Input (b) $\sigma = 0$ (c) $\sigma = 20$ (d) GT Photo

Figure 9: Effect of *noise-injection*.

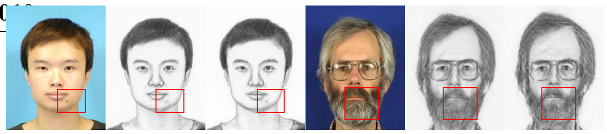


Photo w/o L_{sty} w/ L_{sty} Photo w/o L_{sty} w/ L_{sty}

Figure 10: Examples of improvement on G_{p2s} brought by style loss.

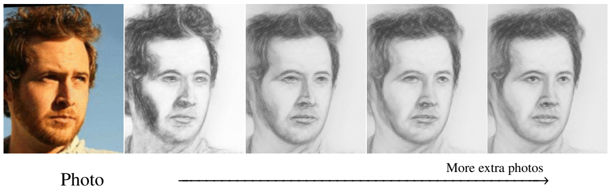


Figure 11: Effectiveness of additional training photos.

4.5. Ablation Study

To study the effectiveness of different components of the proposed method, we gradually modify the baseline Semi-Cycle-GAN and compare their results. Table 5 shows the results of all model variations. We discuss the results below.

Noise injection. We show an example result with and without *noise-injection* in Fig. 9. It can be observed that Fig. 9(c) with $\sigma = 20$ is much better than Fig. 9(b) with $\sigma = 0$. This demonstrates that *noise-injection* can greatly improve the performance of G_{s2p}. This is because the proposed *noise-injection* strategy breaks the steganography in the outputs of G_{p2s}, and increases the generalization ability of G_{s2p}. We explore models with different levels of *noise-injection*, and the results are shown in columns A, B, C, and D of Tab. 5. We can see that adding more noise is not helpful to the performance of G_{s2p} but degrades the performance of G_{p2s}. This is likely because the backward gradients from G_{s2p} are corrupted when noise-injection level is too high. We empirically find $\sigma = 20$ strikes a good balance between the performance of G_{p2s} and G_{s2p}.

Patch size. We present the results with patch size 1, 3, and 5 in columns C, E, and F of Tab. 5 respectively. We can observe that $k = 3$ gives the best performance, while $k = 5$ is worse than $k = 3$. This may be caused by the fact that a large patch in the feature space represents a much larger patch in the pixel space and this leads to undesired extra contents in the pseudo sketch feature. We therefore set $k = 3$ in our experiments.

Second-order style loss. Comparing the results in columns E and G of Tab. 5, we can notice that model with L_{sty} shows better performance for both G_{p2s} and G_{s2p}. This is because L_{sty} provides better style supervision for G_{p2s}, which can in turn benefit the training of G_{s2p}. Figure 10 shows some examples of improvement on G_{p2s} brought by style loss.

Extra 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. 11, as we add more photos to the training set, the results improve significantly, see the eyes region.

5. Conclusion

In this paper, we propose a semi-supervised Cycle-GAN, named Semi-Cycle-GAN (SCG), for face photo-sketch translation. 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. Since directly training G_{s2p} in a self-supervised manner as Cycle-GAN suffers from steganography, we exploit a *noise-injection* strategy to improve the robustness. Experiments show that our method can produce sketches comparable to (if not better than) those produced by other state-of-the-art methods on four public benchmarks, and outperforms them on photo-to-sketch translation in the wild.

References

Bashkirova, D., Usman, B., Saenko, K., 2019. Adversarial self-defense for cycle-consistent gans, in: Advances in Neural Information Processing Systems.

Bradley, R.A., Terry, M.E., 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika 39, 324–345.

Chen, C., Liu, W., Tan, X., Wong, K., 2018a. Semi-supervised learning for face sketch synthesis in the wild, in: Asian Conference on Computer Vision (ACCV).

Chen, C., Tan, X., , Wong, K.Y.K., 2018b. Face sketch synthesis with style transfer using pyramid column feature. IEEE Winter Conference on Applications of Computer Vision .

Chen, L.F., Liao, H.Y.M., Ko, M.T., Lin, J.C., Yu, G.J., 2000. A new lda-based face recognition system which can solve the small sample size problem. Pattern recognition 33, 1713–1726.

Chu, C., Zhmoginov, A., Sandler, M., 2017. Cyclegan, a master of steganography .

Ding, K., Ma, K., Wang, S., Simoncelli, E.P., 2020. Image quality assessment: Unifying structure and texture similarity. CoRR abs/2004.07728. URL: https://arxiv.org/abs/2004.07728.

Fang, Y., Deng, W., Du, J., Hu, J., 2020. Identity-aware cyclegan for face photo-sketch synthesis and recognition. Pattern Recognition 102, 107249.

Gao, X., Wang, N., Tao, D., Li, X., 2012. Face sketch–photo synthesis and retrieval using sparse representation. IEEE Transactions on circuits and systems for video technology 22, 1213–1226.

Gatys, L.A., Ecker, A.S., Bethge, M., 2016. Image style transfer using convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2414–2423.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Advances in neural information processing systems, pp. 2672–2680.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S., 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium, in: Advances in Neural Information Processing Systems, pp. 6626–6637.

Huang, J., Liao, J., Tan, Z.T., Kwong, S., 2021. Multi-density sketch-to-image translation network. IEEE Transactions on Multimedia .

Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. CVPR .

Karacan, L., Erdem, E., Erdem, A., 2013. Structure-preserving image smoothing via region covariances. ACM Transactions on Graphics 32, 176.

Kingma, D., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv:1412.6980 .

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al., 2016. Photo-realistic single image super-resolution using a generative adversarial network. arXiv:1609.04802 .

Li, C., Wand, M., 2016. Combining markov random fields and convolutional neural networks for image synthesis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2479–2486.

Li, X., Gao, F., Huang, F., 2021. High-quality face sketch synthesis via geometric normalization and regularization, in: 2021 IEEE International Conference on Multimedia and Expo (ICME), IEEE. pp. 1–6.

Li, Z., Deng, C., Yang, E., Tao, D., 2020. Staged sketch-to-image synthesis via semi-supervised generative adversarial networks. IEEE Transactions on Multimedia .

Liu, Q., Tang, X., Jin, H., Lu, H., Ma, S., 2005. A nonlinear approach for face sketch synthesis and recognition, in: IEEE Conference on Computer Vision and Pattern recognition, pp. 1005–1010.

Martinez, A., benavente., R., 1998. The AR face database. Technical Report. CVC Tech. Report.

Messer, K., Matas, J., Kittler, J., Jonsson, K., 1999. Xm2vtsdb: The extended m2vts database, in: In Second International Conference on Audio and Video-based Biometric Person Authentication, pp. 72–77.

Nie, L., Liu, L., Wu, Z., Kang, W., 2022. Unconstrained face sketch synthesis via perception-adaptive network and a new benchmark. Neurocomputing .

Parkhi, O.M., Vedaldi, A., Zisserman, A., 2015. Deep face recognition, in: British Machine Vision Conference.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 .

Song, Y., Bao, L., Yang, Q., Yang, M.H., 2014. Real-time exemplar-based face sketch synthesis, in: European Conference on Computer Vision, pp. 800–813.

Tang, X., Wang, X., 2003. Face sketch synthesis and recognition, in: IEEE International Conference on Computer Vision, pp. 687–694.

Ulyanov, D., Vedaldi, A., Lempitsky, V., 2017. Deep image prior. arXiv:1711.10925 .

Wang, L., Sindagi, V.A., Patel, V.M., 2017a. High-quality facial photo-sketch synthesis using multi-adversarial networks. arXiv:1710.10182 .

Wang, N., Gao, X., Li, J., 2017b. Random sampling for fast face sketch synthesis. arXiv:1701.01911 .

Wang, N., Tao, D., Gao, X., Li, X., Li, J., 2013. Transductive face sketch-photo synthesis. IEEE transactions on neural networks and learning systems 24, 1364–1376.

Wang, N., Zha, W., Li, J., Gao, X., 2017c. Back projection: An effective postprocessing method for gan-based face sketch synthesis. Pattern Recognition Letters .

Wang, N., Zhu, M., Li, J., Song, B., Li, Z., 2017d. Data-driven vs. model-driven: Fast face sketch synthesis. Neurocomputing .

Wang, X., Tang, X., 2009. Face photo-sketch synthesis and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 1955–1967.

Yi, R., Liu, Y.J., Lai, Y.K., Rosin, P.L., 2019. Apdrawinggan: Generating artistic portrait drawings from face photos with hierarchical gans, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10743–10752.

Yi, R., Liu, Y.J., Lai, Y.K., Rosin, P.L., 2020a. Unpaired portrait drawing generation via asymmetric cycle mapping, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8217–8225.

Yi, R., Xia, M., Liu, Y.J., Lai, Y.K., Rosin, P.L., 2020b. Line drawings for face portraits from photos using global and local structure based gans. IEEE transactions on pattern analysis and machine intelligence .

Yu, J., Xu, X., Gao, F., Shi, S., Wang, M., Tao, D., , Huang, Q., 2020. Towards realistic face photo-sketch synthesis via composition-aided gans. IEEE Transactions on Cybernatics .

Zhang, D., Lin, L., Chen, T., Wu, X., Tan, W., Izquierdo, E., 2017. Content-adaptive sketch portrait generation by decompositional representation learning. IEEE Transactions on Image Processing (TIP) 26, 328–339.

Zhang, L., Lin, L., Wu, X., Ding, S., Zhang, L., 2015. End-to-end photo-sketch generation via fully convolutional representation learning, in: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (ICMR), pp. 627–634.

Zhang, L., Zhang, L., Mou, X., Zhang, D., 2011a. Fsim: A feature similarity index for image quality assessment. IEEE transactions on Image Processing 20, 2378–2386.

Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O., 2018a. The unreasonable effectiveness of deep features as a perceptual metric, in: CVPR.

Zhang, S., Ji, R., Hu, J., Lu, X., Li, X., 2018b. Face sketch synthesis by multidomain adversarial learning. IEEE transactions on neural networks and learning systems 30, 1419–1428.

Zhang, W., Wang, X., Tang, X., 2010. Lighting and pose robust face sketch synthesis, in: European Conference on Computer Vision, pp. 420–433.

Zhang, W., Wang, X., Tang, X., 2011b. Coupled information-theoretic encoding for face photo-sketch recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 513–520.

Zhou, H., Kuang, Z., Wong, K.Y.K., 2012. Markov weight fields for face sketch synthesis, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1091–1097.

Zhu, J.Y., Park, T., Isola, P., Efros, A.A., 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networkss, in: Computer Vision (ICCV), 2017 IEEE International Conference on.

Zhu, M., Wang, N., Gao, X., Li, J., 2017b. Deep graphical feature learning for face sketch synthesis, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, pp. 3574–3580.

Zhu, M., Wang, N., Gao, X., Li, J., Li, Z., 2019. Face photo-sketch synthesis via knowledge transfer. IJCAI International Joint Conference on Artificial Intelligence 2019-Augus, 1048–1054. doi:10.24963/ijcai.2019/147.

Appendix A. More Methodology Details

Appendix A.1. Visualization of Pseudo Sketch Features

Figure A.13 visualizes examples of the pseudo sketch feature. It can be seen that the pseudo sketch feature provides a good approximation of the real sketch feature (see first two columns of Fig. A.13). We also show naïve reconstruction obtained by directly using the matching index to index the pixel values in the reference 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 face photos under different poses (see last row of Fig. A.13).

Appendix A.2. Hyper-parameters of Pseudo Sketch Feature Loss

For the pseudo sketch feature loss L_p, we set $l = 3, 4, 5$ to relu3_1, relu4_1, and relu5_1 in VGG-19. We choose these 3 layers mainly for two reasons: 1) better texture representation; 2) computation efficiency. Li *et al.* Li and Wand (2016) pointed out that compared with features from shallow layers, deep features after relu3_1 are more robust to appearance changes and geometric transforms. We conduct a simple experiment to verify this, and the results are presented in Fig. A.12. It can be observed that the model cannot synthesize sketch textures using shallow features from relu1_1 and relu2_1, and can generate better textures with high-level features, such as relu5_1. However, with only high-level features, the model also generates artifacts (*e.g.*, eyes of the sketches in Fig. A.12). Besides, shallow feature maps have higher spatial resolutions, and it requires lots of GPU memory to calculate L_p. Based on the above analysis, we set $l = 3, 4, 5$ to strike a balance between performance and computation cost.

Appendix B. More Dataset and Implementation Details

Photo-Sketch Pairs. We use four public datasets, namely the CUHK dataset Tang and Wang (2003), the AR dataset Martinez and benavente. (1998), the XM2VTS dataset Messer et al. (1999), and the CUFSF dataset Zhang et al. (2011b), to evaluate our model. In Wang et al. (2017b); Zhu et al. (2017b), the first three datasets were combined to form the CUFS dataset. Note that the CUFSF dataset used in Zhu et al. (2017b); Wang et al. (2017a); Yu et al. (2020) contains only grayscale photos. In order to train a universal model for all datasets, we collect a color version of the CUFSF dataset[1] containing 986 photo-sketch pairs. Details are summarized in Table B.6, and Fig. B.14 shows some examples of photo-sketch pairs from them.

Table B.6: 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. VGG-Face dataset Parkhi et al. (2015) is a popular dataset containing face photos in the wild. We use a subset of it in this work. VGG-Face has 2,622 subjects with 1,000 photos for each subject. We randomly select $\mathcal{N}$ photos of 2,000 subjects for training. The resulting dataset are named as VGG-Face$\mathcal{N}$, where $\mathcal{N} = 01, 02, \ldots, 10$. The VGG-Face$\mathcal{N}$ datasets are used to validate the performance relationship with increasing training photos. For the test dataset, 2 photos are randomly selected for each subject in the test split (no identity overlap with training dataset), which results in a VGG test set of 1,244 photos. Some examples from training and testing datasets are presented in Fig. B.15.

Preprocessing. For the reference datasets, we need the photo-sketch pairs to be well aligned. We perform alignment with similarity transformation based on 68 face landmarks detected using `dlib`[2]. The output faces and sketches are aligned with two eyes located at $(75, 125)$

[1] Downloaded from https://www.nist.gov/itl/iad/image-group/color-feret-database

[2] http://dlib.net/

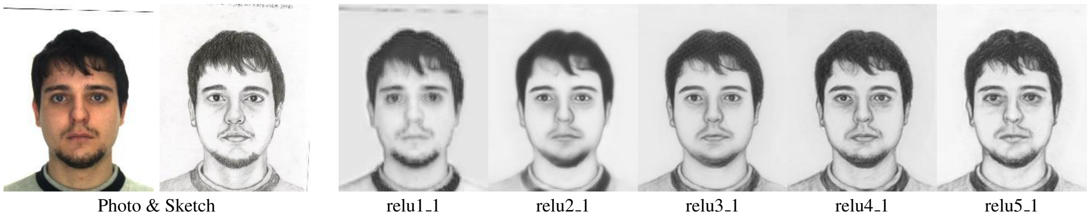


Figure A.12: Results of using different layers in pseudo sketch feature loss.

and $(125, 125)$ respectively. The output size is set to 250×200 in order to perform a fair comparison with previous works. For photos/sketches whose landmarks cannot be detected, we simply discard them.

Appendix B.1. Implementation Details of Patch Matching

Although the patch matching only happens in training stage, it is still time-consuming. We accelerate patch matching in the following three ways. First, feature patches for the photos and sketches in the reference dataset are precomputed and stored in hard disk for fast query. Second, we use a coarse-to-fine strategy to search for the best matching feature patch. We find the best-matched n reference photos (we set $n = 3$ in the whole training process) for the input photo based on the similarity of their relu5_1 feature maps, which can be calculated fast. Fine-scale patch matching is then performed on these n reference photos. Third, we use the convolution operator to implement Eq. (2) which can be greatly accelerated with GPU.

Appendix C. More Experiment Details

Appendix C.1. Metric analysis

Previous works Wang et al. (2017b); Yu et al. (2020); Zhu et al. (2017b) usually adopted structural similarity (SSIM) Karacan et al. (2013) to evaluate the performance of sketch generation for test datasets with ground-truth sketches (e.g., CUFS and CUFSF). However, many works Ledig et al. (2016); Wang et al. (2017c,a)) pointed out that SSIM is not always consistent with the perceptual quality because SSIM favors slightly blurry images and fails to evaluate images with rich textures. To verify this, we show some sketches generated using different methods together with their SSIM scores in Fig. C.17. We can observe that although the results of Pix2Pix-GAN and our model have better textures, the result of RSLCR still has a better SSIM score because it is smoother. When we smooth all sketches with a bilateral filter, we notice that SSIM score for RSLCR remains almost unchanged, while the SSIM scores for Pix2Pix-GAN and our model improve by more than 1.5%.

Due to the drawbacks of SSIM, we choose FSIM Zhang et al. (2011a) as one of our image quality assessment metrics. FSIM takes local structure into account and gives lower scores to smooth results without textures, see Fig. C.17 for reference. Considering that metrics based on VGG feature space demonstrate better consistency with human perception, we also include two recent VGG-based metrics, namely LPIPS and DISTS. We use the `PyTorch` codes provided by Chen *et al.*[3] to calculate these metrics. For the evaluation of face-sketch translation in the wild, there are no ground truth sketches to calculate FSIM, LPIPS, and DISTS. We therefore exploit FID score to measure the feature statistic distance between the generated sketch datasets and real sketch datasets.

Appendix C.2. Face Recognition Details

Following the practice of Wang et al. (2017b), we employed the null-space linear discriminant analysis (NLDA) Chen et al. (2000) to perform the recognition experiments. For CUFS, we randomly selected 150 synthesized sketches and their ground-truth sketches from the test set (338 test photos) to train a classifier and used the rest 188 for testing. For CUFSF(gray, crop), we randomly selected 300 synthesized sketches and their ground-truth sketches from the test set (944 test photos) to train a classifier and used the rest 644 for testing. Each experiment was repeated 20 times. We do not include SCA-GAN in

[3] `https://github.com/chaofengc/IQA-PyTorch`

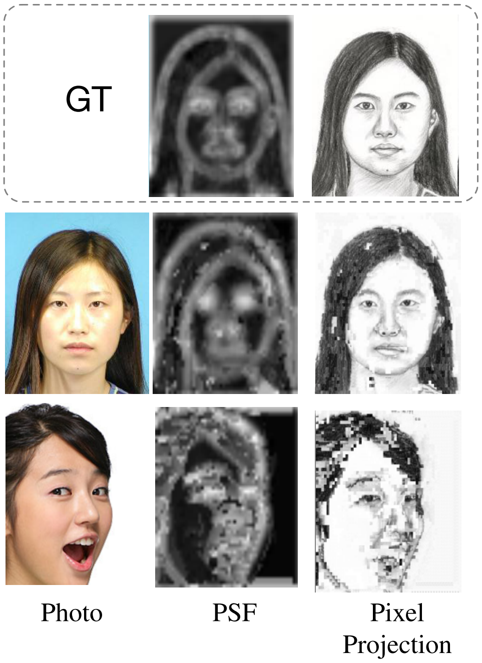


Figure A.13: Examples of PSF in the relu3_1 layer and the pixel level projection of the patch matching result. First row: ground truth feature and sketch with the photo in second row as input. Second row: results of laboratory images. Third row: results of natural images. (*Note that the pixel level results are for visualization only, and they are not actually being computed or used in training.)*

this comparison because it uses an extra face parsing map as guidance, and adopts a slightly different training/test split.

Figure C.16 shows the recognition accuracy of different methods on these two datasets. The results for photo-to-sketch translation are shown in Fig. C.16(a,b). We can see that the proposed SCG achieves the best NLDA scores with different feature dimensions on CUFS and competitive results on CUFSF. We notice that the recognition rate on CUFSF is similar to FSW and slightly worse than KT. This is mainly because the ground truth sketches in CUFSF are deformed too much compared with the input photos. We believe the NLDA scores on CUFSF cannot represent the quality of generated sketches. In fact, we observe that the result of SCG is clearer and without extra shadows or artifacts.

The results for sketch-to-photo translation are given in Fig. C.16(c,d). As mentioned in main text, it is expected that SCG cannot give the best performance because we do not use any ground-truth sketches to train G_{s2p}, and the colors of the results are quite different from the ground-truth photos. Nonetheless, SCG still performs better than PS2MAN on both datasets, and better than Pix2Pix on CUFS. According to the visual examples in main paper, we can observe that SCG preserves the facial components better than these two methods, and produces fewer artifacts. This explains why our results are better despite color inconsistency.

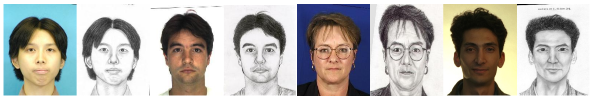

(a) CUHK Student (b) AR (c) XM2VTS (d) CUFSF

Figure B.14: Example photo-sketch pairs from existing datasets. It can be observed that (a)(b) and (c)(d) have different sketch styles (e.g., facial muscles and hair).

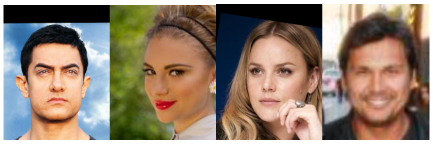
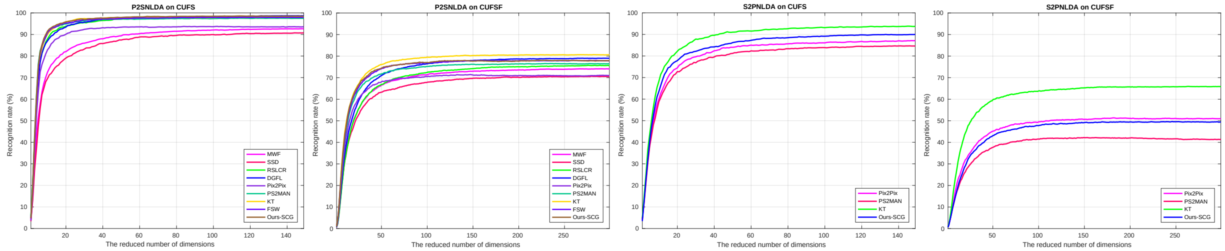

(a) Training photos (b) Test photos

Figure B.15: Example training photos from the VGG-Face01 dataset and example test photos from the VGG test set.

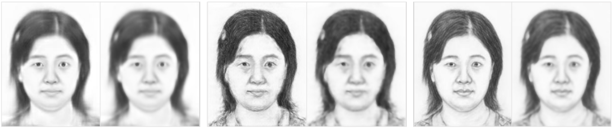
Appendix C.3. User Study Details

We also conduct a user study for sketch synthesis in the wild. The methods considered include RSLCR, Pix2Pix-GAN, PS2MAN, Cycle-GAN, FSW (our previous method), and SCG (proposed). Different from our previous project which asked the subjects to rank results of different methods, we adopt a more comprehensive strategy to do the subjective study on an online human crowdsourcing platform. To be specific, we employed a two-alternative forced choice (2AFC) method. The crowd workers were shown two generated sketches at one time, and were asked to choose the sketch with better quality. An example photo-sketch pair was shown as a reference. We randomly select 30 images from VGG-Test dataset, generate sketches with the above methods and create 6 different surveys. Each survey contains results for 5 different images and $\binom{2}{6} \times 5 = 75$ questions in total for the 6 compared methods. Each worker was asked to do one



(a) NLDA scores on CUFS sketches (b) NLDA scores on CUFSF sketches (c) NLDA scores on CUFS photos (d) NLDA scores on CUFSF photos

Figure C.16: Recognition rates of using synthesized face sketches (a,b) and synthesized face photos (c,d), respectively, against feature dimensions on CUFS and CUFSF.



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

(b) GAN
SSIM: 0.5648/0.5953.
FSIM: 0.7559/0.7506.

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

Figure C.17: SSIM and FSIM scores of some generated sketches (left) and their smoothed counterparts (right).

of these surveys. We collected 60 survey results from different crowd workers in total. We used the well-known Bradley–Terry model Bradley and Terry (1952) to convert the paired comparison results to global ranking. Give method i and j, the probability that i is better than j is defined as

$$P(i > j) = \frac{e^{\beta_i}}{e^{\beta_i} + e^{\beta_j}} \tag{C.1}$$

where e^{β_i} indicates the ranking score of the method i. We estimate $\beta = \beta_1, \ldots, \beta_6$ by minimizing the following negative log-likelihood using gradient descent

$$L(\beta) = -\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \log w_{i,j} P(i > j) \tag{C.2}$$

where $w_{i,j}$ indicates the total numbers that i is better than j. The "prefer score" in Tab. 3 of main text refers to e^{β_i}.

Appendix D. More Results

In this part, we provide more photo-to-sketch results for CUFS in Fig. F.18), CUFSF in Fig. F.19) and natural images of VGGFace in Fig. F.20).

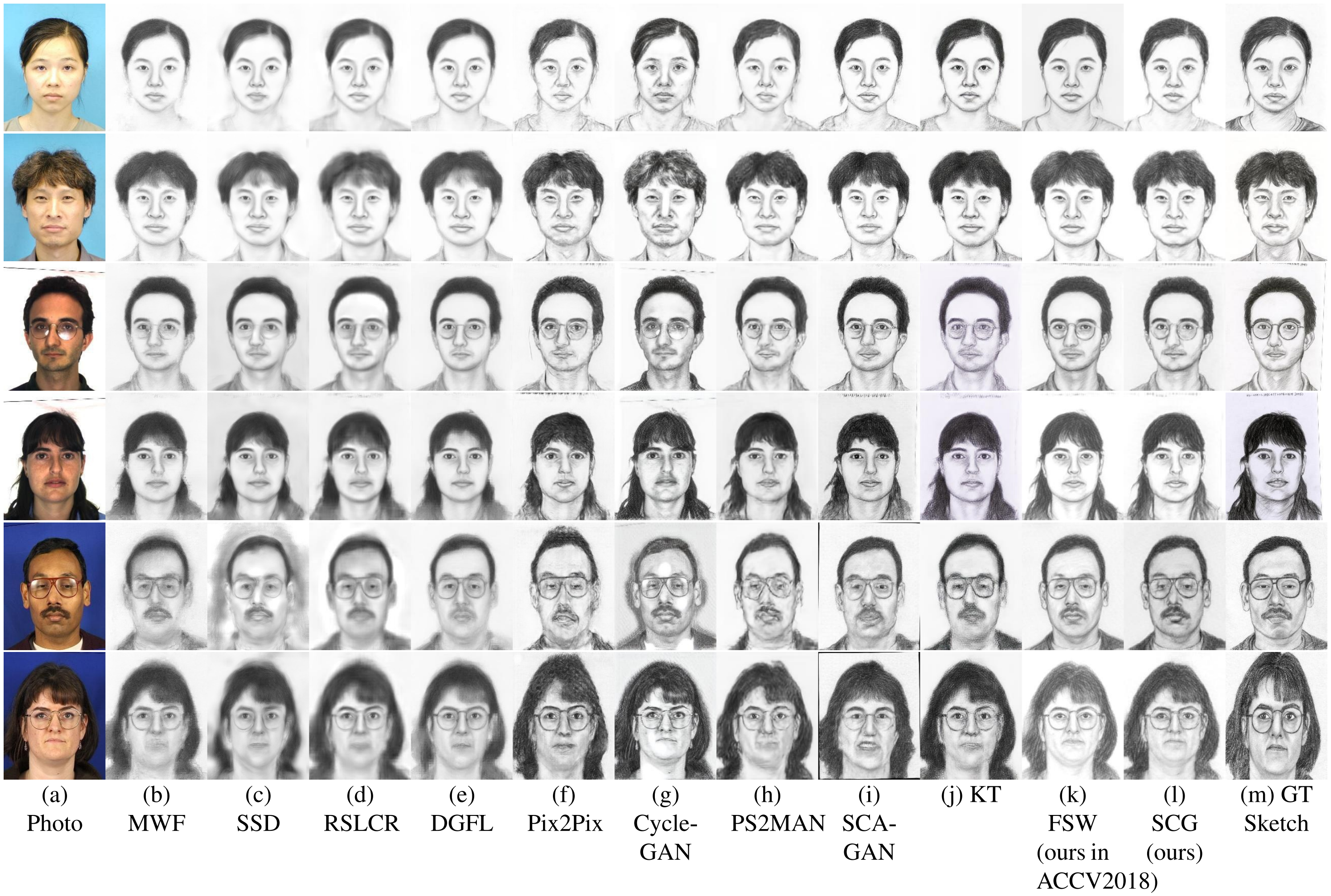
Appendix E. Limitations

Although the proposed model shows good generalization ability to images in the wild, it cannot generate unknown structures which are not included in the small reference dataset. For example, SCG fails to generate the teeth in the last row of Fig. F.20. Note that existing methods cannot even produce pleasant results for these natural images. The proposed SCG also cannot generalize to sketches with different styles, such as sketches drawn in thick lines Yi et al. (2019, 2020b,a). It is quite challenging to synthesize satisfactory sketches with different styles using the same model. In a word, the above two problems are difficult to be solved with current small face sketch datasets, and we will leave them to future work.

Appendix F. Links to public codes

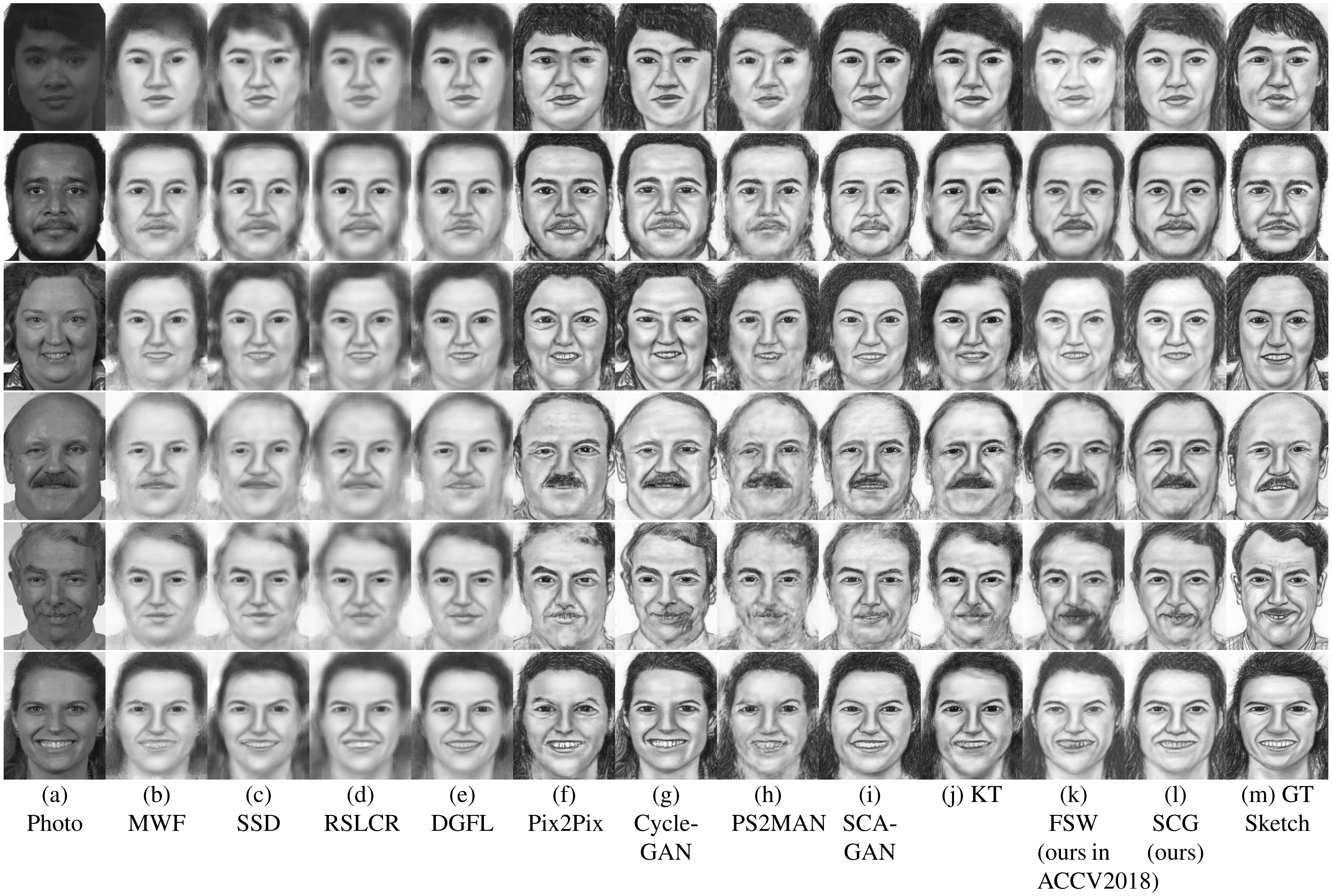
We also provide links to the public codes used in our experiments below:

- SSD: http://www.cs.cityu.edu.hk/~yibisong/eccv14/index.html
- RSLCR: http://www.ihitworld.com/RSLCR.html
- Pix2Pix-GAN: https://github.com/phillipi/pix2pix
- Cycle-GAN: https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix
- PS2-MAN: https://github.com/lidan1/PhotoSketchMAN



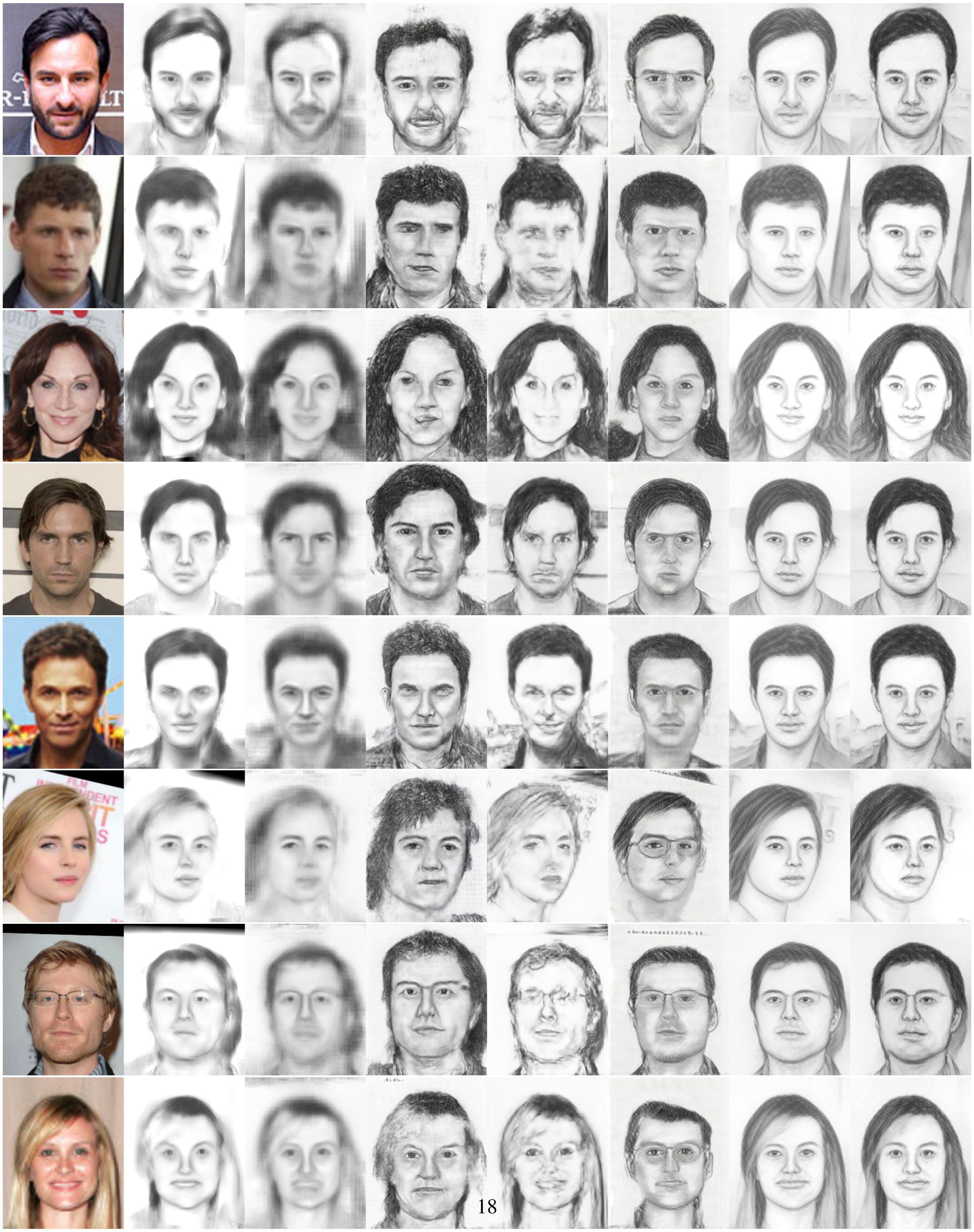
(a) Photo (b) MWF (c) SSD (d) RSLCR (e) DGFL (f) Pix2Pix (g) Cycle-GAN (h) PS2MAN (i) SCA-GAN (j) KT (k) FSW (ours in ACCV2018) (l) SCG (ours) (m) GT Sketch

Figure F.18: More qualitative results for photo-to-sketch translation on CUFS test dataset.



(a) Photo (b) MWF (c) SSD (d) RSLCR (e) DGFL (f) Pix2Pix (g) Cycle-GAN (h) PS2MAN (i) SCA-GAN (j) KT (k) FSW (ours in ACCV2018) (l) SCG (ours) (m) GT Sketch

Figure F.19: More qualitative results for photo-to-sketch translation on CUFSF test dataset.



(a) Photo (b) SSD (c) Fast-RSLCR (d) Pix2Pix (e) PS2MAN (f) Cycle-GAN (g) FSW (ours in ACCV2018) (h) SCG (ours)

Figure F.20: More qualitative results for photo-to-sketch translation in the wild on VGG test dataset.