Deep Facial Synthesis: A New Challenge

Deng-Ping Fan, Ziling Huang, Peng Zheng, Hong Liu, Xuebin Qin, and Luc Van Gool

Abstract—The goal of this paper is to conduct a comprehensive study on the facial sketch synthesis (FSS) problem. However, due to the high costs in obtaining hand-drawn sketch datasets, there lacks a complete benchmark for assessing the development of FSS algorithms over the last decade. As such, we first introduce a high-quality dataset for FSS, named **FS2K**, which consists of 2,104 image-sketch pairs spanning three types of sketch styles, image backgrounds, lighting conditions, skin colors, and facial attributes. FS2K differs from previous FSS datasets in difficulty, diversity, and scalability, and should thus facilitate the progress of FSS research. Second, we present the largest-scale FSS study by investigating 139 classical methods, including 24 handcrafted feature based facial sketch synthesis approaches, 37 general neural-style transfer methods, 43 deep image-to-image translation methods, and 35 image-to-sketch approaches. Besides, we elaborate comprehensive experiments for existing 19 cutting-edge models. Third, we present a simple baseline for FSS, named **FSGAN**. With only two straightforward components, *i.e.*, facial-aware masking and style-vector expansion, FSGAN surpasses the performance of all previous state-of-the-art models on the proposed FS2K dataset, by a large margin. Finally, we conclude with lessons learned over the past years, and point out several unsolved challenges. Our open-source code is available at https://github.com/DengPingFan/FSGAN.

Index Terms—Facial sketch synthesis, facial sketch dataset, benchmark, attribute, style transfer

1 Introduction

Facial sketch synthesis (FSS) aims to generate gray-scale sketches from RGB images of human faces (image-to-sketch, I2S) or the other way around (sketch-to-image, S2I) [1], [2]. FSS is commonly used by law enforcement or in surveillance to assist in face recognition and retrieval, based on a sketch drawing from an eyewitness [1]. It is also used in mobile apps, such as TikTok and Facebook, for entertainment. Besides, it is an attractive topic in digital entertainment [3]. As such, research into FSS has achieved significant progress over the past decade.

Different from other face-related datasets, such as those for face recognition [4], face detection [5], face key-points detection [6], face alignment [7], face synthesis [8], which can be manually labeled by annotators with limited training, face sketch datasets are much more difficult to obtain because only professional artists are able to produce high-quality ground-truths (GTs). Due to the high costs in obtaining professional sketches, existing image-sketch datasets [1], [2], [9] are relatively small with limited diversity. This shortage in datasets has limited the development of this field, especially for data-hungry deep learning models.

In addition, how to evaluate FSS still remains an open question. Structural similarity (SSIM) [10] is one of the most widely used metrics for evaluating image quality, so also it is typically used to evaluate the performance of S2I models. Nevertheless, the characteristics of facial sketches are very different from RGB-based facial images, which makes it difficult to apply the current evaluation metrics to I2S tasks. Therefore, a new objective and quantitative metric, which is also highly consistent with human assessment, is needed for benchmarking the FSS task.

TABLE 1. Comparison with other FSS datasets. Att. = Attributes.

Dataset	Year	Pub.	Total	Train	Test	Att.	Public	Paired
CUFS [1]	2009	TPAMI	606	306	300	×	√	√
IIIT-D [11]	2010	BTAS	231	58	173	×	×	√
CUFSF [12]	2011	CVPR	1,194	500	694	×	√	√
VIPSL [13], [14]	2011	TCSVT	1,000	100	900	×	×	√
DisneyPortrait [15]	2013	TOG	672	-	-	×	×	√
UPDG [16]	2020	CVPR	952	798	154	×	×	×
APDrawing [9]	2020	TPAMI	140	70	70	×	√	√
FS2K (Ours)	2021	Submit	**2,104**	**1,058**	**1,046**	√	√	√

In [17] and [18], CUFS is divided into 268 and 338 images for training and testing.

In addition, due to the lack of *high-quality datasets* and *proper evaluation metrics*, different FSS models (*e.g.*, [1], [2]) are usually built and tested upon diverse training and testing datasets (sometimes because they want to learn a different style of sketches), as well as with different evaluation methods. Hence, it is difficult to provide fair and comprehensive comparisons. Further, many cutting-edge transformation models (*e.g.*, CycleGAN [20], UNIT [24], Pix2pixHD [19], SPADE [25] DSMAP [26], NICE-GAN [21], DRIT++ [27]) designed for related image-to-image transfer tasks, could potentially be employed in FSS tasks. However, these models lack performance evaluation for this task, again because of the shortage in datasets and evaluation metrics, as mentioned above. Therefore, thorough and extensive comparisons and evaluations of FSS-related models, on a standard FSS dataset with unified evaluation metrics, are long overdue. To this end, we have introduced and maintain an online paper list (https://github.com/DengPingFan/FaceSketch-Awesome-List) to track the progress of this fast-developing field.

1.1 Contributions

In this work, our goal is to solve the discussed issues (*i.e.*, limited datasets, metrics, and benchmarks) and further contribute a new

- *Deng-Ping Fan is with the Computer Vision Lab (CVL), ETH Zürich, Zürich, Switzerland.*
- *Ziling Huang is with the Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan.*
- *Peng Zheng is with the Inception Institute of AI (IIAI), Abu Dhabi, UAE.*
- *Hong Liu is with the Digital Content and Media Sciences Research Division, National Institute of Informatics, Tokyo, Japan.*
- *Xuebin Qin is with the Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE.*
- *Luc Van Gool is with the Computer Vision Lab, ETH Zürich, Zürich, Switzerland, and also with KU Leuven, Leuven, Belgium.*
- *Corresponding author: Xuebin Qin (Email: xuebin@ualberta.ca) & Hong Liu (Email: hliu@nii.ac.jp). Ziling Huang and Peng Zheng share equal contribution.*

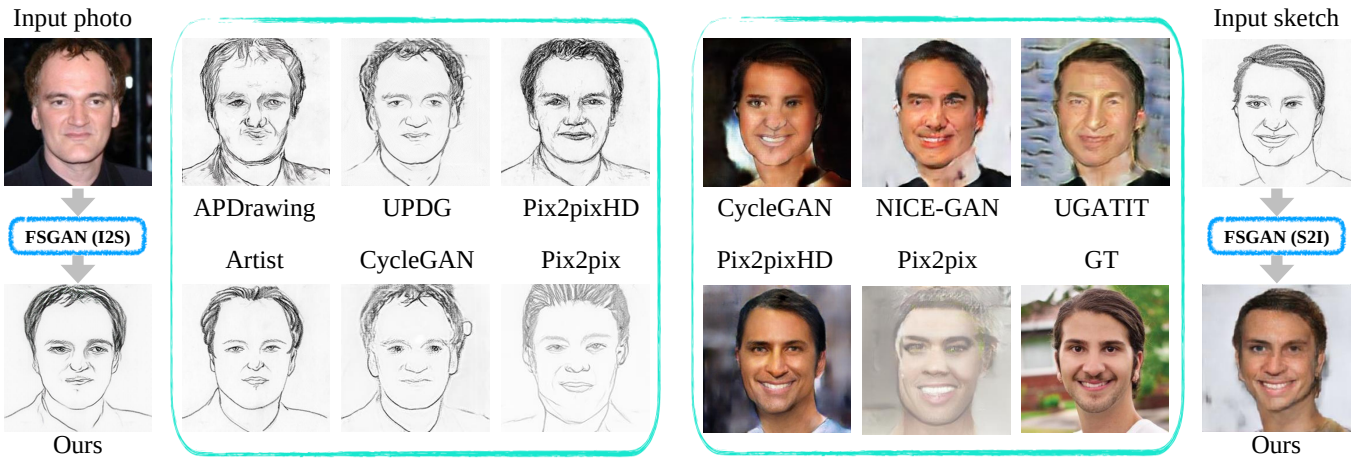


Fig. 1. **Left:** Our FSGAN (I2S) learns from artist drawings and intelligently turns an input photo into a vivid face sketch. In contrast, the five cutting-edge style transfer approaches cannot obtain visually appealing results. Only UPDG [16] and Pix2pixHD [19] perform relatively well, but they generate worse content and style than FSGAN. **Right:** Given a sketch, our FSGAN (S2I) can also transform the input into a vivid facial photo. Meanwhile, the results from the five cutting-edge deep learning models are either structurally damaged (*i.e.*, CycleGAN [20], NICE-GAN [21], UGATIT [22]) or blurry (*i.e.*, Pix2Pix [23]). More results can be found in Fig. 9-12.

challenge for the FSS community. Our main contributions are:

1) **FSS Dataset.** We build a new high-quality FSS dataset, termed **FS2K**. It is the largest (see Table 1) publicly released FSS dataset[1], consisting of 2,104 image-sketch pairs with a wide range of image backgrounds, skin patches, sketch styles, and lighting conditions. In addition, we also provide extra attributes, *e.g.*, *gender*, *smile*, *hair style*, *etc*., to enable deep learning models to learn more detailed cues.
2) **FSS Investigation and Benchmark.** We conduct the largest-scale FSS study, reviewing **139** representative approaches including 24 methods using handcrafted features, 37 models for the general style transfer task, 43 GAN-based works, and 35 I2S transfer algorithms. Based on our FS2K, we introduce the SCOOT metric [29] and conduct a rigorous evaluation of 19 state-of-the-art (SOTA) models from the perspective of content and style.
3) **FSS Baseline.** We design an efficient GAN-based baseline, termed **FSGAN**, which consists of two simple core components, *i.e.*, facial-aware masking and style-vector expansion. The former is utilized to restore details of the facial components while the latter is adopted to learn different styles of the face. FSGAN serves as a unified baseline model for both I2S and S2I tasks on our newly built FS2K dataset. Our code is available at https://github.com/DengPingFan/FSGAN.
4) **Discussions and Future Directions.** In addition to an overall performance assessment, we also conduct an attribute-level evaluation, present detailed discussions, and explore some promising future directions.

1.2 Organization

The remainder of this paper is organized as follows. Sec. 2 provides related works. Sec. 3 introduces our newly built dataset. Sec. 4 illustrates the details of the baseline. In Sec. 5, we provide quantitative and qualitative evaluations of 19 state-of-the-art models on both the S2I and I2S task. In addition, we also offer deeper analyses of the benchmarked models at an attribute level. To better understand the contribution of each component in our baseline, in Sec. 5.4, we implement two variants of our FSGAN as an ablation study. Further, in Sec. 6, some observations and potential directions based on the proposed FS2K are discussed. Finally, concluding remarks are given in Sec. 7.

1. Establishing an FSS dataset drawn by professional artists is more challenging than other face datasets, *e.g.*, face attribute datasets [28], which is why the largest existing FSS dataset, *i.e.*, CUFSF [12], has only ∼1K images in the past 13 years. Although FS2K is only ∼2 times larger than CUFSF, we still took one year to create such a high-quality dataset.

2 Related Works

In this section, we first conduct a complete literature review of the existing FSS datasets. Then, in the second part, we discuss the taxonomy of facial synthesis and highlight particularly innovative and successful approaches for this task, including traditional facial synthesis, neural style transfer, image-to-image translation, and deep facial synthesis. The taxonomy of deep facial synthesis is shown in Fig.4. A summary of the models is provided in Table 2, where we describe their key innovations, datasets, code links, and citation information, *etc*.

2.1 Dataset

We first outline four classical datasets for the FSS task, *i.e.*, CUFS [1], CUFSF [12], VIPSL [14], and IIIT-D [30], and three portrait sketching datasets [9], [16], which are the basis for building most FSS models [31].

CUFS [1] is one of the earliest and most commonly used datasets. It contains 606 photo-sketch pairs, which include 123 samples from the AR face database [32], 188 samples from the CUHK student database, and 295 samples from the XM2VTS database [33]. For each sample, a sketch drawn by an artist and a corresponding photo are provided, where each photo is taken in a frontal pose under normal lighting conditions and maintains a neutral expression. All three sub-databases use solid backgrounds, which can be cyan, white, and blue, *etc*. However, real-world scenes are complex and diverse, and it is impossible to guarantee that every photo will be captured in such a fixed environment.

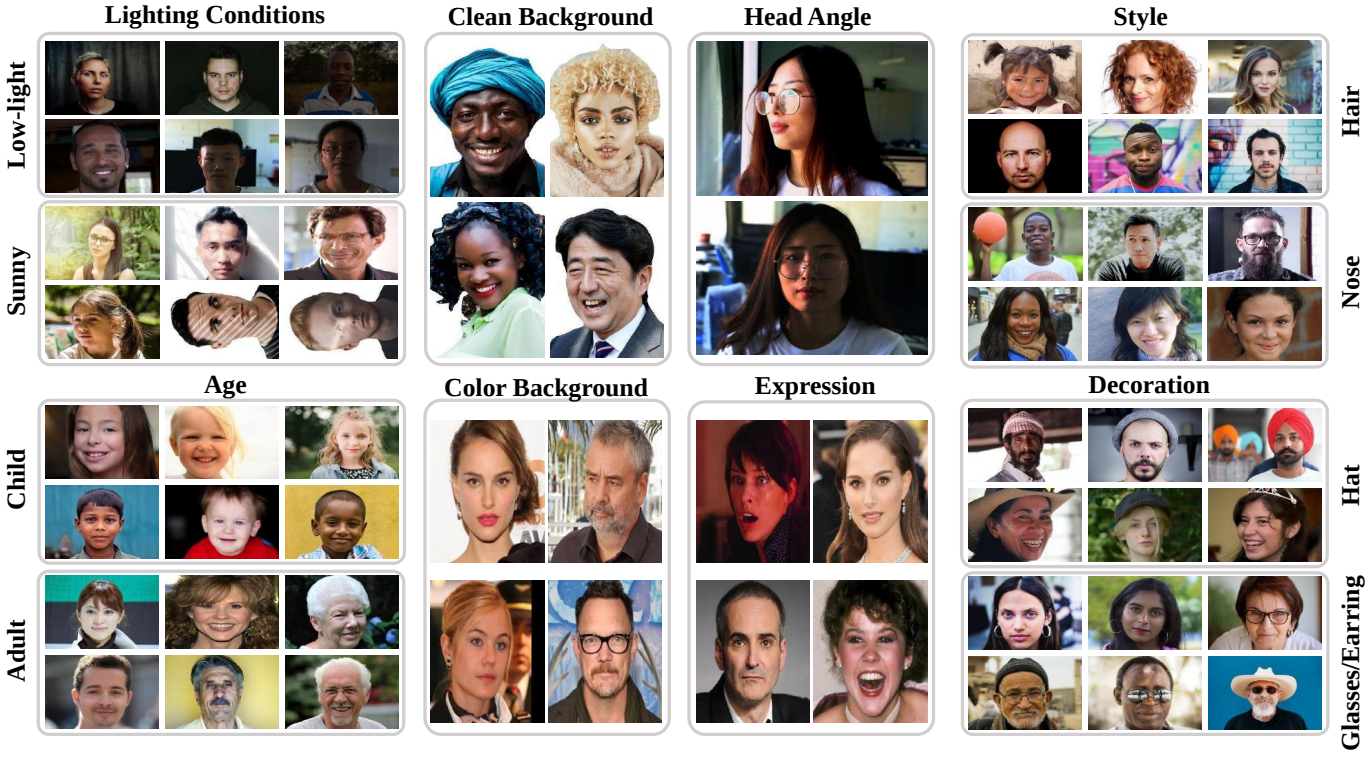


Fig. 2. Representative samples from our FS2K. The collected images depict diverse scenes according to different selection criteria, such as various lighting conditions (*i.e.*, low-light, sunny), ages (*i.e.*, child or adult), backgrounds (*i.e.*, clean or colored), head angles, facial expressions (*e.g.*, serious, smiling, laughing), hair styles (*e.g.*, black, blonde, long, short), and accessories (*i.e.*, hat or earrings).

Besides, all the sketches in this dataset were created by the same artist, so they are of limited style.

CUFSF [12] is another commonly used database for assessing the performance of FSS models. It contains 1,194 photo-sketch pairs, collected from the FERET database [34]. All sketches were drawn by an artist after viewing the corresponding photo. This dataset has a similar photo collection environment to CUFS, but is more challenging. Because the photos in the dataset undergo illumination changes, each face has low contrast with the background, and each sketch contains exaggerated shapes.

VIPSL [14] contains 200 face photos collected from the FRAV2D [35], FERET [34] and Indian face databases [14]. Different from CUFS and CUFSF, VIPSL has five sketches for each face, drawn by five artists with different styles, while viewing the same photo under the same conditions as CUFS.

IIIT-D [30] consists of three types of sketch databases, including a viewed sketch database, semi-forensic sketch database, and forensic sketch database. All photos are derived from the CUHK student database and IIIT-Delhi Sketch database [11]. The first viewed sketch database contains 238 sketch-digital image pairs, with all sketches drawn by professional artist based on a given photo. The second sub-database has 140 sketch-face image pairs, where all the sketches are drawn by memory after the artist has observed the corresponding photo. The third forensic sketch database consists of 190 sketches which are drawn by a sketch artist according to the description of an eyewitness, based on their recollection of a crime scene. IIIT-D contains multiple styles of sketch portraits, making it more challenging. However, obtaining forensic sketches is quite difficult, since they are usually derived from law enforcement investigations.

Portrait Sketching Dataset. Yi *et al.* [9], [36] provided two datasets that simulate artistic portrait drawing (APDrawing). The first dataset [9] contains 140 pairs of face photos and corresponding sketch portraits, all drawn by a single portrait artist. This was later extended a larger dataset in [36], which has 952 face photos and 625 portrait sketches. For the collected photos, 220 of them are from three famous painters, and the remaining 212 photos are from a photography website.[2] It is worth noting that the photos and portraits in this dataset are not paired. Finally, Disney Research published a portrait dataset [15] composed of 24 faces from the face database [37] and 672 sketches from seven artists under four levels of abstraction. Besides, they also provided each stroke as a transparent bitmap to be used later to create new sketches.

Unlike existing datasets, we provide a more challenging, high-quality, attribute-annotated dataset, which is currently the largest dataset of facial sketches. The new dataset contains a total of 2,104 pairs of photos and sketches, 1,058 of which are used for model training and the remaining for evaluation. The strengths of our dataset include, multiple drawing styles, highly accurate alignment between sketches and photos, multiple attribute information, complex backgrounds, *etc*. Detailed comparisons of the datasets are shown in Table 1.

2.2 Traditional Facial Synthesis

In the early years, researchers used heuristic image transformations to interactively or automatically synthesize facial sketches [3], [38]–[42]. However, these methods tend to generate artificial and inexpressive sketches that lack the artistic style. Therefore, in recent years, more attention has been focused on learning-based facial synthesis schemes, whose taxonomy is shown in

2. https://vectorportal.com/

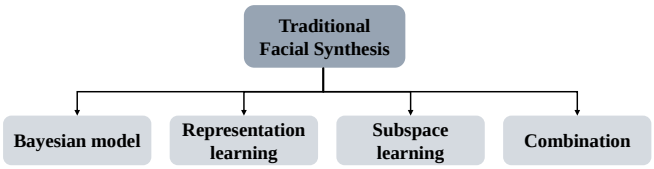


Fig. 3. A taxonomy of traditional facial synthesis and the representative methods.

Fig. 3. These can be categorized into Bayesian inference models, subspace learning models, representation learning models, *etc*.

2.2.1 Bayesian Inference Models

Bayesian inference is a classical machine learning method, which has been widely used in FSS [43]. In [44], Chen *et al.* were the first to use a learning-based algorithm to generate facial sketches automatically. Later, the embedded hidden Markov model [45] was used to model the non-linear relationships in photo-sketch pairs followed by a selective ensemble strategy to generate facial sketches [46]. Wang and Tang [1] followed a similar idea but considered face structures across different scales, using a multi-scale Markov random field (MRF) to build the relationships between photo-sketch pairs. Xu *et al.* [47] proposed a hierarchical compositional model that considers the regularity and structural variation of faces. These methods have made great progress in generating sketches, but they only consider simple controlled conditions, ignoring variations in lighting and pose. Zhang *et al.* [48] addressed this by simultaneously considering patch matching, intensity compatibility, gradient compatibility and shape priors, resulting in better visual effects. However, MRF-based models have two major drawbacks: (1) They struggle to synthesize unseen facial information; (2) Their optimization is NP-hard. Zhou *et al.* [49] used Markov weight fields and cascaded decomposition to build a robust facial synthesis system, which uses a linear combination of candidate patches to approximate new sketch patches. Wang *et al.* [50] built a non-parametric model to transform a photograph into a portrait painting, where an MRF is used to enhance the spatial coherence of the style parameters, and an active shape model and graph-cut model are used to learn the local information of facial features. Wang *et al.* [51] presented a transductive learning method to synthesize facial sketches, which employs an on-the-fly optimization process to minimize the loss on the given test samples. Peng *et al.* [52] designed a superpixel method built on the Markov model, which improves the flexibility without dividing the photo into regular rectangular patches. Then, they not only used the Markov network to model the relationships between image patches, but also retained many visual aspects of the cues (such as edges) through multiple visual features [53].

2.2.2 Representation Learning Models

Ji *et al.* [54] demonstrated that personalized features are not effectively captured through the synthesis process. As such, several works [54]–[56] use different regression models, such as k-NN [54], Lasso [54], multivariate output regression [55] and support vector regression [56], to build the transformation between photos and sketches. To improve the quality of the generated facial sketches, Wang *et al.* [13], [14] used the local linear embedding (LLE) [57] to estimate an initial sketch or photo, and then introduced a sparse multi-dictionary representation model that can focus on high-frequency and detailed information. However, most representation-based models assume that same representations are shared by the source input and the target output, which limits the local structures of a particular style in the synthesis process. To relax this constraint, Wang *et al.* [58] introduced a semi-coupled dictionary learning method, in which a linear transformation is used to bridge the gap between two different domain-specific representations. Gao *et al.* [14] also took a two-step algorithm [56] into consideration, presenting a selection scheme to generate the initial pseudo-images and introducing a sparse-representation-based enhancement (SRE) to synthesize sketches.

2.2.3 Subspace Learning Models

Tang and Wang [59]–[61] proposed a series of example-based approaches based on the linear eigen-transformation method. These methods are global linear systems and they cannot fully explain the relationships between photo-sketch pairs, because such a transformation is not a simple linear relationship. Liu *et al.* [62] used the LLE to handle this problem, making photo and sketch patches have manifolds with similar local geometric shapes in two different image spaces. However, the pseudo-image generation and the representation learning are divided into two independent processes, leading to sub-optimal results. Huang and Wang [63] proposed a joint learning framework, which contains domain-specific dictionary learning and common subspace learning.

2.2.4 Combination Models

Berger *et al.* [15] were the first to use a data-driven method to study the style and abstraction of facial sketches. Song *et al.* [64] introduced a real-time FSS method, which first uses a k-NN algorithm to find the top-k similar local patches, then uses linear combination to compute the corresponding sketch image, and finally uses image denoising technology to enhance the visual quality. The real-time model in [64] is time-consuming due to the k-NN process, so Wang *et al.* [65] addressed this problem by replacing offline random sampling with an online scheme that is further combined with a recognition weight representation. Most existing traditional methods are fully dependent on the scale of the training data, so Zhang *et al.* [66] presented a robust model trained on a template stylistic sketch. The model includes representation learning, MRF, and a cascaded model. Li *et al.* [67] proposed a free-hand sketch synthesis method, combining a perceptual grouping model with a deformable stroke model. The work in [68] introduces an adaptive learning method that combines representation learning and a Markov network.

2.3 Neural Style Transfer

Recently, neural style transfer (NST), which aims at generating visually appealing images via the neural networks, has been introduced into the FSS task [69]. Specifically, NST is used to render a content image in different styles. NST methods can be categorized into optimization-based online methods and model-based offline methods.

2.3.1 Optimization-Based Online Methods.

In optimization-based online NST methods, a given input image is iteratively optimized with the goal of matching the desired CNN features, including both the photo's content information and artistic style information. Gatys *et al.* [133], [134] made the first contribution to this field, using a classical CNN (*i.e.*, VGG [233])

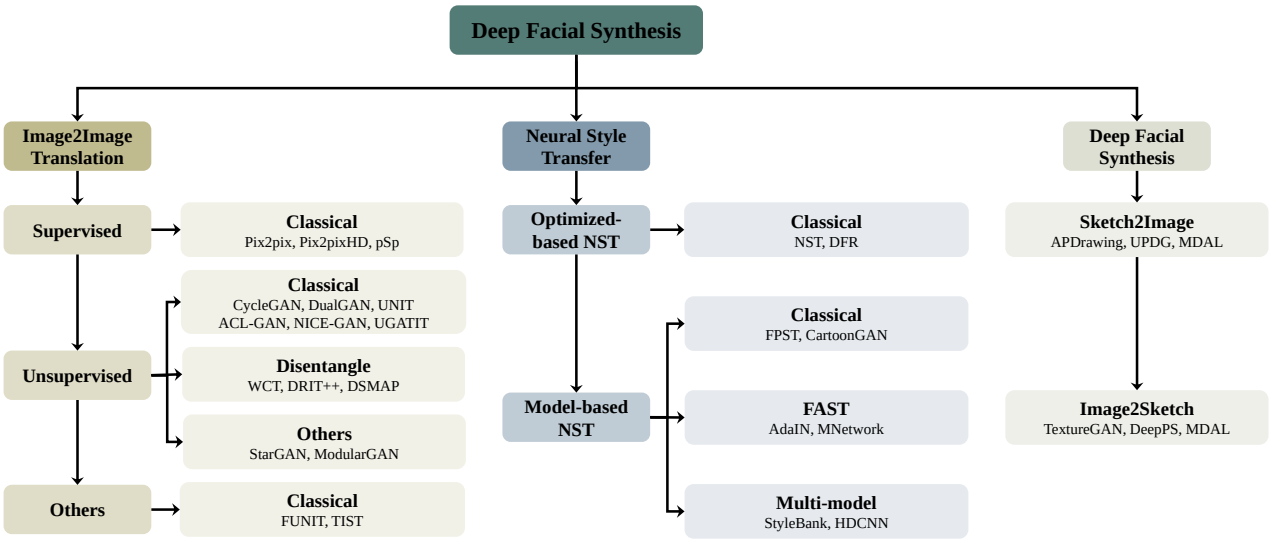


Fig. 4. A taxonomy of deep facial synthesis and the representative methods.

to render an image with famous painting styles. Later, Li and Wand [135] demonstrated that parametric NST methods tend to neglect the spatial layout, leading to less visually plausible results. They therefore proposed a non-parametric neural method that combines a deep neural network with classical MRF-based texture synthesis. Selim *et al.* [139] extended the classical NST [134] to portrait painting, using the gain map to constrain the spatial information, with the aim of preserving the facial structures while transferring the target style. Berger and Memisevic [141] proposed an approach that makes use of long-range consistency constraints to preserve the global symmetry properties, and applies texture generation to in-painting. However, most NST methods output blurred style images, which are therefore not photo-realistic. To address this, Luan *et al.* [144] introduced photo-realism and segmentation regularization into the classical NST model [134], constraining the transformation from the input to the output via the local affinity in the color space. Liao *et al.* [154] presented a novel weakly supervised NST model, which is based on patch matching and does not rely on a large-scale training set. Later, Gu *et al.* [155] theoretically proved that reshuffling deep features can minimize both the global and local style losses simultaneously. They therefore proposed an objective function that combines reshuffling loss with a classical content loss to help extract more powerful features. Recently, Men *et al.* [159] proposed a common framework for interactive texture transfer with structure guidance. Their model dynamically implements the synthesis process using multiple channels, including structure extraction, structure propagation, and guided texture transfer. Besides, StyleGAN [120] uses a latent space to maintain consistent results for image synthesis. However, it is difficult to achieve promising results under the given conditions. Abdal *et al.* [167] integrated the classical NST [133], [134] into the StyleGAN model, using NST to project the input image into the latent space defined in StyleGAN.

2.3.2 *Model-Based Offline Methods.*

Optimization-based online methods achieve satisfactory results, but there are still some limitations. One major drawback is the slow computational speed and high cost because of the online iterative optimization. To address this issue, several works introduce a feed-forward network to mimic the optimization objective of style transfer [69].

End-to-End Models. End-to-end models can be divided into those that design a basic deep neural architecture and those that introduce a new loss function. For basic architectures, Johnson *et al.* [136] took advantage of the benefits of the neural network and optimization-based NST model and proposed a method for training a feed-forward network using a new perceptual loss. TextureNet [138] follows a similar idea, but with a different neural network architecture. Both [136] and [138] are real-time style transfer methods. Chen and Schmidt [140] introduced a style swap operation to exchange the patches with visual context and those with style, further formulating a new optimization objective that aims to learn an inverse neural network for arbitrary style transfer. In terms of methods based on loss function, CartoonGAN [156] was presented to transfer real-world photos into cartoon-style images. It consists of two novel loss functions that were designed to preserve clear edge information and cope with the stylistic difference between photos and cartoons, respectively. Instead of employing the Gram loss [134], which is widely used in NST, Mechrez *et al.* [162] built their NST model on a new contextual loss without data alignment that compares similar local semantic regions while considering the context of the entire image. The previous models are limited by the training data, so a style-aware content loss [163] was used to train an auto-encoder that can overcome this issue.

Feature-Based NST. Although the model-based methods can produce stylized images faster than the optimization-based ones, they require multiple deep networks with a higher number of parameters. To address this, several researchers have begun using a small number of parameters to characterize each style, *i.e.,* changing the parameters in the normalization layer for style transfer. Dumoulin *et al.* [142] made the interesting observation that normalization layers can reflect the statistical properties of different styles. Therefore, they scaled and shifted the parameters in these layers, while keeping the convolutional parameters unchanged, to obtain better NST. Further, they introduced flexible conditional instance normalization, enabling style transfer to be achieved by simply changing the normalization parameters online.

TABLE 2. Summary of popular related works. These can be categorized into four types: *Traditional Facial Synthesis*, *General Neural Style Transfer*, *Deep Image-to-Image Translation*, and *Deep Image-to-Sketch Synthesis*. **Publ.**: Publication information. **Year**: Publication year. **Code**: The link of the corresponding open resources. **Component**: The key components of each model. **Dataset**: A = TU-Berlin Sketch Dataset [70], B = Disney Portrait Dataset [15], C = FERET [71], D = AR [32], E = Self-Collected, F = MSCOCO [72], G = ImageNet [73], H = MITPortraits [74], I = CelebA [28], J = Sintel [75], [76], K = DAVIS [77], L = DTD [78], M = Places365 [79], N = Pascal3D+ [80], O = FlyingThings3D [81], P = Wikiart [82], Q = Cityspace [83], R = CMP Facades [84], S = Edge2photo [85], [86], T = Sketch2photo [87], U = Day2night [88], V = MNIST [89], W = USPS [90], X = SVHN [91], Y = CUFS [1], CU = CUFSF [12], Z = Caltech-200 Bird [92], AA = Oxford-102 Flower [93], AB = CIFAR10 [94], AC = CelebAHQ [95], AD = NYU Indoor RGBD dataset [96], AE = ADE20K [97], AF = Helen Face Dataset [98], [99], AG = FERET [34], AH = FaceScrub [100], AI = RaFD [101], AJ = Emotionet [102], AK = QMUL-Shoe-Chair-V2 [103], AL = QuickDraw dataset [104], AM = Standford Dogs [105], AN = VGG Face [106], AO = MT Dataset [107], AP = Yosemite [20], AQ = cat2dog [27], AR = Flickr Landscapes [25], AS = APDrawing Dataset [2], AT = Anime Faces of Getchu [108], AU = Selfie2anime [22], AV = hourse2zebra [20], AW = photo2vangogh [20], AX = photo2portrait [27], AY = BRATS Dataset [109], AZ = Dayton Dataset [110], BA = CVUSA Dataset [111], BB = Ego2Top Dataset [112], BC = Animal Faces [113], BD = Birds [114], BE = Flowers [93], BF = Foods [115], BG = GTA5 [116], BH = Berkeley Deep Drive [117], BI = SYNTHIA dataset [118], BJ = UPDG [16], BK = DeepFashion [119], BU = FFHQ [120], BV = DIV2K [121], BW = LHI [122], BX = VIPSL [13], BY = IIIT-D [11], BZ = Map2Aerial [23], CA = ColorMNIST [123], CB = StandfordCars [124], CC = BAM [125], CD = YFCC100M [126], CE = UTKFace [127], CF = OxfordCats [128], CH = LSUN [129], CI = Car [130], CJ = High-quality Animal Face [131], CK = Creative Sketch Dataset [132]. **Assist.**: Assistant Information, *e.g.*, Bm.= Background map, Sm.= Segmentation map, Fl. = Facial landmark, Sv. = Style vector, Cm. = Color map, Attri. = Facial Attribute, Km. = Keypoint map, Td. = Semantically relevant text, Au. = Facial action units, Tp. = Texture patch. **Cite.**: Google citation statistics are from 2021-12-21.

#	Model	Publ.	Year	Code	Component	Dataset	Assist.	Cite.
					Traditional Facial Synthesis			
1	EFSGNS [44]	ICCV	2001	-	Active Shape Model, Non-parametric Sampling	E	-	157
2	Nonlinear [62]	CVPR	2005	-	Local Linear Preserving, Eigentransform	Y	-	386
3	E-HMM [46]	TCSVT	2008	-	Embedded Hidden Markov Model, Selective Ensemble	Y	-	163
4	HCM [47]	PAMI	2008	-	Graph, Minimum Description Length	C, D, BW, E	-	92
5	MRF [1]	PAMI	2009	Code	Multi-scale Markov Random Fields	Y	-	843
6	LPR [48]	ECCV	2010	-	Local Evidence Function, Patch Matching, Shape Prior, MRF	Y	-	118
7	LRM [54]	ICIG	2011	-	Local Regression, kNN	Y	-	19
8	MOR [55]	HCII	2011	-	Multivariate Output Regression	Y	-	20
9	MDSR [13]	ICIG	2011	-	LLE, Dictionary Learning, Sparse Representation	Y, BX	-	54
10	SVR [56]	ICIP	2011	-	Support Vector Regression	Y, BX	-	39
11	SCDL [58]	CVPR	2012	-	Sparse Coding, Semi-coupled Dictionary Learning	Y	-	596
12	MWF [49]	CVPR	2012	-	Markov Weight Fields, Cascade Decomposition	Y, E	-	165
13	SR [14]	TCSVT	2012	-	Sparse Neighbor Selection, Sparse-Representation Enhance	Y, BX	-	179
14	SAPS [15]	TOG	2013	-	Edge Detection, Shape Deformation	B	-	105
15	FESM [50]	BMVC	2013	-	Markov Random Field, Graph-cut	E	-	22
16	Transductive [51]	TNNLS	2013	-	Probabilistic graph model,Transductive Learning	Y, CU	-	162
17	CDFSL [63]	ICCV	2013	-	Coupled Dictionary and Feature Space Learning	Y	-	173
18	REB [64]	ECCV	2014	Project	kNN, Linear Estimation, Sketch Denoising	Y, D	-	121
19	RobustStyle [66]	TIP	2015	-	Sparse Representation, Multi-scale Selection	Y, E	-	47
20	SPP [52]	TCSVT	2015	Project	Superpixels, Markov Networks	Y, CU, BY	-	43
21	MR [53]	TNNLS	2016	-	Markov Networks, Edge Enhancement, Alternating Opt.	Y, BY	-	101
22	DSM [67]	IJCV	2017	Project	Perceptual Grouping, Deformable Stroke Model	A, B	-	31
23	AR [68]	NC	2017	-	Adaptive Representation, Markov Networks	Y	-	8
24	RS [65]	PR	2018	-	Offline Random Sampling, Locality Constraint	Y, CU	-	90
					General Neural Style Transfer			
25	NST [133], [134]	CVPR	2016	Github	Parametric Texture Mode, Representation Inversion	E	-	3434
26	CNNMRF [135]	CVPR	2016	Github	MRF Priors & CNN	E	-	559
27	FNS [136]	ECCV	2016	Github	Image Transformation Network, Loss Network, Perceptual Loss	F	-	6240
28	MGANs [137]	ECCV	2016	Github	Markovian Deconvolutional Networks, Markovian GAN	E, I	-	1096
29	TextureNet [138]	ICML	2016	Github	Generator Network, Descriptor Network,	E	-	736
30	EDSC [139]	TOG	2016	Project	NST [134], Gain map, Two-step Method	H	Bm.	155
31	FPST [140]	NeurIPSW	2016	Github	CNN, Style Swap, Inverse Network	F, P	-	243
32	ILC [141]	ICLR	2017	Github	Transformed Gramian	E	-	35
33	CIN [142]	ICLR	2017	Github	Conditional Instance Normalization	G, E	-	748
34	CPF [143]	CVPR	2017	Github	Spatial Control, Scale Control	E	-	319
35	DPST [144]	CVPR	2017	Github	Photorealism Regularization, Segmentation	E	Sm.	509
36	FFN [145]	CVPR	2017	Github	VGG Loss Network, Diversity Loss, Incremental Learning	L	-	181
37	StyleBank [146]	CVPR	2017	Code	Encoder-Decoder Network, Style Bank Layer	E, F	-	323
38	ITN [147]	CVPR	2017	Github	Instance Normalization, Julesz Generator Network	E	-	497
39	HDCNN [148]	CVPR	2017	Github	VGG Loss Network, Style/Enhance/Refine Subnet	E, F	-	119
40	AdaIN [149]	ICCV	2017	Github	Adaptive Instance Normalization	F, P	-	1791
41	CI [150]	ICCV	2017	Github	RCNN [151], Temporal Consistency Loss	J, K, E	-	92
42	DNLRF [152]	ICCV	2017	-	Lightweight Feature Reconstruction, Feature Decoder	F, P	Sm.	42
43	WCT [153]	NeurIPS	2017	Github	Multi-level Stylization, Whitening and Coloring Transforms	F, L	-	489
44	VAT-DIA [154]	TOG	2017	Github	Weakly-supervised Image Analogy, Nearest-Neighbor Field	G, E, N	-	326
45	DFR [155]	CVPR	2018	Github	Neural Feature Reshuffle, Reshuffle Loss	G, E	-	92
46	CartoonGAN [156]	CVPR	2018	Github	GAN, Semantic Content Loss, Edge-promoting Loss	E	-	195
47	MNetwork [157]	CVPR	2018	Github	Meta Networks, Image Transform Networks	F, P	-	73
48	Avatar-net [158]	CVPR	2018	Github	Style Decorator, Style-augmented Hourglass Network	E, F, J	-	129
49	CFITT [159]	CVPR	2018	Github	PatchMatch, Guided Texture Transfer	E	Sm.	15
50	SSC [160]	CVPR	2018	Github	Style/Content Encoder, Mixer Network, Decoder Network	E	-	85
51	SNST [161]	CVPR	2018	-	Disparity Loss, StyleNet, DispOccNet	F, O, J	-	78
52	CL [162]	ECCV	2018	Github	Contextual Loss	E	-	199
53	SACL [163]	ECCV	2018	Github	Encoder-decoder network, Style-Aware Content Loss	P, M	-	101
54	ARF [164]	ECCV	2018	Github	Stroke Pyramid, VGG Loss Network, Stroke Decoder	E, F	-	64
55	LinearTransfer [165]	CVPR	2019	Github	Linear Transformation, Spatial Propagation Network, Whitening	F, P	-	65
56	SANet [166]	CVPR	2019	Demo	AdaIN, Non-local Block, Identity Loss	F, P	-	77
57	Image2StyleGAN [167]	CVPR	2019	Github	StyleGAN, Embedding	AC, BU	-	299
58	DIN [168]	AAAI	2020	-	Dynamic Instance Normalization	F, P	-	36

TABLE 3. Summary of popular related works. Please refer to Table 2 for more detailed descriptions.

#	Model	Publ.	Year	Code	Component	Dataset	Assist.	Cite.
					Deep Image-to-Image Translation			
59	RST [169]	CVPR	2021	Github	Differentiable Renderer, Brushstrokes Parameterization	E	-	5
60	LPN [170]	CVPR	2021	Github	Drafting Network, Revision Network, AdaIN	F, P	-	3
61	pSp [171]	CVPR	2021	Github	StyleGAN, Disentangled Latent Feature, Map2Style	AC, BU	-	117
62	Pix2pix [23]	CVPR	2017	Github	Generator with Skip, PatchGAN	G, Q, R, S, T, U, BZ	-	11539
63	SisGAN [172]	ICCV	2017	Github	Adaptive Loss, GAN-based Encoder-Decoder	Z, AA	Td.	150
64	CycleGAN [20]	ICCV	2017	Github	Map Functions and Discriminators, Cycle Consistency Loss	G, Q, R, S, T, U, AV, AW	-	11024
65	DualGAN [173]	ICCV	2017	Github	Trained in Closed Loop, Reconstruction Loss	R, U, Y, CU, BZ, E	-	1396
66	DiscoGAN [174]	ICML	2017	Github	GAN with a Reconstruction Loss	CI, K, I, AH, S	-	1527
67	DTN [175]	ICLR	2017	Code	Multi-class GAN Loss, f-Constancy Component	I, V, X, E	-	862
68	BicycleGAN [176]	NeurIPS	2017	Github	cVAE-GAN, cLR-GAN	R, S, U, BZ	-	1027
69	UNIT [24]	NeurIPS	2017	Github	Common Latent Space, VAEs, Cycle-consistency, GAN	G, I, Q, V, W, X, BI	-	1933
70	DistanceGAN [177]	NeurIPS	2017	Github	CycleGAN, Distance Constraints	I, V, X	-	174
71	TriangleGAN [178]	NeurIPS	2017	Github	cGANs, Adversarially Learned Inference	I, F, S, V, AB	-	120
72	Pix2pixHD [19]	CVPR	2018	Github	Coarse-to-fine Generator, Multi-scale Discriminator	Q, AD, AE, AF	-	2218
73	DA-GAN [179]	CVPR	2018	Code	Deep Attention Encoder, Instance-level Constraints	V, X, Z, AH, E	-	123
74	StarGAN [180]	CVPR	2018	Github	Multi-Domain Classifier, cGAN	I, AI	Sv., Attri.	2172
75	ModularGAN [181]	ECCV	2018	Code	Reusable and Composable Transformer Modules	I, CA	-	58
76	GANimation [182]	ECCV	2018	Github	Attention, Conditional GAN with Action Units	AI, AJ, E	Au.	378
77	SCANs [183]	ECCV	2018	-	Multi-stage Transformations, CycleGAN	G, Q, R, S, U, AV, AW	-	60
78	MUNIT [184]	ECCV	2018	Github	Content/Style Encoder, AdaIN, Decoder	S, T, AP, BI, E	-	1395
79	Elegant [185]	ECCV	2018	Github	Multi-scale Discriminator, Feature Swapping	I	Attri.	118
80	EGSC-IT [186]	ICLR	2019	Github	Exemplar-based AdaIN, Feature Masks, VAE-GAN	V, BG, BH	-	91
81	HarmonicGAN [187]	ICLR	2019	-	CycleGAN, Smooth Regularization	Q, AV, AY	-	29
82	GDWCT [188]	CVPR	2019	Github	Group-wise Whitening and Coloring Transformation	I, K, AQ, AP, CC	-	52
83	SPADE [25]	CVPR	2019	Github	Spatially-Adaptive Normalization, Pix2pixHD	F, Q, AE, AR	Sm.	1120
84	TransGaGa [189]	CVPR	2019	Project	Geometry/Appearance Transformer, Conditional VAE	CC, CD, I, E	-	64
85	MS-GAN [190]	CVPR	2019	Github	Mode Seeking Regularization, Conditional GANs	R, Z, AP, AQ	-	207
86	Selection-GAN [191]	CVPR	2019	Github	Multi-Channel Attention Selection Module	AZ, BA, BB	Sm., Km.	111
87	AttentionGAN [192]	IJCNN	2019	Github	Attention-Guided Generator and Discriminator	D, I, AI	-	49
88	FUNIT [113]	ICCV	2019	Github	Multi-task Discriminator, Few-shot Image Translator	BC, BD, BE, BF	-	316
89	U-GAT-IT [22]	ICLR	2020	Github	Attention map, Adaptive Layer-Instance Normalization	AU, AV, AW, AX	-	173
90	CrossNet [193]	WACV	2020	-	Latent Cross-Translation and Cycle-Consistency	AP, AV	-	-
91	StarGAN v2 [131]	CVPR	2020	Github	Multi-Domain Discriminator, Mapping/Style Encoder	AC, CJ	Sv.	375
92	HiDT [194]	CVPR	2020	Github	Adaptive U-Net with AdaIN and Upsampling	E	-	30
93	NICE-GAN [21]	CVPR	2020	Github	Encoder with Discriminator	AV, AP, AW, AQ	-	40
94	SEAN [195]	CVPR	2020	Github	Semantic Region-Adaptive Normalization	Q, AC, AE	Bm.	137
95	CoCosNet [196]	CVPR	2020	Github	Cross-domain Correspondence, Translation Network	AE, AC, BK	-	65
96	TSIT [197]	ECCV	2020	Github	Multi-scale Feature Normalization, Two-stream Network	Q,AE, AP, AW, BH	-	22
97	DSMAP [26]	ECCV	2020	Github	Domain-specific Content Mappings	AQ, AW, AX	-	9
98	ACL-GAN [198]	ECCV	2020	Github	Adversarial Consistency Loss, MUNIT	I, AU	-	18
99	DRIT++ [27]	IJCV	2020	Github	Disentangled Representation with Cross-cycle Consistency	AP, AQ, AW, AX, I	-	166
100	LLS [199]	ICLR	2021	Github	Online Latent Code Learning	V, AC, G, CE, CF, S, T, BZ	-	2
101	GH-feat [200]	CVPR	2021	Github	Hierarchical Encoder, StyleGAN	G, V, BU, CH	-	16
102	Divco [201]	CVPR	2021	Github	Contrastive Learning, cGANs	R, BZ, AQ, AP	-	6
103	CoCosNet v2 [202]	CVPR	2021	Github	ConvGRU Module, Hierarchical Strategy, PatchMatch	AE	-	9
104	SDEdit [203]	Arxiv	2021	Project	Stochastic Differential Equations, cGANs	I, AC, CH	-	9
					Deep Image-to-Sketch Synthesis			
105	FCRL [204]	ICMR	2015	-	Fully Convolutional Network	Y	-	119
106	DGFL [205]	IJCAI	2017	-	Deep CNNs, Graphic model	Y	-	31
107	Scribbler [206]	CVPR	2017	Project	Encoder-decoder with residual connections, GAN	Y, E	-	390
108	FSSC2F [207]	AAAI	2018	-	U-Net, Probabilistic Graphic Model	Y	-	9
109	TextureGAN [208]	CVPR	2018	Github	Local Texture Loss, VGG Loss, Scribbler	E, S, T	Bm. Tp.	197
110	SCC-GAN [209]	CVPR	2018	Code	Hybrid model, Shortcut Cycle Consistency	AK, AL	-	61
111	ContextualGAN [210]	ECCV	2018	Github	Contextual Loss, Joint Representation, GAN	I, Z, CB	-	63
112	pGAN [211]	IJCAI	2018	Github	UNet, Parametric Sigmoid, CycleGAN	Y, CU	Bm.	18
113	MRNF [212]	IJCAI	2018	-	Markov Random Neural Fields	Y	-	12
114	PS^2-MAN [213]	FG	2018	Github	Multi-Adversarial Networks, CycleGAN	Y, CU	-	87
115	DualT [214]	TIP	2018	-	Deep Features, Intra- and Inter-Domain Transfer	Y	-	42
116	MDAL [215]	TNNLS	2018	Github	Domain alignment, Interpreting by Reconstruction	Y, CU	-	36
117	FAG-GAN [216]	WACVW	2018	-	Attribute Classification, Conditional CycleGAN	I, AG	-	25
118	Geo-GAN [217]	BIOSIG	2018	Github	Geometry Discriminator, CycleGAN	CU, AG	-	14
119	PI-REC [218]	arXiv	2019	Github	Corse-to-Fine, LSGAN, VGG Loss	I, S, T, AT	Cm.	12
120	DLLRR [219]	TNNLS	2019	-	Coupled Autoencoder, Low-rank Representation	Y	-	21
121	Col-cGAN [220]	TNNLS	2019	-	Collaborative Loss, cGAN, Deep Collaborative Nets	Y, CU	-	37
122	CFSS [221]	TIP	2019	-	cGAN, VGG, Feature Selection	Y	-	12
123	KT [222]	IJCAI	2019	-	Knowledge Transfer, Teacher-Student Net	Y, CU	-	11
124	im2pencil [223]	CVPR	2019	Github	Outline and Shading Branch Networks, Pix2pix	E	Sv.	23
125	ISF [224]	ICCV	2019	Project	Shape and Appearance Generators, Two-stage	S, AC, E	-	34
126	APDrawing [2]	CVPR	2019	Github	Hierarchical GAN, DT Loss, Local Transfer Loss	AS	Fl., Bm., Sv.	63
127	APDrawing++ [9]	TPAMI	2020	Github	APDrawing, Line Continuity Loss	AS	Fl., Bm., Sv.	9
128	UPDG [36]	CVPR	2020	Github	Asymmetric CycleGAN, Cycle-consistency Loss	BJ	Fl., Bm., Sv.	14
129	WCR-GAN [225]	CVPR	2020	Github	Cartoon Representation Learning, GAN	F, BU, BV, E	-	20
130	EdgeGAN [226]	CVPR	2020	Project	SketchyCOCO, Divide-and-Conquer strategy	F	Attri.	22
131	DeepPS [227]	ECCV	2020	Github	Sketch Refinement with Dilations, Pix2pixHD	AC, I	-	15
132	DeepFaceDrawing [228]	TOG	2020	Github	Component Embedding, Feature Mapping, Image Synthesis	AC, E	Km.	29
133	CA-GAN [229]	TC	2020	Github	Composition/Appearance Encoder, P-Net, Stacked GAN	Y, CU	Fl.	38
134	IDA-CycleGAN [230]	PR	2020	-	CycleGAN, Identity Loss, Recognition Model	Y, CU	-	20
135	IPAM-GAN [231]	SPL	2020	-	Identity-preserved Adversarial Model, U-Net	Y, CU	-	10
136	MyDT [232]	TIP	2020	Github	CNN [233] Features, Hand-crafted Features	Y, E	-	8
137	MSG-SARL [234]	TIFS	2021	-	Self-attention Residual Learning, Multi-scale Gradients	Y, CU	-	2
138	GAN Sketching [235]	ICCV	2021	Project	Weight Adjusting, Cross-domain Fine-tuning	CH, AL	-	3
139	DoodleFormer [236]	Arxiv	2021	-	Transformer, Part Locator and Part Sketcher Networks	CK	-	-

Ulyanov *et al.* [147] improved their previous TextureNet [138] by simply applying normalization to each individual image rather than a batch of images, which they called instance normalization. Moreover, they also demonstrated that the style transfer network with instance normalization can converge faster than that with batch normalization, while achieving visually better results. Later,

Huang and Belongie [149], following a similar idea, introduced adaptive instance normalization into the GAN model, aligning the content and style features. Li *et al.* [153] further used the first few layers of a pre-trained VGGNet [233] to extract the feature representation. However, they replaced the AdaIN layer with whitening and coloring transformations, enabling the universal style transfer. Along this line, LinearTransfer [165] integrates both a whitening and linear transformation into an auto-encoder based network for photo-realistic style transfer. Meanwhile, Park and Lee [166] introduced SANet, which takes advantage of an AdaIN layer, a style-attention module and a new identity loss to preserve the content structure while enriching the local and global style patterns. Similar to Image2StyleGAN [167], Richardson *et al.* [171] improved the classical StyleGAN with a novel encoder network that learns many style vectors that are fed into a pre-trained generator, forming an extended $\mathscr{W}+$ latent space.

Improved NST. Although the NST models achieve satisfactory results, they tend to over-simplify the transferring procedure, resulting in distorted and unwanted style patterns. Sheng *et al.* [158] proposed an Avatar-Net that enables multi-scale transfer for any styles. The key innovation is a "style decorator" that semantically aligns the content and style features. This module not only matches the feature distributions but also preserves the detailed style patterns. In [160], Zhang *et al.* first extracted the style and content features via a classical encoder. Then, they introduced a customized bi-linear EMD model to combine the style and content features. Finally, they used a standard decoder to map the combined feature representation to the output image with target style and content. Meanwhile, Jing *et al.* [168] revisited the normalization methods in NST and claimed that the current normalization-based NST is sub-optimal, due to its reliance on manual designs. To address this issue, they introduced dynamic instance normalization, combining the original instance normalization with a dynamic convolutional process, designed to achieve flexible and more efficient arbitrary style transfers. More recently, Lin *et al.* proposed a Laplacian pyramid network (LapStyle) for fast high-quality artistic style transfer, which transfers low-resolution style patterns via a drafting network and revises the high-resolution local details via a revision network.

Fast NST. Li *et al.* [145] proposed to use one deep model to synthesize multiple texture images, employing a selection-based sub-network to encode the style information into a one-hot vector in which each bit represents a specific style. At the same time, Chen *et al.* [146] decoupled style and content via separate network components, and proposed a flexible model built on a classical auto-encoder and a newly defined StyleBank layer. The auto-encoder is designed to learn the content information, while the StyleBank layer is responsible for learning the different styles. Besides, the StyleBank can easily be utilized for incremental learning, where a new unseen style can be added and trained into the module. Wang *et al.* [148] proposed a coarse-to-fine training procedure for fast multi-model style transfer, which learns artistic style at multiple cues, including color, coarse texture pattern and fine brushwork. Lu *et al.* [152] developed a new framework for fast semantic style transfer, which consists of a reconstruction module and a feature decoder. The reconstruction module is designed to approximate the optimization process in [134], which extracts the stylized features by minimizing the corresponding content and style losses. Then, the reconstructed features are fed into the decoder, which is based on a classical auto-encoder, to generate the stylized image. Shen *et al.* [157] designed a meta-learning framework, in which a meta network is used to approximate the stochastic gradient descent in [134], with a stylized image as input and corresponding image transformation networks as output. Moreover, this model has minimal parameters and can run in real-time. Finally, Chen *et al.* [161] proposed the first deep model for stereoscopic style transfer, which contains two components, *i.e.*, StyleNet and DispOccNet. Then, a standard decoder with a warp module and a fuse scheme is used to fuse all the domain information and extract the mid-level feature for generating a better stylized image.

Others. Li and Wand [137] enhanced their MRF-based NST method [135] via a Markovian generative adversarial network with adversarial learning, reducing the number of calculations. Furthermore, [143], [164] introduced additional constraints over the stylization results by controlling the spatial location, color information, spatial scale and stroke size. Moreover, they also extended the existing methods [136], [138] to synthesize high-resolution images via a coarse-to-fine model with downsampling-stylizing and upsampling-stylizing. Gupta *et al.* [150] studied the instability problem in existing NST models based on the technique of Gram matrix matching, and claimed that the instability is correlated to the trace of the Gram matrix of the style image. Therefore, they presented a recurrent convolutional network (RNN) for real-time video style transfer. More recently, Deng *et al.* [237] introduced a transformer-based style transfer framework, which aims to reduce content leak and achieve unbiased stylization.

2.4 Image-to-Image Translation

Image-to-image translation (I2I) [238] is a hot topic in computer vision and machine learning, where the goal is to transform the input image from a source domain to a different target domain, while retaining the intrinsic source content and transferring the extrinsic target style. Current I2I models are typically built on a generative adversarial network, and can be categorized into supervised, unsupervised, semi-supervised, and few-shot I2I.

Supervised I2I. Supervised I2I uses aligned image pairs as the source and target domain in order to learn a transformation model that can convert the source image into the target image. One representative I2I method is Pix2pix [23], which applies a conditional GAN (cGAN) [239] to the task. The main difference from the original cGAN is that the generator in Pix2pix is a U-Net [240]. However, Wang *et al.* [19] observed that the adversarial training in Pix2pix is unstable, preventing the model from generating high-resolution images. Therefore, they extended the original Pix2pix with a new adversarial loss, which can generate high-resolution images of size 2048×1024. Zhu *et al.* [176] proposed the BicycleGAN, which includes a conditional VAE and a conditional latent regressor GAN, to resolve the collapse problem, and achieved improved performance. Further, to reduce the loss of semantic information in the Pix2pixHD model [19], Park *et al.* [25] introduced a SPADE-based generator, which adds spatially-adaptive normalization into the generator of Pix2pixHD so as to enhance the semantic information throughout the network. However, SPADE adopts only one style code to adjust the overall style of an image, which is unsuitable for generating high-quality images and controlling the changing of the target image. To address this shortcoming, Zhu *et al.* [195] presented the SEAN model, which contains a new semantic region-adaptive normalization layer to enhance the style information.

Unsupervised I2I. Training with paired data is not practical, because it is time-consuming as well as labor-intensive. Therefore,

several unsupervised I2I models have been proposed to train two different generative networks under the constraint of a cycle-consistency loss. Examples include CycleGAN [20], DiscoGAN [174], and DualGAN [173]. Later, Liu *et al.* [24] proposed an unsupervised I2I model (UNIT), in which the same latent code in a shared latent feature space can represent image pairs in different domains. Meanwhile, Taigman *et al.* [175] presented a domain transfer network (DTN) to transfer a sample from one domain to another. They employed a compound loss function, which consists of a multi-class GAN loss, an f-constancy component, and a regularizing component that encourages the generator to map samples from the target domain to themselves. Li *et al.* [183] proposed the Stacked Cycle-Consistent Adversarial Network (SCAN), which uses a stacked network architecture with cycle-consistency to increase the image translation quality and generate higher-resolution images. SCAN is built on a coarse-to-fine framework, in which the coarse stage is used to sketch a result in low resolution, and the refinement stage is employed to improve the result through a novel adaptive fusion block. More recently, Zhang *et al.* [196] proposed a CoCosNet for exemplar-based image translation, which contains two sub-networks. The first embeds the inputs from different domains into a feature domain that depends on the semantic correspondence. Meanwhile, the second uses a series of denormalization blocks to progressively synthesize the target images. Zhou *et al.* further extended the CoCosNet with full-resolution semantic correspondence learning [202], with the main difference being the used of a regular and GRU-based propagation applied iteratively at each semantic level.

Despite their potential, cGAN-like models tend to collapse during training. The work in [190] therefore introduced an MS-GAN model that uses a mode-seeking regularization to handle this issue. The proposed regularization can be embedded into most existing cGAN frameworks, such as Pix2pix. Further, Zhang *et al.* [187] presented a HarmonicGAN for medical disease diagnosis, which takes the manifold into account and introduces a smoothing term on the affinity graph to enforce consistent mappings during translation. By rethinking the standard GAN model, Chen *et al.* [21] proposed a NICE-GAN with the key idea of coupling discriminators and encoders, *i.e.*, reusing the discriminator parameters for encoding the input. Zhao *et al.* [198] later proposed ACL-GAN, which utilizes a new adversarial consistency loss instead of a cyclic loss to emphasize the commonality between the source and target domains.

However, current models are typically deterministic at inference, making them inflexible for many practical scenarios. Therefore, Ramasinghe *et al.* [199] introduced a generalized generative model to address, which they called Conditional Generation by Modeling the Latent Space. During inference, this model dynamically observes the latent code by learning and updating an approximator, which it then applies to find the optimal solutions corresponding to multiple output patterns. More recently, Xu [200] observed that a well-trained styleGAN generator can be used as a learned loss function for extracting hierarchical features that have strong transferability to both generative and discriminative tasks. Also, focusing on the issue of latent features, Liu *et al.* [201] introduced a Divco framework for preventing model collapse and achieving diverse conditional image synthesis. Their model takes contrastive learning into account and employs a novel latent augmented contrastive loss.

Ma *et al.* [179] introduced an attention module into a GAN for instance-level I2I translation. The proposed deep attention-based encoder decomposes two different sets of samples into a highly-structured latent space, where the instance-level correspondence can be found by the joint attention mechanism, and the generator outputs the translated image according to the input of two different latent codes. Based on CycleGAN [20], Tang *et al.* [192] introduced an attention-guided GAN (AGGAN), which integrates the classical attention mechanism into a generator that can detect the most discriminative semantic objects and produce high-quality images. Besides, Tang *et al.* [191] further extended their AGGAN to solve the cross-view transfer problem, *i.e.,* when there is little or no overlap in different views. Further, they proposed a new SelectionGAN that utilizes a two-stage scheme with a multi-channel attention selection module. Kim *et al.* [22] later proposed a novel attention module with a new normalization function, which they integrated into a GAN model to flexibly supervise texture and shape variations.

MUNIT [184] and ElEGANT [185] simultaneously decouple the image representation into a domain-invariant content feature and a target style feature, and then recombine the content and target code to synthesize a new image via a generator. Note that MUNIT uses adaptive normalization [149] to achieve the recombination, while ELEGANT does so by exchanging certain parts of the latent codes. To improve the results, Cho *et al.* [188] took advantage of the whitening-and-coloring transformation and proposed the GDWCT model, which achieves competitive image quality. Wu *et al.* [189] later proposed the TransGaGa model to tackle the I2I translation task for complex objects. This model uses a VAE to decompose each domain into a geometric space and an appearance space, and further uses two transformers to transform each feature into the target-style image. Ma *et al.* [186] argued that the content of an image is shared across domains, while the style is specific to each domain. Based on this, they proposed the EGSC-IT model, which uses AdaIN and feature masks to transfer styles from the source image while maintaining semantic consistency. Sendik *et al.* [193] proposed CrossNet to relax the consistency constraints in CycleGAN-like models, which often contain information irrelevant to the I2I task. During training, CrossNet uses three new cross-consistency regularizations to constrain the learned image translation operators. To improve the content representation ability, Chang *et al.* [26] proposed DSMAP to leverage the relationship between content and style. Specifically, the model maps content features from a shared domain-invariant feature space into two separate domain-specific ones. Further, DRIT++ [27] uses two image generators, two content encoders, a content discriminator, two attribute encoders, and two domain discriminators to embed an image into a domain-invariant content space and a domain-specific attribute space.

However, most unsupervised I2I methods struggle to handle more than two domains, since the generator is usually dependent on each corresponding domain pair. Therefore, Choi *et al.* [180] introduced the StarGAN model, which uses just one generator to perform the I2I task for multiple domains. Specifically, they designed a special discriminator with an auxiliary classifier, which not only discriminates whether the input is true or false but also distinguishes which domain the input belongs to. Besides, they also modified the conditional GAN [23], where the input is the image together with the domain label. Meanwhile, [181] presents the ModularGAN, including a generator, an encoder, a discriminator, a reconstructor and a transformer, to map an image into multiple domains. The main difference between ModularGAN and StarGAN is that ModularGAN contains multiple special

transformers that transform the input to a better representation according to the attribute conditions. However, both StarGAN and ModularGAN are restricted to discrete conditional distributions. To address this, GANimation [182] was proposed to generate facial animation movements under the control of the activation magnitude of each action unit (AU). More recently, Choi *et al.* [131] further improved their StarGAN, introducing a new style encoder and a mapping function. This new version of StarGAN can synthesize more diverse images without handcrafted attributes.

Others. Semi-supervised learning, in which a small number of labeled samples and abundant unlabeled data are used to train the desired model, has been extensively studied. For semi-supervised I2I, Gan *et al.* [178] introduced a semi-supervised method for cross-domain joint distribution matching, called the Triangle Generative Adversarial Network. It consists of four neural networks, *i.e.*, two generators and two discriminators, which learn the bidirectional mappings between different domains with a few paired samples. Usually, supervised, unsupervised, and semi-supervised learning require significant data for training. In contrast, humans can learn from limited exemplars and achieve remarkable results. Benaim and Worf [177] were the first to take the few-shot scenario into consideration, proposing a one-shot unsupervised domain mapping method, called DistanceGAN. Its key innovation is to learn a unidirectional mapping function that maintains the distance between a pair of samples. Dong *et al.* [172] later introduced a zero-shot semantic image synthesis framework, which synthesizes a new image under the guidance of a natural language description. In [113], Liu *et al.* explored how to translate source images to analogous images with target conditions, without model having seen the target class during training. Therefore, they proposed a few-shot unsupervised I2I translation model, called FUNIT, which is built on a few-shot image translator and a multitask adversarial discriminator. Although FUNIT somewhat alleviates the reliance on domain annotations, they are still needed during training. To address this issue, Anokhin *et al.* [194] proposed the HiDT model, which combines an AdaIN-based U-Net and a new upsampling scheme that allows image translation to be applied at high resolution. Besides, Jiang *et al.* [197] proposed two-stream I2I translation (TSIT) to learn both semantic structural features and stylistic features, and then fuse the feature maps of the content and style in a coarse-to-fine manner. Recently, Meng *et al.* [203] introduce a novel SDEdit algorithm, which hijacks the reverse stochastic process of stochastic differential equations based generative model [241]. The SDEidt transforms a stroke painting or an image with strokes to the expected image, while preserving the overall structure.

2.5 Deep Photo-Sketch Synthesis

Deep photo-sketch synthesis is a recent branch of the FSS task, in which deep learning is used to improve the performance and quality. The related works can be divided into three categories. The first aims to translate a sketch into an RGB image, the second tries to convert a given RGB image into a sketch image, and the last mainly focuses on facial synthesis.

Sketch-to-Image. Xian *et al.* [208] proposed the TextureGAN model to synthesize an image under the supervision of a sketch, color, and texture. TextureGAN consists of a ground-truth pre-training module and an external texture fine-tuning part. Then, Lu [210] *et al.* introduced a two-stage contextual GAN to achieve sketch-to-image generation. This framework trains a classical GAN model [242] with a newly defined contextual loss, which represents the joint distribution and captures the inherent relation between a sketch and its corresponding image. Inspired by image in-painting [243], You *et al.* [218] proposed the PI-REC model, which contains three phases: an imitation phase, generating phase, and refinement phase. PI-REC is progressively trained using only one generator and one discriminator. The ISF introduced in [224] is a gating-based approach, which allows a single generator to be used to generate distinct classes without feature mixing. Recently, Gao *et al.* [226] proposed EdgeGAN for object-level image synthesis given a freehand scene sketches. This framework contains two sequential modules: foreground generation and background generation. Yang *et al.* [227] presented a deep plastic surgery model to simulate the coarse-to-fine painting process of human artists. Chen *et al.* [228] proposed a local-to-global framework to allow any user to produce high-quality face images. Their model consists of three modules, including a component embedding module, a feature mapping module, and an image synthesis module.

Image-to-Sketch. Song *et al.* [209] proposed the first deep stroke-level photo-to-sketch synthesis method, which is a hybrid model with a shortcut cycle consistency constrained by a VAE-style reconstruction loss. As the default setting of I2I and NST, both can synthesize artistic portrait drawing (APD) images. However, they do not meet practical requirements because APD images usually have a highly abstract style and contain special graphic elements. Therefore, Yi *et al.* [2] proposed an APDrawing to transform an input face image into its corresponding APD image, in which a hierarchical GAN model is built by combining both a global and a local network. Then, they further proposed an APDrawing++ [9], in which they used an auto-encoder to refine the subtle facial features and presented a novel line continuity loss to enhance line continuity of APDrawing. However, both of these APDrawing methods require pair-wise data for training. To handle this problem, Yi *et al.* thus later proposed an asymmetric cycle-structure GAN [36], which contains a relaxed forward cycle consistency loss (*a.k.a.* truncation loss) to prevent the reconstructed photo from being noisy, and a strict cycle consistency loss to enhance the performance. This method also uses multiple local discriminators to ensure the quality of the facial portrait drawings. Different to portrait drawing, Wang *et al.* [225] observed the behavior and properties of cartoon paintings and proposed three different representations considering surface, texture, and shape information, respectively. In addition, they also released the new SketchyCOCO dataset to better train and evaluate the performance of their model. Based on Pix2pix, Li *et al.* [223] designed a two-branch network (called im2Pencil) to implement photo-to-pencil translation, which can simulate sketch outlines and shadows. Bhunia *et al.* [236] introduced a new transformer architecture to generate various yet realistic creative sketches, consisting of a part locator network and a part sketcher network. The part locator networks aim to capture the coarse structure by observing the relationship between local visual patterns, which aims to generate diverse coarse structures. The part sketcher network follows the standard GAN, which aims to synthesize high-quality sketch images.

Photo-Sketch Synthesis. Zhang *et al.* [204] were the first to use a fully convolutional neural network (FCNN) to build a deep photo-to-sketch synthesis model. Then, the works [85], [207], [212] integrated deep features into probabilistic graph model learning, achieving better performance than traditional models [1],

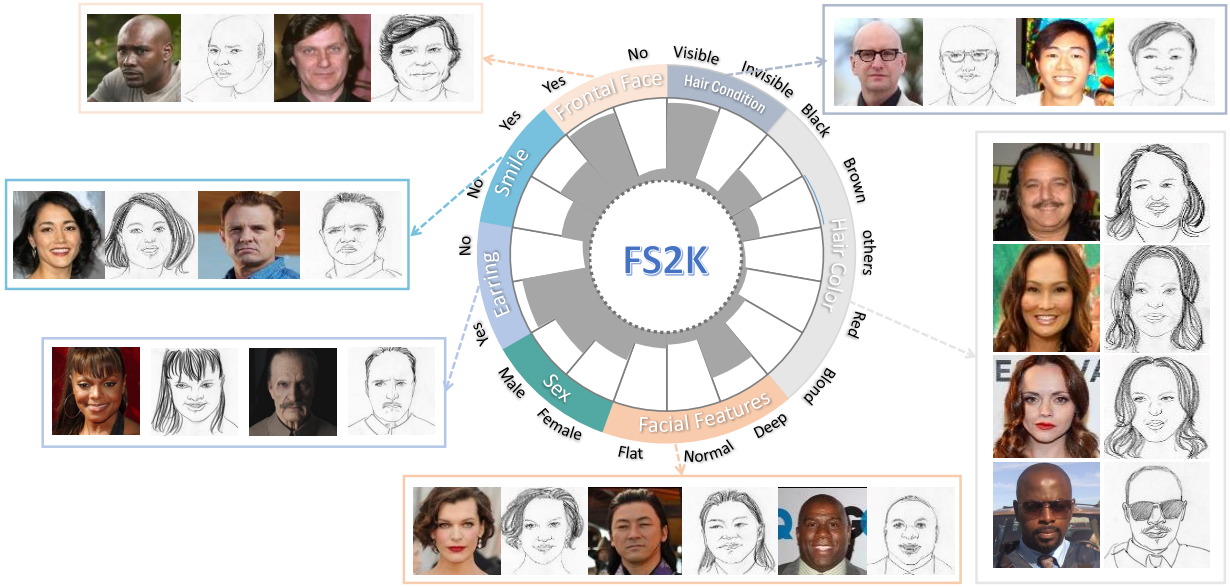


Fig. 5. Statistics and examples from the FS2K dataset. Please refer to Sec. 3 for details.

[49]. To make the network more flexible, Zhang *et al.* [211] took the key idea of CycleGAN and proposed a novel pGAN, which uses a special parametric Sigmoid activation function to reduce the effects of photo priors and illumination variations. To improve the quality of the generated photo/sketch, Wang *et al.* [213] introduced a synthesis method using multi-adversarial networks (PS^{2}MAN). Their model uses two U-Nets to generate high-quality images from low to high resolution. To achieve the same goal, Zhang *et al.* [215] further proposed a facial sketch synthesis by multi-domain adversarial learning (MDAL), which overcomes the defects of blurs and deformations. The basic idea behind MDAL is the concept of "interpretation through synthesis", which is built upon two diverse generators. Kazemi *et al.* [216], [217] proposed an improved version of CycleGAN, which focuses on the facial attributes during the portrait synthesis process. Zhang *et al.* [219], [221] introduced two methods by combining an auto-encoder and traditional subspace learning, which is more effective than the traditional FSS methods. Besides, Zhu *et al.* [220] proposed a collaborative framework that exploits the interaction information of two opposite generators by introducing a collaborative loss. However, due to the lack of large-scale training data, it is difficult to train a good model. Therefore, Zhu *et al.* [222] proposed to use classical knowledge distillation to learn two well-defined student mapping networks via two strong teacher networks. More recently, the works in [230], [231] introduced identity-aware models, which use a new perceptual loss to train a better image generative model, and thus consider the downstream task, *e.g.*, face recognition, as the final goal. Yu *et al.* [229] proposed a new composition-assisted generative adversarial network, which helps synthesize realistic facial sketches/photos by using facial composition information. By leveraging the relationships between features, [234] implemented a multi-scale self-attention residual learning framework for face photo-sketch conversions. Finally, the method proposed in [232] does not need any images from the source domain for training, enabling it to leverage both deep features (extracted from convolutional neural network) and handcrafted features flexibly.

3 Proposed FS2K Dataset

In this section, we introduce the proposed FS2K. Some example images are shown in Fig. 2. We describe the details of FS2K in terms of two key aspects, namely dataset collection and data annotation. Overall, FS2K includes 2,104 photo-sketch pairs, which are split into 1,058 for training and 1,046 for testing. The complete dataset is available at https://github.com/DengPingFan/FS2K.

3.1 Data Collection

To establish a long-lasting benchmark, the data should be carefully selected to cover diverse scenes from different views, such as lighting conditions, skin colors, sketch styles, and image backgrounds. To this end, we introduce FS2K, a new high-quality dataset[3] for the FSS task.

Our FS2K includes 2,104 photos from real scenes, the Internet, and other datasets. The majority, however, come from CASIA-WebFace [244], which is a large-scale (*i.e.*, 500K images) labeled dataset of faces in the wild. CASIA-WebFace was collected from the IMDb[4] website and contains well-organized information, such as name, gender, and birthday. Thanks to the rich and clean open-source data from CASIA-WebFace, it could be used to build our high-quality and representative benchmark. We manually selected 1,529 images to cover a large span of major challenges faced in realistic scenes, such as varying background, hair style (*e.g.*, long, short), accessories (*e.g.*, glasses, earring), and skin information (*e.g.*, patch image on a given face). Because the photos selected in CASIA-WebFace are taken from a single angle, multi-angle face images for the same person are missing. To this end, we invited eight actors to take 98 photos under different settings (*e.g.*, lighting conditions, face angles). In addition, to further increase the diversity, we also collected some children photos and some faces with smaller face-to-image ratios. The remaining 477 face

3. This dataset is for academic communication only and not for commercial purposes.
4. http://www.imdb.com

(a) Schematic (b) Specification (c) Virtual example (d) Our example

Fig. 6. Use of the copy table and an example. Zoom-in for the best view. See Sec. 3.2 for more details.

TABLE 4. Number of images for each attribute in the training and test datasets.

FS2K (Ours)	w/ H	w/o H	H(b)	H(bl)	H(r)	H(g)	M	F	w/ E	w/o E	w/ S	w/o S	w/ F	w/o F	S1	S2	S3
Train	1010	48	288	423	60	239	574	484	209	849	645	413	917	141	357	351	350
Test	994	52	290	418	44	242	632	414	187	859	670	376	872	174	619	381	46

photos come from other free stock photos websites, including Unsplash[5], Pexels[6], Pngimg[7], and Google.

3.2 Data Annotation

There are four types of annotations in our FS2K, including sketch drawing, sketch style, color, contour feature annotations.

3.2.1 Sketch Drawing

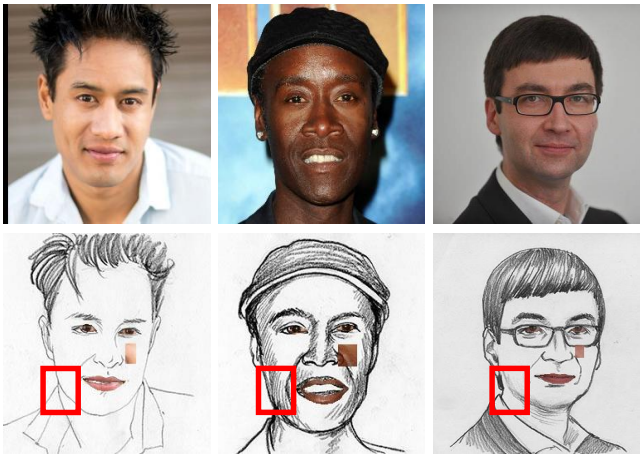
Participants. Three senior artists (including two male and one female) from the Sichuan Fine Arts Institute were hired to take part in the study.[8] All three participants had normal or corrected to normal vision. None of the participants suffered color-blindness or color-weakness. The participants ranged in age from 20 to 23 years, with an average of five years of professional experience in sketch drawing.

Apparatus. The three artists drew all sketch images with the assistance of a Copy Table LED Board. [9] Fig. 6 shows the copy table we used and an example (Fig. 6 (d)) of a face sketch drawn by our artists. The touch switch region in our device supports three levels of adjustable brightness, so the artists could use the button to change the brightness they desired. This helped them locate the contours of facial features according to the photo information from the bottom of the LED board. Moreover, this equipment also helped to ensure content similarity and face alignment between sketches and corresponding photos. At the same time, the drawings retain the artist's sketch style.

3.2.2 Sketch Style Annotation

Our FS2K contains three different styles, which enrich the diversity of sketches, as shown in Fig. 7. This enables different artists' skill to be captured, while the same time making FS2K more challenging than previous FSS datasets.

We created a balanced dataset to facilitate the comparison of different methods, *i.e.*, the number of the images with the three different styles are equally distributed. Specifically, in the training set, the number of samples with style1, style2, and style3 are 357, 351 and 350, respectively. In the test set, they are 619, 381, and 46, respectively.



(a) Style 1 (b) Style 2 (c) Style 3

Fig. 7. Three sketch styles in our FS2K. As shown in the cheek region, the styles include simple lines (style 1), long strokes (style 2) and repeated wispy details (style 3).

3.2.3 Facial Feature Annotation

Sketches are rapidly executed freehand drawings, which have less attribute information than the original images, *e.g.*, facial texture, facial expressions, facial posture, *etc*. Therefore, it is difficult to restore real images (*i.e.*, S2I task) based on a single sketch image. Meanwhile, in real-world applications, we can use auxiliary facial information (such as gender, accessories, hair style) to better narrow down a suspect in a database. Following [245], we added some additional facial feature annotations, including *gender, smile, face pose, hair condition, hair color, earring, and skin texture*. We hired two data annotators to label all photos and performed cross-checking to ensure the accuracy of the final annotations. An overall summary of the labels can be found in Table 4, while the details of each are described below.

Gender. Gender is a high-level human attribute commonly used in traditional face databases such as CelebA [28] and LFW [246]. It has been extensively studied in face detection and recognition [247]–[249]. Therefore, we carefully labeled all photos in FS2K with gender attributes. Specifically, there are 574 male

5. http://www.unsplash.com
6. http://www.pexels.com/
7. http://pngimg.com/
8. The Sichuan Fine Arts Institute is one of the four most prominent art academies in China. The three senior artists are all from the Design Academy.
9. Fig. 6 (a) presents the copy table, which has an LCD backlight. It requires a high voltage input of $100 \sim 240$V and 0.6A working current. Its size is A4 (*i.e.*, $300 \times 200 \times 3.5$mm) in Fig. 6 (b), and the luminous intensity is 300∼350LM. Therefore, it has become the most popular copy table product, after the aluminum alloy copy table, for animators (see Fig. 6 (c)).

photos and 484 female photos in the training set, and 632 male photos and 414 female photos in the test set.

Smile. Smiling is a basic human action that represents a positive emotional state. As such, many studies have focused on smile detection [250], [251], or used smile as an attribute for recognition [252]. Therefore, we also consider smile as a key attribute in our dataset. Specifically, the training set contains 645 smiling people and 413 with no obvious expression, while the test set contains 670 smiling people and 376 with no expression. We make sure that the proportion of smiling people in the training and test sets is as close as possible.

Face Pose. The facial attributes may cover only a small part of the image, but the photo is usually dominated by the effects of pose [253]. Moreover, pose will affect the performance of face recognition [254], tracking [255], and synthesis [256]. Therefore, the facial pose is useful auxiliary information. We define a portrait with the head rotated within 30 degrees as a frontal face pose. According to this definition, the training set has 917 frontal photos, while the test set has 872. The remaining have side face poses.

Hair Status and Color. Hair is a saliency feature of the head that may change in different situations. Even if there is sufficient information in the internal features of the face for recognition, manipulating the hair can have a negative effect on the performance [257], [258]. Moreover, facial synthesis and retrieval systems often use hair as an important cue [259], [260] to improve the quality of generated images. For FSS, although the sketches contain the contour of the hair, the corresponding color information and hair status (with or without hair) are missing. Therefore, in FS2K, we provide annotations of the hair status, which includes four general colors (*i.e.*, black, brown, red, and blond) and another status (*i.e.*, bald or wearing a hat), as shown in Fig. 5. In other words, for faces with hair, we mark the color information directly, while cases of thinning hair or wearing hat are marked as separate attributes. The statistical results of this annotation can be found in Table 4.

Earrings. The simplified characteristics of sketch drawings lead to unclear earring contours. Meanwhile, as shown in Fig. 5, earrings in real photos are clearly visible. Therefore, in FS2K, we provide annotations for whether or not earrings are present, which can help the model training. Specifically, the training set has 209 people with earrings, and the test set has 187.

Skin Texture. Skin texture provides a large amount of detailed local information and is used as an important feature for face recognition [261], [262]. However, this important information is completely lost in sketch images. Therefore, we clip a small patch from the real photo and use it as the skin texture, as shown in Fig. 7. To provide more information for future research, we also include the average RGB value for the corresponding lip and eyeball region.

4 Proposed FSGAN Baseline

4.1 Problem Definition

Facial synthesis (FS) aims to generate target representations of human faces based on the given inputs. This process can be formulated as $X_o = F(X_i)$, where X_i and X_o denote the input and output (*e.g.*, RGB images and sketches) of facial representations, and F indicates the synthesis function. In this paper, we propose a novel baseline model, FSGAN, for both the I2S $X_{ske} = F(X_{img})$ and S2I $X_{img} = F(X_{ske})$ task, inspired by pix2pixHD [19]. Instead of focusing on direct image-level facial synthesis, we propose a two-stage "bottom-up" facial synthesis architecture, as shown in Fig. 8. Hence, our FSGAN consists of two cascaded stages built upon multiple generative models (*i.e.*, GANs).

The first stage is comprised of five parallel GANs, which are designed to synthesize the local facial components separately. Given an input, four facial regions (*e.g.*, left eye, right eye, nose, and mouth), as well as the rest of the input, are cropped and fed into their corresponding GANs in the first stage for synthesizing key facial features. These synthesized facial component patches are then stitched together to obtain the intact facial representation. Since the local facial patches are synthesized independently, the connecting region of the stitching, as well as their appearances, are inconsistent with each other. Therefore, the second stage is introduced to further refine the results by taking the global structure and texture into consideration. In this stage, the style vectors of the facial sketches are utilized to assist the synthesis.

4.2 Facial Components Synthesis

Almost all human faces have the same global structure. The differences lie in the details of the local facial components, such as eyes, eyebrows, nose, and mouth. To capture more details of different facial components, the first stage of our model synthesizes them separately. Specifically, given a facial input, the four key patterns, including the left eye, right eye, nose and mouth, are first detected by MTCNN [263]. The input X_i is then divided into five parts, $X_{parts} = \{X_{leye}, X_{reye}, X_{nose}, X_{mouth}, X_{rest}\}$, based on the detection results. These include the left eye, right eye, nose, mouth and remaining components. For these parts, five parallel GANs are utilized to synthesize their corresponding patches. Therefore, the problem can be formulated as $G_{parts} = \{G_{leye}, G_{reye}, G_{nose}, G_{mouth}, G_{rest}\}$ and $D_{parts} = \{D_{leye}, D_{reye}, D_{nose}, D_{mouth}, D_{rest}\}$, where G and D indicate the generator and discriminator, respectively.

First, the four GANs for synthesizing the left eye, right eye, nose and mouth have the same architecture. Each GAN consists of a generator and a discriminator. The generator is designed as a encoder-decoder, which consists of an encoder, a bottom connection and a decoder. The encoder is composed of three convolutional blocks, each of which is a combination of a convolutional layer (with a kernel size of 3 and stride of 2), a batch normalization layer and a ReLU activation layer. Meanwhile, the second bottom connection consists of nine bottleneck residual blocks that are similar to [264]. Finally, the decoder is built upon three deconvolutional blocks, which consist of a deconvolutional layer, a batch normalization layer, and a ReLU activation layer. Note that the GAN, which is used for synthesizing X_{rest}, is similar to the previous described ones. However, the encoder contains four convolutional blocks and the decoder contains four deconvolutional blocks, in order to achieve larger receptive fields.

The discriminators of the above five GANs are the same. Each consists of three cascaded convolutional layers (with a kernel size of 3 and stride of 2) followed by global average pooling. Then, a 1×1 convolutional layer and a sigmoid function are used to predict the probability of the generated results being real or fake.

Based on the above design, the first stage of FSGAN is able to restore details of the facial components in both the I2S and S2I tasks. At the end of this stage, the synthesized patches are stitched together to restore the intact facial synthesis result X_{intact}. Since the patches are synthesized by different generators, their overall appearances are inconsistent, which becomes even more obvious in the stitched result. To address this issue, the stitched result is then fed to the next stage to adjust and refine the global structure and appearance.

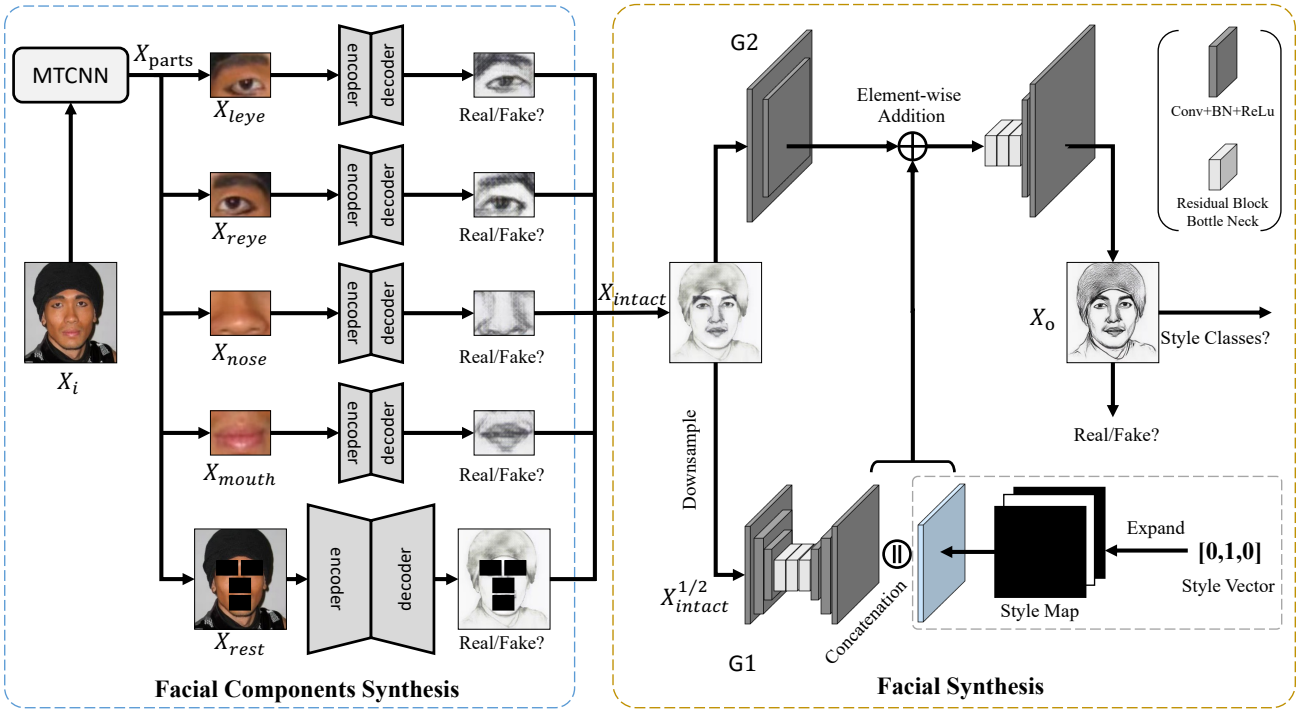


Fig. 8. Pipeline of our FSGAN baseline for the I2S task. It consists of two stages: 1) facial features synthesis and 2) facial synthesis. Please refer to Sec. 4.2 and Sec. 4.3 for more details.

4.3 Facial Synthesis

To address the inconsistency issue of the output from the first stage, we introduce the second stage, which is designed as another GAN model inspired by Pix2pixHD [19], for local detail refinement and global structure adjustment.

In this stage, we use the multi-scale discriminators D_{fs} and the coarse-to-fine generator G_{fs} following Pix2pixHD [19]. Specifically, the generator G_{fs} consists of two sub-networks $G1$ and $G2$, both of which follow on encoder-decoder architecture, as shown on the right of Fig. 8. We first sample the output of the first stage using a downsampling operation with a sampling rate of 50%. This newly sampled image $X_{intact}^{1/2}$ $((height/2, width/2))$ is then fed into the first sub-network $G1$, which is designed to capture global features. The other sub-network $G2$ is employed to capture the local details, which takes the output of the first stage as input. We use both concatenation and element-wise addition operations to fuse the style, local, and global information. Specifically, the concatenation operation is used to combine the style feature map and the output of $G1$ and generate a new fused feature map. Then, the element-wise addition is utilized to combine this new feature map with the latent feature of the encoder part of $G2$. Finally, we use the decoder part of $G2$ to generate the final output X_o. It is worth noting that the style vector can control the style of the generated sketches, which helps improve their quality and diversity. Besides, the style of the real photo is often fixed, and independent from the artists' style. Therefore, we introduce the style information in the I2S task, but exclude that in the S2I task.

4.4 Loss Function

We use a combination of several loss functions to train our model. We denote X and Y as the input and its corresponding ground truth, respectively. For simplicity, we define $G(X)$ as the generated output of the given input X, and $D_k(X,Y)$ as the corresponding predicted probabilities of the k-th discriminator. Then, we denote the i-th layer feature extractor of discriminator D_k as D_k^i, where k is the index of the discriminator.

Adversarial Loss. We use the adversarial loss [242] to make the generated image more visually appealing. The adversarial loss we use is defined as:

$$L_{\text{adv}}(G,D) = \mathbb{E}_{X,Y}[\log D(X,Y)] + \mathbb{E}_X[1 - \log D(X,G(X))]. \quad (1)$$

Feature Matching Loss. Similar to [19], we use the feature matching loss to improve the adversarial loss based on the k-th discriminator. The feature matching loss is defined as:

$$L_{\text{fm}}(G,D_k) = \mathbb{E}_{X,Y} \sum_{i=0}^{T} \frac{1}{N_i} \left[\left\| D_k^i(X,Y) - D_k^i(X,G(X)) \right\|_1 \right], \quad (2)$$

where T denotes the total number of layers in each discriminator and N_i is the number of feature maps in the i-th layer. This loss is used to match the intermediate feature maps of the real and synthesized image, making the generator produce multi-scale statistical information. Besides, it not only stabilizes the training process but also restores highly realistic outputs.

Perceptual Loss. To maintain perceptual and semantic consistency, we use a perceptual loss [136] to measure the difference between the original image and the corresponding synthesized image. We extract the perceptual features from the i-th layer activations of a pre-trained VGGNet [233], which is denoted as $\phi_i(\cdot)$. The perceptual loss is defined as follows:

$$L_{\text{per}}\big(G(X),Y\big) = \mathbb{E}_{G(X),Y} \sum_{i=0}^{t} \left\| \phi_i(X) - \phi_i(G(X)) \right\|_1. \quad (3)$$

Pixel-Wise Loss. The L_1 distance between a generated image $G(X)$ and ground-truth Y is regarded as the pixel-wise loss, which is defined as:

$$L_1\big(G(X),Y\big) = \frac{1}{h\times w}\sum_{(i,j)=(0,0)}^{(h,w)} \big\|Y(i,j) - G(X(i,j))\big\|_1, \quad (4)$$

where (i,j) and (h,w) are the pixel coordinates and the (height, width) of the output, respectively.

Style Classification Loss. Similar to [180], [181], we define an auxiliary classifier to predict the sketch style of the generated image. For any generated image $G(X)$, the style classification loss is defined as:

$$L_{\text{sty}}(G,S,c) = \mathbb{E}_{X,c}\big[l_{\text{ce}}\big(S(G(X)),c\big)\big], \quad (5)$$

where $l_{\text{ce}}(\cdot,\cdot)$ is the cross-entropy loss, $S(\cdot)$ is a CNN that outputs the probability over different styles, and c is the label of a given artist's style. Note that we only use the style classification loss in the second stage for the I2S task.

Overall Loss. Finally, the overall loss function for the multi-scale discriminators is:

$$L_{D\sim(D_{parts},D_{fs})} = \sum_i^K -L_{\text{adv}} + \lambda_{\text{fm}}L_{\text{fm}}, \quad (6)$$

and the overall loss function for generator is:

$$L_{G\sim(G_{parts},G_{fs})} = L_{\text{adv}} + \lambda_{\text{fm}}L_{\text{fm}} + \lambda_1 L_1 + \lambda_{\text{per}}L_{\text{per}} + \lambda_{\text{sty}}L_{\text{sty}} \quad (7)$$

where λ_{fm}, λ_1, λ_{per}, and λ_{sty} are hyperparameters that control the importance of the feature matching loss, perceptual loss, pixel-wise loss, and style classification loss, respectively.

4.5 Implementation Details

We use PyTorch [265] to implement our FSGAN. The experiments are conducted on an NVIDIA V100S.

For the I2S task, we set λ_{fm} = 25.0, λ_1 = 25.0, λ_{per} = 12.5 to train the model in the facial components synthesis stage, and set λ_{fm} = 100.0, λ_1 = 100.0, λ_{per} = 50.0, and λ_{sty} = 100.0 for facial synthesis. The Adam optimizer [266] is used for training the whole network. The initial learning rates for the generator and discriminator are $2e-4$ and $1e-5$, respectively. The other hyperparameters of the optimizer are set to the default values as recommended in PyTorch. We set the number of epochs to 50. All generators and discriminators are trained iteratively.

For the S2I task, we set λ_{fm} = 50.0, λ_1 = 50.0, and λ_{per} = 0.2 to train the neural network for facial components synthesis stage, and set λ_{fm} = 100.0, λ_1 = 100.0 and λ_{per} = 0.2 for facial synthesis. We again use the Adam optimizer, with initial learning rates of $2e-4$ for both the generators and discriminators. The training strategy is almost the same as that for the I2S task. However, we set the number of epochs to 400,[10] freezing the weights of the facial components synthesis module after 250 epochs, and further training the facial synthesis module for the remaining epochs.

5 Benchmark

In this section, we provide comprehensive comparisons and analyses of the existing models on our newly proposed dataset, in terms of both the I2S and S2I tasks.

10. Because the S2I task needs to restore more detailed information of the RGB images, more training epochs are required.

5.1 Experimental Settings

5.1.1 Evaluation Metrics

For the I2S task, the most popular facial sketch metric is the structural similarity index metric (SSIM) [10], [43]. However, it ignores the perceptual similarity between a prediction and the ground truth. Therefore, we further adopt the recently proposed structure co-occurrence texture (SCOOT) metric [29], which provides a unified evaluation for both structure and texture. For the S2I task, we still adopt the widely used SSIM metric to evaluate the synthesized faces. Our evaluation toolbox is available at https://github.com/DengPingFan/FS2KToolbox.

5.1.2 Compared Models

To evaluate the performance on the I2S task, we present the empirical results of 19 representative approaches, including DualGAN [173], DRIT++ [27], WCT [153], CycleGAN [20], AdaIN [149], FPST [140], ACL-GAN [198], CartoonGAN [156], Pix2pix [23], DSMAP [26], UGATIT [22], NICE-GAN [21], NST [133], [134], MDAL [215], TSIT [197], UNIT [24], APDrawing [2], Pix2pixHD [19], UPDG [36], and our newly proposed FSGAN baseline.

For the S2I task, we again select 19 cutting-edge models, *i.e.*, FNS [136], DualGAN [173], DRIT++ [27], WCT [153], CycleGAN [20], AdaIN [149], FPST [140], ACL-GAN [198], Pix2pix [23], DSMAP [26], UGATIT [22], NICE-GAN [21], NST [133], [134], TSIT [197], UNIT [24], Pix2pixHD [19], pSp [171], SPADE [25], DeepPS [227], and our newly proposed model, to evaluate the performance on our FS2K.

5.1.3 Training/Testing Protocols

All compared methods are selected based on three criteria: a) widely regarded technology, b) open-source code, c) state-of-the-art (SOTA) performance. The models are trained and tested on our FS2K with the image sizes specified in their papers. If size setting is not provided in their paper, 512×512 is utilized as default.

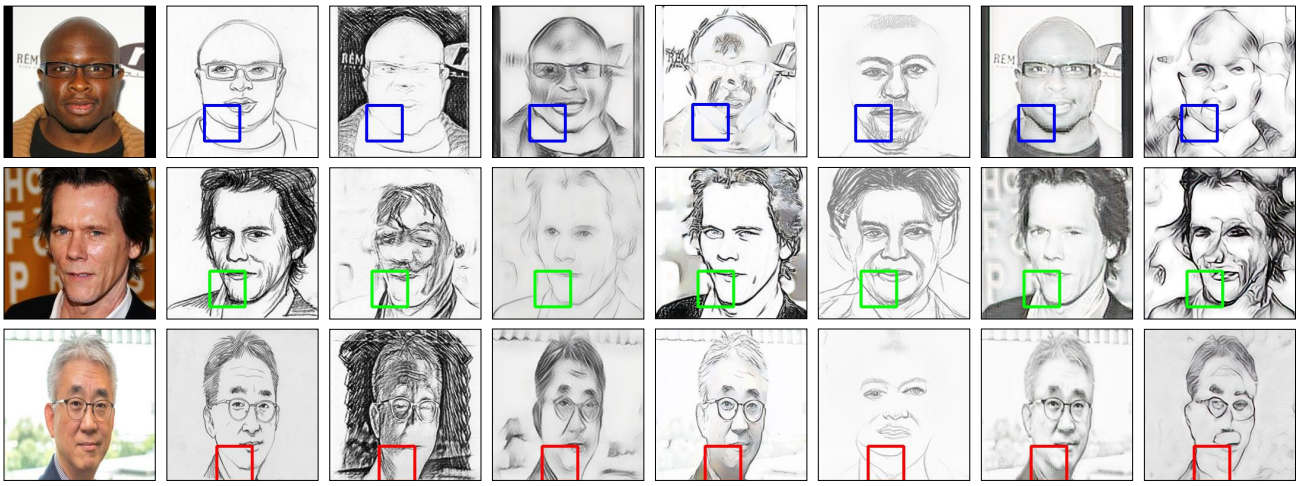
5.2 Overall Results and Analysis

5.2.1 I2S Task

We first provide a performance summary of the I2S task in terms of both SCOOT and SSIM scores. Quantitative results

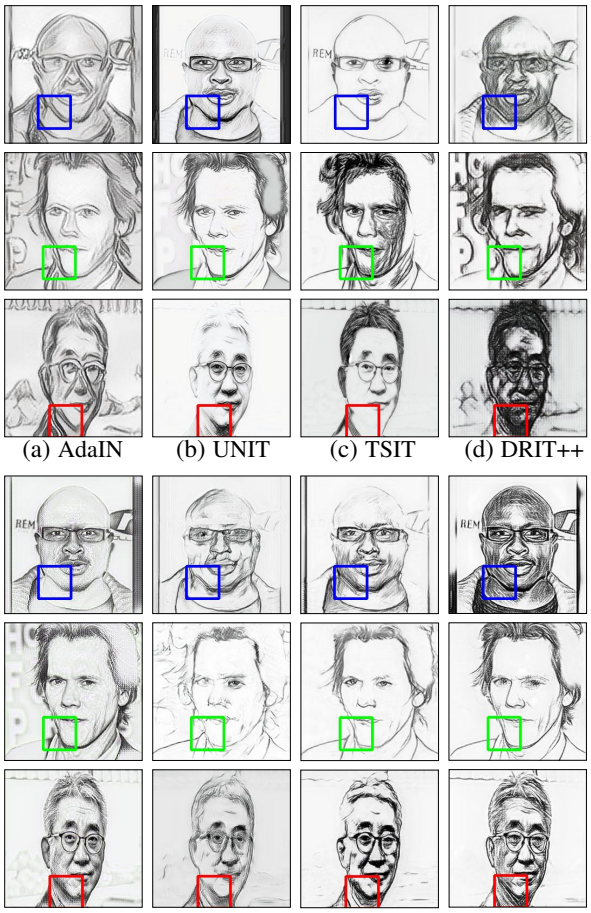
TABLE 5. Quantitative results of popular models on the I2S task. "↑" means the higher, the better. Publ.: Publication information.

#	Model	Publ.	SCOOT↑	SSIM↑
1	DualGAN [173]	Yi *et al.* ICCV	0.261	0.324
2	FPST [140]	Chen *et al.* NeurIPSW	0.271	0.460
3	NST [133], [134]	Gatys *et al.* CVPR	0.273	0.326
4	Pix2pix [23]	Isola *et al.* CVPR	0.275	0.438
5	ACL-GAN [198]	Zhao *et al.* ECCV	0.278	0.404
6	WCT [153]	Li *et al.* NeurIPS	0.282	0.369
7	AdaIN [149]	Huang *et al.* ICCV	0.303	0.365
8	UNIT [24]	Liu *et al.* NeurIPS	0.304	0.504
9	TSIT [197]	Jiang *et al.* ECCV	0.307	0.441
10	DRIT++ [27]	Lee *et al.* IJCV	0.308	0.492
11	CartoonGAN [156]	Chen *et al.* CVPR	0.319	0.400
12	UGATIT [22]	Kim *et al.* ICLR	0.323	0.457
13	NICE-GAN [21]	Chen *et al.* CVPR	0.327	0.473
14	CycleGAN [20]	Zhu *et al.* ICCV	0.348	0.435
15	MDAL [215]	Zhang *et al.* TNNLS	0.355	0.466
16	UPDG [36]	Yi *et al.* CVPR	0.364	0.471
17	Pix2pixHD [19]	Wang *et al.* CVPR	0.374	0.492
18	APDrawing [2]	Yi *et al.* CVPR	0.375	0.464
19	DSMAP [26]	Chang *et al.* ECCV	0.378	0.493
20	**FSGAN (Ours)**		**0.405**	**0.510**



(a) Input (b) GT (c) DualGAN (d) FPST (e) NST (f) Pix2pix (g) ACL-GAN (h) WCT

Fig. 9. From left to right: input face, ground truth (GT), DualGAN [173], FPST [140], NST [133], [134], Pix2pix [23], ACL-GAN [198], and WCT [153]. We mark the three styles with blue, green, and red boxes for each result. Zoom-in for details.



(a) AdaIN (b) UNIT (c) TSIT (d) DRIT++

(e) CartoonGAN (f) UGATIT (g) NICE-GAN (h) CycleGAN

Fig. 10. Comparison of AdaIN [149], UNIT [24], TSIT [197], DRIT++ [27], CartoonGAN [156], UGATIT [22], NICE-GAN [21], and CycleGAN [20]. Their inputs and GTs are shown in Fig. 9.

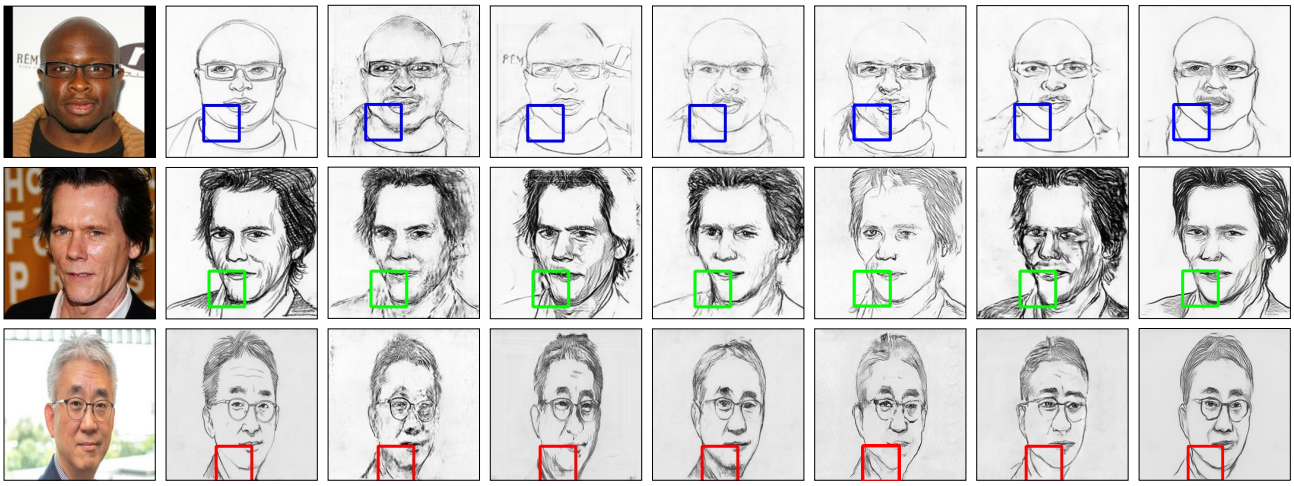
and qualitative comparisons are shown in Table 5 and Fig. 9-11, respectively. The experimental observations indicate that our FSGAN baseline achieves better results. For further analysis, we divide all compared methods into three categories based on their SCOOT score:

- score $<= 0.3$;
- $0.3 <$ score $<= 0.35$;
- $0.35 <$ score.

Analysis. Methods in the first group achieve a SCOOT below 0.3. These include DualGAN [173], FPST [140], NST [133], [134], Pix2pix [23], ACL-GAN [198], and WCT [153]. As shown in Fig. 9, DualGAN, NST, and WCT suffer from structural distortion, where many local facial details are lost. The images produced by the DualGAN are poor and it is difficult to detect facial components in them. This explains why it has lower SSIM and SCOOT scores. In addition, compared with other results, Pix2pix and FPST generate blur results. In terms of visual appeal, ACL-GAN seems to achieve satisfactory results, yielding a higher SSIM score. However, ACL-GAN reproduces the original facial structure almost exactly, lacking artistic style.

The second group includes AdaIN [149], UNIT [24], TSIT [197], DRIT++ [27], CartoonGAN [156], UGATIT [22], NICE-GAN [21], and CycleGAN [20], whose SCOOT scores range from 0.3 to 0.35. As shown in Fig. 10, the synthesized sketch images are better in terms of structure-preserving compared to the first group. However, except for AdaIN, all models are thrown off by the complex backgrounds (see the hair region in the second row). Besides, the results of CartoonGAN seem to alter the color of the input images, leading to lower SSIM scores.

MDAL [215], UPDG [36], Pix2pixHD [19], APDrawing [2], DSMAP [26] and the proposed FSGAN are categorized into the third group, which can generate sketches without distortion or losing too much of the global details. However, UPDG and APDrawing miss some details in the hair region, leading to poor visual effects. APDrawing introduces a lot of extra strokes, especially for the first sketch style. Besides, APDrawing usually results in a lack and distortion of the local structure, as can be seen in the fair region. Meanwhile, the sketches generated by UPDG have better style elements, but the model cannot handle complex backgrounds. Pix2pixHD generates relatively good sketches with global structure and clean background, but it does not generate the best facial components. For example, in Fig. 11 (e), the region around the eyes is unclear, and many details are lost.



(a) Input (b) GT (c) MDAL (d) UPDG (e) Pix2pixHD (f) APDrawing (g) DSMAP (h) Ours

Fig. 11. Comparison results with MDAL [215], UPDG [36], Pix2pixHD [19], APDrawing [2], and DSMAP [26]. Please refer to Fig. 9 for more descriptions.

TABLE 6. Quantitative results of popular models on the S2I task. "↑" means the higher, the better. Publ.: Publication information.

#	Model	Publ.	SSIM↑
1	DualGAN [173]	Yi *et al.* ICCV	0.241
2	WCT [153]	Li *et al.* NeurIPS	0.311
3	ACL-GAN [198]	Zhao *et al.* ECCV	0.314
4	TSIT [197]	Jiang *et al.* ECCV	0.316
5	UGATIT [22]	Kim *et al.* ICLR	0.317
6	NST [133], [134]	Gatys *et al.* CVPR	0.335
7	CycleGAN [20]	Zhu *et al.* ICCV	0.339
8	Pix2pix [23]	Isola *et al.* CVPR	0.346
9	SPADE [25]	Park *et al.* CVPR	0.361
10	UNIT [24]	Liu *et al.* NeurIPS	0.362
11	AdaIN [149]	Huang *et al.* ICCV	0.373
12	DRIT++ [27]	Lee *et al.* IJCV	0.381
13	FNS [136]	Johnson *et al.* ECCV	0.391
14	NICE-GAN [21]	Chen *et al.* CVPR	0.397
15	FPST [140]	Chen *et al.* NeurIPSW	0.400
16	pSp [171]	Richardson *et al.* CVPR	0.428
17	Pix2pixHD [19]	Wang *et al.* CVPR	0.433
18	DSMAP [26]	Chang *et al.* ECCV	0.471
19	DeepPS [227]	Yang *et al.* ECCV	0.487
20	**FSGAN (Ours)**		**0.503**

Take the third style, for instance, the eyeglasses are partially lost, while the eyeball is completely black. We further observe that DSMAP and MDAL tend to achieve better sketch images, but with distortions in local facial information. Finally, our proposed baseline can synthesize high-quality sketches that focus on the global structure and local details, while taking diverse styles into account. Moreover, as shown in the highlighted boxes (with green, blue and red), we find that the outputs of our proposed FSGAN are more similar to the ground-truth, compared to other SOTA models.

5.2.2 S2I Task

We report our experimental results in Table 6 and Fig. 12. We find that our FSGAN achieves the best results on our challenging FS2K compared to the existing SOTA models.

Analysis. From the results shown in Fig. 12, we observe that most compared methods are unable to successfully recover accurate images, revealing that the S2I task is more complicated than I2S. We argue that this is because the sketches are highly abstract, and the loss of valuable information makes it difficult for neural networks to restore the original image. We also observe that the high-resolution models, such as Pix2pixHD and ours, tend to output more visually appealing results.

The results presented in Fig. 12 show that FNS and FPST fail to transfer the sketches into colored images. SPADE and Pix2pix generate poor results with facial outlines (*e.g.*, Pix2pix) or black backgrounds (*e.g.*, SPADE). Five models (*i.e.*, NST, WCT, DeepPS, DSMAP, and UNIT) produce noise patches in salient regions, which corrupt the global facial structure. Meanwhile, AdaIN, ACL-GAN, DualGAN, and UGATIT perform better than the above-mentioned models, but result in unrealistic cartoon-style images. Only CycleGAN, NICE-GAN, TSIT, pSp, and Pix2pixHD overcome various challenges and achieve good results in terms of facial completeness. In particular, the eye regions from Pix2pixHD [19] and pSp [171] are better than other models. However, compared with the results of our model, the facial features of Pix2pixHD are relatively inferior, because they are learned by a pixel-wise rather than block-wise strategy. Although pSp [171] can generate high-quality results, its results lack diversity compared with ours. For example, pSp generates the similar facial expressions under two different sketch styles, while our model can synthesize diverse contents, as shown in Fig. 13.

5.3 Attribute-Based Results and Analysis

5.3.1 SCOOT Metric Results

To provide a deeper understanding of the models, we present an attribute-based performance evaluation in Table 9.

Analysis. Hair is one of the dominant features of the head. In Table 9, we find that most models achieve slightly better or comparable performance on images without hair than with, except for three models, such as AdaIN, CartoonGAN, and CycleGAN. Meanwhile, we find that red and black hair are the most challenging and easiest to detect/reconstruct, respectively. We argue that this is because images with red and black hair make up the lowest and largest (>40%) proportion of all data, respectively. Thus, the models are unfamiliar/familiar with these attributes.

In addition, we also notice that females (F) are more challenging than males (M) for almost all models, since women usually have diverse accessories and hairstyles. For example, the models tend to perform worse on images with earrings (w/ E) than those without. Also, the facial images with smiles are more challenging than those without smiles. Interestingly, existing models achieve

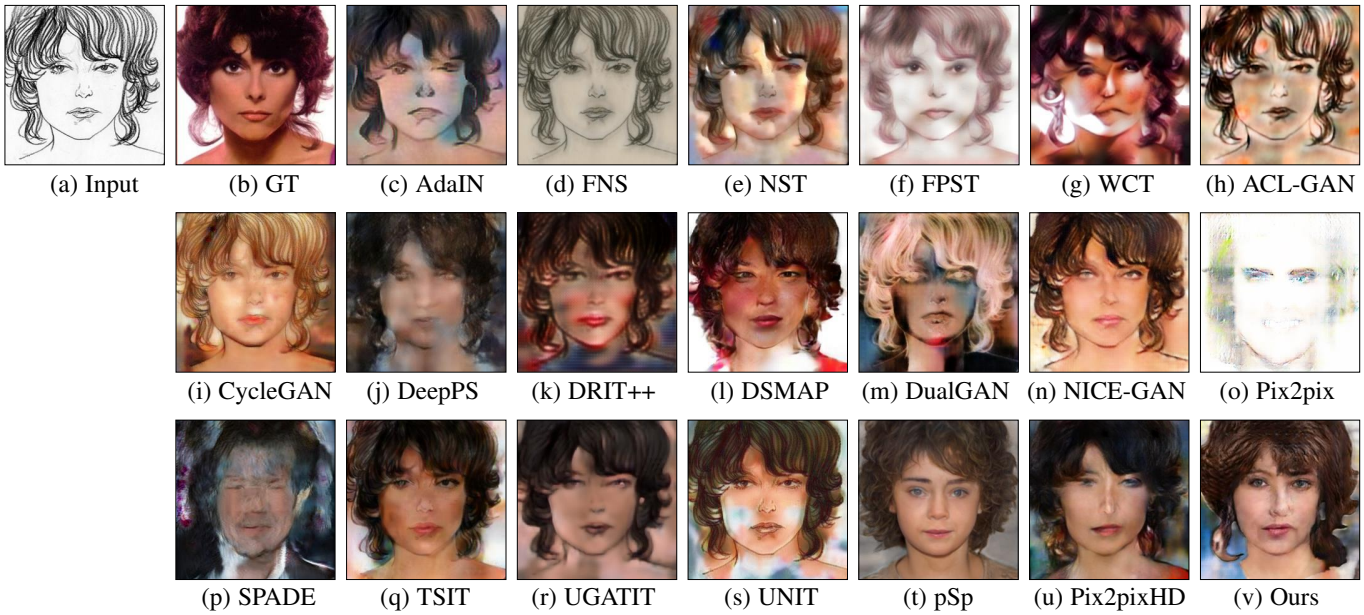


Fig. 12. We select 19 classical models, including AdaIN [149], FNS [136], FPST [140], WCT [153], ACL-GAN [198], CycleGAN [20], DeepPS [227], DRIT++ [27], DSMAP [26], DualGAN [173], NICE-GAN [21], Pix2pix [23], SPADE [25], TSIT [197], UGATIT [22], UNIT [24], pSp [171], and Pix2pixHD [19], for qualitative comparison.

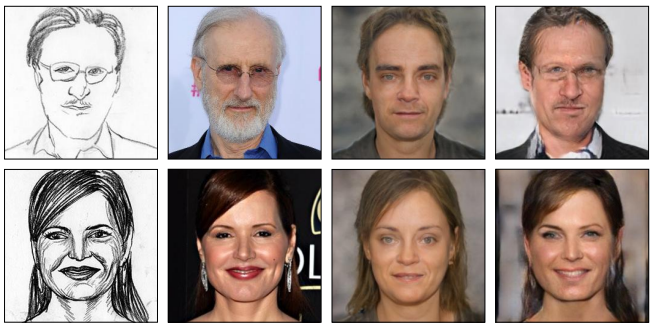


Fig. 13. Visual diversity of the data generated for the S2I task.

diverse performance irrespective to the color or hair (*e.g.*, H(b), H(bl), H(r), H(g)). Finally, compared to style 1 (simple lines) and style 3 (*i.e.*, repeated wispy details), we see that style 2 (long strokes) is the most challenging for all models.

5.3.2 SSIM Metric Results

In addition to the SCOOT metric results, we also provide the SSIM metric results for the I2S task in Table 10.

Analysis. We find that the overall performance tends to be similar to the SCOOT metric results in terms of several key attributes, such as hair, gender, accessories, and style. However, we also observe some differences as follow: 1) Note that the sketches in our FS2K only provide limited smile expressions, therefore, converting a sketch with a simple smile into a variable face image is not easy (*e.g.*, Ours: SSIM <0.50). 2) The performance on "w/ F" is lower than on "w/o F", as shown in Table 10. One possible reason is that frontal faces preserve more structural features than non-frontal ones. Therefore, in the S2I task, images with attributes such as "w/ F" are more challenging than those "w/o F".

TABLE 7. Ablation study of our FSGAN model on the I2S task.

Setting	multi-patch	style vec.	SCOOT↑	SSIM↑
Baseline			0.381	0.487
	✓		0.386(+1.31%)	0.500(+2.67%)
FSGAN (Ours)	✓	✓	**0.405**(+6.30%)	**0.510**(+4.72%)

TABLE 8. Ablation study of our model on the S2I task.

Setting	multi-patch	SSIM↑
Baseline		0.487
FSGAN (Ours)	✓	**0.503**(+3.3%)

5.4 Ablation Study

In this section, we provide a detailed analysis of our FSGAN on the proposed FS2K dataset. Unlike most existing facial synthesis models [19], our model has a two-stage GAN architecture for both I2S and S2I tasks. Besides, a sketch style vector is introduced to enable diversified style synthesis in the second stage of the I2S task. Therefore, the ablation studies on the I2S task are conducted on the following two key components: (1) the multi-patch generation stage, and (2) the style vector assisted generation. Note that we adopt the same hyperparameters described in Sec. 4.5 during our ablation experiments.

Table 7 shows the ablation results for the I2S task. We find that the multi-patch generation stage increases the SCOOT and SSIM scores by 1.31% (relative) and 2.67%, respectively, while the style vector further increases them by 6.30% and 4.72%. Further, as illustrated in Fig. 14, without the multi-patch strategy, the lines in the synthesized lips are often missing structure details. Meanwhile, with the multi-patch stage, the lines become smoother. Moreover, the synthesized drawings are messier without the style vector component and may introduce shadows in the lip regions.

For the S2I task, an ablation study is conducted to validate the effectiveness of the multi-patch generation stage, as shown in Table 8. Similar to the I2S task, we find that the multi-patch component achieves a large performance gain (*i.e.*, 3.3%) over the baseline model. Fig. 15 provides examples of the results

TABLE 9. Comparison of 19 state-of-the-art models in terms of attribute-based performance on the I2S task. Here, w/ H = hair visible, w/o H = hair invisible, H(b) = brown hair, H(bl) = black hair, H(r) = red hair, H(g) = golden hair, M = male, F = female, w/ E = with earring, w/o E = without earring, w/ S = with smile, w/o S = without smile, w/ F = frontal face, w/o F = non-frontal face, S1 = style1, S2 = style2, and S3 = style3.

Model	SCOOT↑																
	w/ H	w/o H	H(b)	H(bl)	H(r)	H(g)	M	F	w/ E	w/o E	w/ S	w/o S	w/ F	w/o F	S1	S2	S3
DualGAN [173]	0.260	0.279	0.250	0.267	0.216	0.279	0.275	0.240	0.239	0.266	0.255	0.271	0.261	0.262	0.298	0.194	0.319
FPST [140]	0.269	0.304	0.254	0.294	0.214	0.304	0.288	0.245	0.246	0.276	0.262	0.286	0.269	0.278	0.329	0.168	0.332
NST [133], [134]	0.272	0.283	0.268	0.287	0.236	0.283	0.280	0.262	0.258	0.276	0.268	0.282	0.272	0.276	0.310	0.205	0.332
Pix2pix [23]	0.272	0.335	0.255	0.300	0.217	0.335	0.298	0.240	0.250	0.281	0.267	0.290	0.276	0.272	0.333	0.178	0.302
ACL-GAN [198]	0.276	0.309	0.265	0.298	0.226	0.309	0.292	0.256	0.254	0.283	0.270	0.291	0.276	0.284	0.330	0.183	0.355
WCT [153]	0.281	0.315	0.271	0.302	0.229	0.315	0.296	0.261	0.262	0.287	0.277	0.292	0.281	0.290	0.332	0.195	0.346
AdaIN [149]	0.303	0.295	0.307	0.317	0.258	0.295	0.306	0.298	0.283	0.307	0.298	0.310	0.300	0.314	0.348	0.215	0.419
UNIT [24]	0.301	0.364	0.292	0.328	0.225	0.364	0.330	0.265	0.261	0.313	0.293	0.324	0.301	0.319	0.376	0.175	0.411
TSIT [197]	0.307	0.307	0.308	0.320	0.259	0.307	0.320	0.288	0.283	0.313	0.300	0.320	0.306	0.316	0.359	0.208	0.432
DRIT++ [27]	0.305	0.348	0.291	0.336	0.248	0.348	0.329	0.276	0.279	0.314	0.299	0.323	0.305	0.323	0.380	0.181	0.378
CartoonGAN [156]	0.319	0.318	0.320	0.337	0.262	0.318	0.329	0.304	0.291	0.325	0.314	0.329	0.317	0.332	0.382	0.204	0.428
UGATIT [22]	0.321	0.365	0.315	0.347	0.265	0.365	0.339	0.298	0.298	0.328	0.314	0.338	0.322	0.325	0.391	0.204	0.400
NICE-GAN [21]	0.325	0.355	0.320	0.357	0.262	0.355	0.342	0.303	0.302	0.332	0.317	0.343	0.325	0.333	0.398	0.201	0.401
CycleGAN [20]	0.348	0.343	0.358	0.362	0.287	0.343	0.351	0.343	0.326	0.353	0.341	0.360	0.346	0.357	0.397	0.252	0.483
MDAL [215]	0.354	0.363	0.348	0.380	0.292	0.363	0.369	0.333	0.329	0.360	0.345	0.372	0.352	0.365	0.436	0.211	0.446
UPDG [36]	0.362	0.411	0.349	0.390	0.290	0.411	0.390	0.325	0.336	0.371	0.356	0.379	0.363	0.370	0.423	0.259	0.448
APDrawing [2]	0.374	0.395	0.372	0.399	0.322	0.395	0.380	0.369	0.356	0.380	0.370	0.385	0.373	0.390	0.456	0.227	0.524
Pix2pixHD [19]	0.374	0.392	0.365	0.403	0.307	0.385	0.392	0.351	0.343	0.378	0.371	0.392	0.371	0.381	0.462	0.212	0.508
DSMAP [26]	0.375	0.431	0.357	0.405	0.322	0.431	0.400	0.343	0.354	0.383	0.369	0.393	0.377	0.381	0.437	**0.276**	0.423
FSGAN (Ours)	**0.403**	**0.435**	**0.389**	**0.435**	**0.335**	**0.435**	**0.423**	**0.377**	**0.381**	**0.410**	**0.395**	**0.422**	**0.403**	**0.414**	**0.481**	0.268	**0.509**

TABLE 10. Comparison of 19 top models in terms of attribute-based performance on the I2S task. Please refer to Table 9 for details.

Model	SSIM↑																
	w/ H	w/o H	H(b)	H(bl)	H(r)	H(g)	M	F	w/ E	w/o E	w/ S	w/o S	w/ F	w/o F	S1	S2	S3
DualGAN [173]	0.320	0.393	0.310	0.342	0.276	0.393	0.352	0.282	0.292	0.331	0.313	0.343	0.318	0.354	0.364	0.247	0.424
FPST [140]	0.459	0.481	0.442	0.492	0.383	0.481	0.492	0.411	0.416	0.469	0.448	0.481	0.455	0.486	0.517	0.351	0.597
NST [133], [134]	0.325	0.347	0.317	0.349	0.256	0.347	0.339	0.306	0.305	0.330	0.316	0.344	0.324	0.338	0.372	0.241	0.417
Pix2pix [23]	0.434	0.526	0.410	0.470	0.332	0.526	0.478	0.377	0.391	0.449	0.425	0.461	0.438	0.439	0.503	0.319	0.558
ACL-GAN [198]	0.402	0.432	0.392	0.430	0.334	0.432	0.427	0.369	0.363	0.413	0.393	0.423	0.398	0.434	0.445	0.316	0.583
WCT [153]	0.368	0.389	0.368	0.387	0.316	0.389	0.389	0.339	0.334	0.377	0.362	0.381	0.367	0.380	0.407	0.297	0.461
AdaIN [149]	0.364	0.367	0.364	0.382	0.319	0.367	0.378	0.343	0.340	0.370	0.359	0.375	0.362	0.379	0.399	0.297	0.460
UNIT [24]	0.501	0.556	0.488	0.528	0.421	0.556	0.539	0.450	0.460	0.514	0.492	0.526	0.498	0.532	0.563	0.395	0.616
TSIT [197]	0.439	0.465	0.430	0.461	0.371	0.465	0.465	0.404	0.408	0.448	0.431	0.458	0.435	0.468	0.485	0.351	0.587
DRIT++ [27]	0.490	0.534	0.479	0.519	0.411	0.534	0.524	0.444	0.451	0.501	0.480	0.512	0.487	0.515	0.547	0.387	0.617
CartoonGAN [156]	0.399	0.420	0.397	0.421	0.345	0.420	0.419	0.372	0.368	0.407	0.392	0.416	0.395	0.425	0.438	0.321	0.552
UGATIT [22]	0.455	0.497	0.445	0.476	0.386	0.497	0.489	0.409	0.416	0.466	0.447	0.476	0.451	0.491	0.499	0.373	0.593
NICE-GAN [21]	0.472	0.497	0.463	0.492	0.398	0.497	0.505	0.424	0.429	0.483	0.464	0.490	0.468	0.498	0.518	0.384	0.603
CycleGAN [20]	0.433	0.461	0.429	0.455	0.374	0.461	0.460	0.395	0.401	0.442	0.425	0.452	0.429	0.463	0.471	0.358	0.580
MDAL [215]	0.465	0.487	0.457	0.491	0.399	0.487	0.496	0.420	0.426	0.475	0.458	0.481	0.462	0.488	0.506	0.386	0.593
UPDG [36]	0.468	0.507	0.456	0.500	0.391	0.507	0.501	0.424	0.431	0.479	0.459	0.493	0.465	0.501	0.534	0.355	0.584
APDrawing [2]	0.461	0.522	0.441	0.497	0.373	0.522	0.504	0.402	0.419	0.473	0.452	0.484	0.458	0.492	0.512	0.371	0.582
Pix2pixHD [19]	0.492	0.552	0.473	0.523	0.419	0.546	0.531	0.431	0.457	0.505	0.481	0.513	0.488	0.524	0.537	0.402	0.618
DSMAP [26]	0.490	0.551	0.472	0.527	0.405	0.551	0.532	0.433	0.447	0.503	0.481	0.515	0.488	0.518	0.557	0.373	0.622
FSGAN (Ours)	**0.507**	**0.565**	**0.491**	**0.539**	**0.424**	**0.565**	**0.549**	**0.451**	**0.466**	**0.520**	**0.498**	**0.531**	**0.505**	**0.534**	**0.568**	**0.403**	**0.629**

produced by our model and the model without the multi-patch generation stage. As we can see, our model with multi-patch generation captures more details and ensures more realistic overall appearance (see Fig. 15(c)).

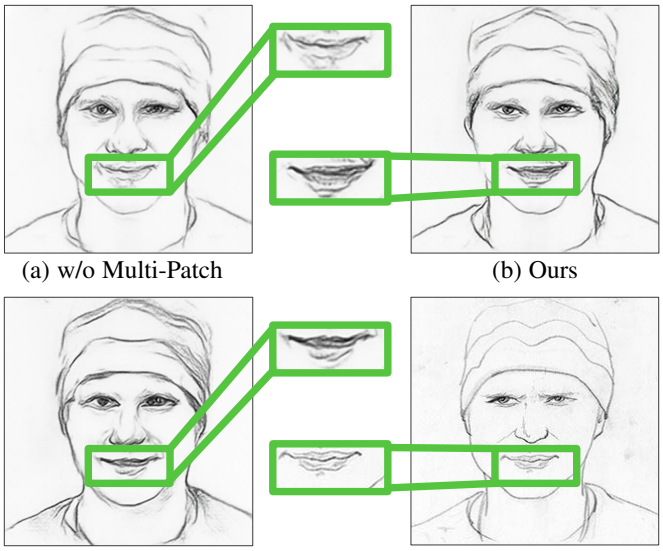
6 Discussion and Future Directions

Although human facial sketch synthesis has achieved significant progress, there is still a large room for improvement. In this section, we summarize the possible future research directions related to FSS, as follows.

(1) Datasets. Due to the relative shortage of professional sketch artists, achieving large numbers of images remains an open problem, impeding the development of FSS. Furthermore, more diversified sketch (or drawing) styles are needed for building more attractive models and achieving better synthesis results. To address these issues, we believe novel data augmentation techniques [103], [267], [268] and transfer learning strategies [269]–[271] specifically designed for FSS are promising directions of study.

(2) Models. Currently, most SOTA models are trained with a large number of paired images and sketches [16], [19] to overcome data shortages. However, more attention could be paid to techniques like few-shot [272], semi-supervised [273], weakly-supervised [274] and self-supervised [275] learning to achieve the style transfer with limited datasets. Besides, developing novel human-in-the-loop [276] models is another promising direction, which would provide more interactive options to users for generating and editing personalized styles. Interactive models could also serve as drawing tools provided to professional artists for facilitating the creation of sketches and other styles of drawing. Furthermore, FSS in the wild is still challenging, because the image quality, including resolution, noise, and background, varies drastically. In addition to the above-mentioned techniques, basic model units could also be focused on for the development of new techniques. For example, most current models are built upon CNN [277] units. Therefore, more exploration of other frameworks, such as MLPs [278] and Transformers [279], [280], could also be conducted.

(3) Evaluation. Evaluation metrics are essential for the development of new models and the benchmarking of existing ones. Currently, several quantitative evaluation metrics [10], [281] and human visual ranking methods [64] are used. However, as these aim to provide relatively objective and fair comparisons between all models, the different applications of FSS are not taken into consideration. This may lead to biased or unreliable evaluation on certain tasks. Therefore, more task-specific evaluation metrics and methods could be another important direction for future research.



(a) w/o Multi-Patch (b) Ours

(c) w/o Style Vector (d) GT

Fig. 14. Ablation study for the I2S task. We provide the results produced by FSGAN without the first-stage multi-patch generation (w/o Multi-Path); the results produced without the style vector (w/o Style Vector); and the results produced by our final FSGAN (Ours).

(a) Input (b) w/o multi-patch (c) Ours

Fig. 15. Ablation study for the S2I task: (a) The results produced by our FSGAN model without the first stage, *i.e.*, multi-patch generation; (b) the results produced by our final model.

(4) Applications. Currently, the only direct applications of FSS (I2S and S2I) are entertainment and law enforcement [1], [43]. With the development of FSS techniques, many other promising applications could also be implicitly or explicitly facilitated by FSS research, such as art design, animation production and so on. In addition to these industry applications, we believe that FSS methods and ideas could also benefit other fields of research. For example, sketches could be used to assist image resizing [282], super-resolution [283], *etc*. Further, the sketches usually contain the most conspicuous information of an image and can be therefore be considered compressed versions of RGB images. This characteristic makes sketches useful for the image compression task. Besides, the S2I task can be considered a specific case of image super-resolution in a broad sense, because both tasks aim to reconstruct detailed RGB images from the given inputs. The only difference is that the input of S2I is high-frequency information, while that of the standard super-resolution task is the low-frequency information of the original image.

7 Conclusion

We have presented a new challenge for the facial sketch synthesis (FSS) problem. To the best of our knowledge, this is the first systematic study on deep FSS in terms of both sketch-to-image and image-to-sketch tasks. To achieve this, we established a new challenging dataset, named FS2K. We also introduced a copy table for the proposed FS2K to address the alignment issue between the sketches drawn by artists and the original images. With a two-stage architecture, our proposed simple baseline, FSGAN, achieves the new state-of-the-art performance. Finally, as the largest existing survey (*i.e.*, 139 literature methods) and benchmark (*i.e.*, of 19 cutting-edge models), we have revealed that the development of this field is still in its infancy. The main goal of this investigation is therefore to spark novel ideas rather than rank all benchmarked works. It isn't easy to benchmark all of the existing models due to the prosperity of the field. We hope this investigation will attract the community's attention and yield exciting follow-up directions, such as generating vivid sketches with music, developing cartoons from sketches, synthesizing sketch videos, *etc*.

References

[1] X. Wang and X. Tang, "Face photo-sketch synthesis and recognition," *IEEE TPAMI*, vol. 31, no. 11, pp. 1955–1967, 2009.

[2] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, "APDrawingGAN: Generating artistic portrait drawings from face photos with hierarchical GANs," in *CVPR*, 2019, pp. 10 743–10 752.

[3] H. Koshimizu, M. Tominaga, T. Fujiwara, and K. Murakami, "On kansei facial image processing for computerized facial caricaturing system picasso," in *CSMC*, vol. 6, 1999, pp. 294–299.

[4] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *ICCV*, 2009, pp. 365–372.

[5] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," UMass Amherst technical report, Tech. Rep., 2010.

[6] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *ECCV*, 2014, pp. 94–108.

[7] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks)," in *ICCV*, 2017, pp. 1021–1030.

[8] J. Sun, Q. Li, W. Wang, J. Zhao, and Z. Sun, "Multi-caption text-to-face synthesis: Dataset and algorithm," in *ACMMM*, 2021, pp. 2290–2298.

[9] R. Yi, M. Xia, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, "Line drawings for face portraits from photos using global and local structure based GANs," *IEEE TPAMI*, vol. 43, no. 10, pp. 3462–3475, 2020.

[10] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE TIP*, vol. 13, no. 4, pp. 600–612, 2004.

[11] H. S. Bhatt, S. Bharadwaj, R. Singh, and M. Vatsa, "On matching sketches with digital face images," in *BTAS*, 2010, pp. 1–7.

[12] W. Zhang, X. Wang, and X. Tang, "Coupled information-theoretic encoding for face photo-sketch recognition," in *CVPR*, 2011, pp. 513–520.

[13] N. Wang, X. Gao, D. Tao, and X. Li, "Face sketch-photo synthesis under multi-dictionary sparse representation framework," in *ICIG*, 2011, pp. 82–87.

[14] X. Gao, N. Wang, D. Tao, and X. Li, "Face sketch–photo synthesis and retrieval using sparse representation," *IEEE TCSVT*, vol. 22, no. 8, pp. 1213–1226, 2012.

[15] I. Berger, A. Shamir, M. Mahler, E. Carter, and J. Hodgins, "Style and abstraction in portrait sketching," *ACM TOG*, vol. 32, no. 4, pp. 1–12, 2013.

[16] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, "Unpaired portrait drawing generation via asymmetric cycle mapping," in *CVPR*, 2020, pp. 8217–8225.

[17] S. Zhang, R. Ji, J. Hu, X. Lu, and X. Li, "Face sketch synthesis by multidomain adversarial learning," *IEEE TNNLS*, vol. 30, no. 5, pp. 1419–1428, 2019.

[18] M. Zhu, J. Li, N. Wang, and X. Gao, "Knowledge distillation for face photo-sketch synthesis," *IEEE TNNLS*, pp. 1–14, 2020.

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

[20] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017, pp. 2223–2232.

[21] R. Chen, W. Huang, B. Huang, F. Sun, and B. Fang, "Reusing discriminators for encoding: Towards unsupervised image-to-image translation," in *CVPR*, 2020, pp. 8168–8177.

[22] J. Kim, M. Kim, H. Kang, and K. Lee, "U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation," in *ICLR*, 2020.

[23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017, pp. 1125–1134.

[24] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *NeurIPS*, 2017.

[25] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019, pp. 2337–2346.

[26] H.-Y. Chang, Z. Wang, and Y.-Y. Chuang, "Domain-specific mappings for generative adversarial style transfer," in *ECCV*, 2020, pp. 573–589.

[27] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. K. Singh, and M.-H. Yang, "DRIT++: Diverse image-to-image translation via disentangled representations," *IJCV*, vol. 128, pp. 2402–2417, 2020.

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

[29] D.-P. Fan, S. Zhang, Y.-H. Wu, Y. Liu, M.-M. Cheng, B. Ren, P. L. Rosin, and R. Ji, "Scoot: A perceptual metric for facial sketches," in *ICCV*, 2019, pp. 5612–5622.

[30] H. S. Bhatt, S. Bharadwaj, R. Singh, and M. Vatsa, "Memetically optimized MCWLD for matching sketches with digital face images," *IEEE TIFS*, vol. 7, no. 5, pp. 1522–1535, 2012.

[31] C. Peng, X. Gao, N. Wang, and J. Li, "Face recognition from multiple stylistic sketches: Scenarios, datasets, and evaluation," *PR*, vol. 84, pp. 262–272, 2018.

[32] A. M. Martinez, "The ar face database," *CVC Technical Report24*, 1998.

[33] K. Messer, J. Matas, J. Kittler, J. Luettin, G. Maitre *et al.*, "XM2VTSDB: The extended m2vts database," in *ICABPA*, vol. 964, 1999, pp. 965–966.

[34] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE TPAMI*, vol. 22, no. 10, pp. 1090–1104, 2000.

[35] Á. Serrano, I. M. de Diego, C. Conde, E. Cabello, L. Shen, and L. Bai, "Influence of wavelet frequency and orientation in an SVM-based parallel gabor PCA face verification system," in *ICIDEAL*, 2007, pp. 219–228.

[36] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, "Unpaired portrait drawing generation via asymmetric cycle mapping," in *CVPR*, 2020, pp. 8217–8225.

[37] M. Minear and D. C. Park, "A lifespan database of adult facial stimuli," *Behav. Res. Meth. Instrum. Comput.*, vol. 36, no. 4, pp. 630–633, 2004.

[38] J. Nishino, T. Kamyama, H. Shira, T. Odaka, and H. Ogura, "Linguistic knowledge acquisition system on facial caricature drawing system," in *ICFS*, 1999, pp. 1591–1596.

[39] S. Iwashita, Y. Takeda, and T. Onisawa, "Expressive facial caricature drawing," in *ICFS*, 1999, pp. 1597–1602.

[40] Y. Li and H. Kobatake, "Extraction of facial sketch image based on morphological processing," in *ICIP*, 1997, pp. 316–319.

[41] M. Tominaga, S. Fukuoka, K. Murakami, and H. Koshimizu, "Facial caricaturing with motion caricaturing in PICASSO system," in *ICAIM*, 1997, p. 30.

[42] S. E. Brennan, "Caricature generator," Ph.D. dissertation, MIT, 1982.

[43] N. Wang, D. Tao, X. Gao, X. Li, and J. Li, "A Comprehensive Survey to Face Hallucination," *IJCV*, vol. 106, no. 1, pp. 9 – 30, 2013.

[44] H. Chen, Y.-Q. Xu, H.-Y. Shum, S.-C. Zhu, and N.-N. Zheng, "Example-based facial sketch generation with non-parametric sampling," in *ICCV*, 2001, pp. 433–438.

[45] A. V. Nefian and M. H. Hayes III, "Face recognition using an embedded hmm," in *AVBPA*, 1999.

[46] X. Gao, J. Zhong, J. Li, and C. Tian, "Face sketch synthesis algorithm based on e-hmm and selective ensemble," *IEEE TCSVT*, vol. 18, no. 4, pp. 487–496, 2008.

[47] Z. Xu, H. Chen, S.-C. Zhu, and J. Luo, "A hierarchical compositional model for face representation and sketching," *IEEE TPAMI*, vol. 30, no. 6, pp. 955–969, 2008.

[48] W. Zhang, X. Wang, and X. Tang, "Lighting and pose robust face sketch synthesis," in *ECCV*, 2010, pp. 420–433.

[49] H. Zhou, Z. Kuang, and K.-Y. K. Wong, "Markov weight fields for face sketch synthesis," in *CVPR*, 2012, pp. 1091–1097.

[50] T. Wang, J. P. Collomosse, A. Hunter, and D. Greig, "Learnable stroke models for example-based portrait painting," in *BMVC*, 2013.

[51] N. Wang, D. Tao, X. Gao, X. Li, and J. Li, "Transductive face sketch-photo synthesis," *IEEE TNNLS*, vol. 24, no. 9, pp. 1364–1376, 2013.

[52] C. Peng, X. Gao, N. Wang, and J. Li, "Superpixel-based face sketch–photo synthesis," *IEEE TCSVT*, vol. 27, no. 2, pp. 288–299, 2015.

[53] C. Peng, X. Gao, N. Wang, D. Tao, X. Li, and J. Li, "Multiple representations-based face sketch–photo synthesis," *IEEE TNNLS*, vol. 27, no. 11, pp. 2201–2215, 2015.

[54] N. Ji, X. Chai, S. Shan, and X. Chen, "Local regression model for automatic face sketch generation," in *ICIG*, 2011, pp. 412–417.

[55] L. Chang, M. Zhou, X. Deng, Z. Wu, and Y. Han, "Face sketch synthesis via multivariate output regression," in *HCI*, 2011, pp. 555–561.

[56] J. Zhang, N. Wang, X. Gao, D. Tao, and X. Li, "Face sketch-photo synthesis based on support vector regression," in *ICIP*, 2011, pp. 1125–1128.

[57] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[58] S. Wang, L. Zhang, Y. Liang, and Q. Pan, "Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis," in *CVPR*, 2012, pp. 2216–2223.

[59] X. Tang and X. Wang, "Face photo recognition using sketch," in *ICIP*, 2002, pp. I–I.

[60] X.Tang and X. Wang, "Face sketch synthesis and recognition," in *ICCV*, 2003, pp. 687–694.

[61] X. Tang and X. Wang, "Face sketch recognition," *IEEE TCSVT*, vol. 14, no. 1, pp. 50–57, 2004.

[62] Q. Liu, X. Tang, H. Jin, H. Lu, and S. Ma, "A nonlinear approach for face sketch synthesis and recognition," in *CVPR*, vol. 1, 2005, pp. 1005–1010.

[63] D.-A. Huang and Y.-C. F. Wang, "Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition," in *ICCV*, 2013, pp. 2496–2503.

[64] Y. Song, L. Bao, Q. Yang, and M.-H. Yang, "Real-time exemplar-based face sketch synthesis," in *ECCV*, 2014, pp. 800–813.

[65] N. Wang, X. Gao, and J. Li, "Random sampling for fast face sketch synthesis," *PR*, vol. 76, pp. 215–227, 2018.

[66] S. Zhang, X. Gao, N. Wang, and J. Li, "Robust face sketch style synthesis," *IEEE TIP*, vol. 25, no. 1, pp. 220–232, 2015.

[67] Y. Li, Y.-Z. Song, T. M. Hospedales, and S. Gong, "Free-hand sketch synthesis with deformable stroke models," *IJCV*, vol. 122, no. 1, pp. 169–190, 2017.

[68] J. Li, X. Yu, C. Peng, and N. Wang, "Adaptive representation-based face sketch-photo synthesis," *Neurocomputing*, vol. 269, pp. 152–159, 2017.

[69] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE TVCG*, vol. 26, no. 11, pp. 3365–3385, 2019.

[70] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM TOG*, vol. 31, no. 4, pp. 44:1–44:10, 2012.

[71] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The feret database and evaluation procedure for face-recognition algorithms," *IMAVIS*, vol. 16, no. 5, pp. 295–306, 1998.

[72] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[73] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[74] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand, "Style transfer for headshot portraits," *ACM TOG*, vol. 33, no. 4, 2014.

[75] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *ECCV*, 2012, pp. 611–625.

[76] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black, "Lessons and insights from creating a synthetic optical flow benchmark," in *ECCV*, 2012, pp. 168–177.

[77] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *CVPR*, 2016, pp. 724–732.

[78] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *CVPR*, 2014, pp. 3606–3613.

[79] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *NeurIPS*, 2014.

[80] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *WACV*, 2014, pp. 75–82.

[81] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *CVPR*, 2016, pp. 4040–4048.

[82] S. Y. Duck, "Painter by numbers, wikiart.org," in *https://www.kaggle.com/c/painter-by-numbers*, 2016.

[83] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016, pp. 3213–3223.

[84] R. Tyleček and R. Šára, "Spatial pattern templates for recognition of objects with regular structure," in *GCPR*, 2013, pp. 364–374.

[85] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *ECCV*, 2016, pp. 597–613.

[86] A. Yu and K. Grauman, "Fine-grained visual comparisons with local learning," in *CVPR*, 2014, pp. 192–199.

[87] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM TOG*, vol. 31, no. 4, pp. 1–10, 2012.

[88] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays, "Transient attributes for high-level understanding and editing of outdoor scenes," *ACM TOG*, vol. 33, no. 4, pp. 1–11, 2014.

[89] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[90] J. Friedman, T. Hastie, R. Tibshirani *et al.*, *The elements of statistical learning*, 2001, vol. 1, no. 10.

[91] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[92] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[93] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *ICCVGI*, 2008, pp. 722–729.

[94] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[95] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *ICLR*, 2018.

[96] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012, pp. 746–760.

[97] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *CVPR*, 2017, pp. 633–641.

[98] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *ECCV*, 2012, pp. 679–692.

[99] B. M. Smith, L. Zhang, J. Brandt, Z. Lin, and J. Yang, "Exemplar-based face parsing," in *CVPR*, 2013, pp. 3484–3491.

[100] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *ICIP*, 2014, pp. 343–347.

[101] O. Langner, R. Dotsch, G. Bijlstra, D. H. Wigboldus, S. T. Hawk, and A. Van Knippenberg, "Presentation and validation of the radboud faces database," *CM*, vol. 24, no. 8, pp. 1377–1388, 2010.

[102] C. Fabian Benitez-Quiroz, R. Srinivasan, and A. M. Martinez, "Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild," in *CVPR*, 2016, pp. 5562–5570.

[103] Q. Yu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketchx!-shoe/chair fine-grained SBIR dataset," 2017.

[104] D. Ha and D. Eck, "A neural representation of sketch drawings," in *ICLR*, 2018.

[105] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *CVPRW*, vol. 2, no. 1, 2011.

[106] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015, pp. 41.1–41.12.

[107] T. Li, R. Qian, C. Dong, S. Liu, Q. Yan, W. Zhu, and L. Lin, "BeautyGAN: Instance-level facial makeup transfer with deep generative adversarial network," in *ACM MM*, 2018, pp. 645–653.

[108] Y. Jin, J. Zhang, M. Li, Y. Tian, H. Zhu, and Z. Fang, "Towards the automatic anime characters creation with generative adversarial networks," in *NeurIPSW*, 2017.

[109] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest *et al.*, "The multimodal brain tumor image segmentation benchmark (BRATS)," *IEEE TMI*, vol. 34, no. 10, pp. 1993–2024, 2014.

[110] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *ECCV*, 2016, pp. 494–509.

[111] S. Workman, R. Souvenir, and N. Jacobs, "Wide-area image geolocalization with aerial reference imagery," in *ICCV*, 2015, pp. 3961–3969.

[112] S. Ardeshir and A. Borji, "Ego2top: Matching viewers in egocentric and top-view videos," in *ECCV*, 2016, pp. 253–268.

[113] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *ICCV*, 2019, pp. 10 551–10 560.

[114] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie, "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection," in *CVPR*, 2015, pp. 595–604.

[115] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *ECCV*, 2014, pp. 3–17.

[116] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *ECCV*, 2016, pp. 102–118.

[117] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *CVPR*, 2017, pp. 2174–2182.

[118] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *CVPR*, 2016, pp. 3234–3243.

[119] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *CVPR*, 2016, pp. 1096–1104.

[120] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019, pp. 4401–4410.

[121] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *CVPRW*, 2017, pp. 126–135.

[122] B. Yao, X. Yang, and S.-C. Zhu, "Introduction to a large-scale general purpose ground truth database: methodology, annotation tool and benchmarks," in *CVPRW*, 2007, pp. 169–183.

[123] P. H. Seo, A. Lehrmann, B. Han, and L. Sigal, "Visual reference resolution using attention memory for visual dialog," in *NeurIPS*, 2017, pp. 3719–3729.

[124] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *ICCVW*, 2013, pp. 554–561.

[125] M. J. Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. Belongie, "Bam! the behance artistic media dataset for recognition beyond photography," in *ICCV*, 2017, pp. 1202–1211.

[126] S. Kalkowski, C. Schulze, A. Dengel, and D. Borth, "Real-time analysis and visualization of the yfcc100m dataset," in *ONSW*, 2015, pp. 25–30.

[127] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *CVPR*, 2017, pp. 5810–5818.

[128] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *CVPR*, 2012, pp. 3498–3505.

[129] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[130] S. Fidler, S. Dickinson, and R. Urtasun, "3d object detection and viewpoint estimation with a deformable 3d cuboid model," *NeurIPS*, 2012.

[131] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "StarGAN v2: Diverse image synthesis for multiple domains," in *CVPR*, 2020, pp. 8188–8197.

[132] S. Ge, V. Goswami, C. L. Zitnick, and D. Parikh, "Creative sketch generation," in *ICLR*, 2021.

[133] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[134] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *CVPR*, 2016, pp. 2414–2423.

[135] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," in *CVPR*, 2016, pp. 2479–2486.

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

[137] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *ECCV*, 2016, pp. 702–716.

[138] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in *ICML*, 2016, p. 1349–1357.

[139] A. Selim, M. Elgharib, and L. Doyle, "Painting style transfer for head portraits using convolutional neural networks," *ACM TOG*, vol. 35, no. 4, pp. 1–18, 2016.

[140] T. Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," in *NeurIPSW*, 2016.

[141] G. Berger and R. Memisevic, "Incorporating long-range consistency in cnn-based texture generation," in *ICLR*, 2017.

[142] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *ICLR*, 2017.

[143] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," in *CVPR*, 2017, pp. 3985–3993.

[144] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," in *CVPR*, 2017, pp. 4990–4998.

[145] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Diversified texture synthesis with feed-forward networks," in *CVPR*, 2017, pp. 3920–3928.

[146] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stylebank: An explicit representation for neural image style transfer," in *CVPR*, 2017, pp. 1897–1906.

[147] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *CVPR*, 2017, pp. 6924–6932.

[148] X. Wang, G. Oxholm, D. Zhang, and Y.-F. Wang, "Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer," in *CVPR*, 2017, pp. 5239–5247.

[149] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *ICCV*, 2017, pp. 1501–1510.

[150] A. Gupta, J. Johnson, A. Alahi, and L. Fei-Fei, "Characterizing and improving stability in neural style transfer," in *ICCV*, 2017, pp. 4067–4076.

[151] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[152] M. Lu, H. Zhao, A. Yao, F. Xu, Y. Chen, and L. Zhang, "Decoder network over lightweight reconstructed feature for fast semantic style transfer," in *ICCV*, 2017, pp. 2469–2477.

[153] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *NeurIPS*, 2017, pp. 385—-395.

[154] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang, "Visual attribute transfer through deep image analogy," *ACM TOG*, vol. 36, no. 4, pp. 1–15, 2017.

[155] S. Gu, C. Chen, J. Liao, and L. Yuan, "Arbitrary style transfer with deep feature reshuffle," in *CVPR*, 2018, pp. 8222–8231.

[156] Y. Chen, Y.-K. Lai, and Y.-J. Liu, "Cartoongan: Generative adversarial networks for photo cartoonization," in *CVPR*, 2018, pp. 9465–9474.

[157] F. Shen, S. Yan, and G. Zeng, "Neural style transfer via meta networks," in *CVPR*, 2018, pp. 8061–8069.

[158] L. Sheng, Z. Lin, J. Shao, and X. Wang, "Avatar-net: Multi-scale zero-shot style transfer by feature decoration," in *CVPR*, 2018, pp. 8242–8250.

[159] Y. Men, Z. Lian, Y. Tang, and J. Xiao, "A common framework for interactive texture transfer," in *CVPR*, 2018, pp. 6353–6362.

[160] Y. Zhang, Y. Zhang, and W. Cai, "Separating style and content for generalized style transfer," in *CVPR*, 2018, pp. 8447–8455.

[161] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stereoscopic neural style transfer," in *CVPR*, 2018, pp. 6654–6663.

[162] R. Mechrez, I. Talmi, and L. Zelnik-Manor, "The contextual loss for image transformation with non-aligned data," in *ECCV*, 2018, pp. 768–783.

[163] A. Sanakoyeu, D. Kotovenko, S. Lang, and B. Ommer, "A style-aware content loss for real-time hd style transfer," in *ECCV*, 2018, pp. 698–714.

[164] Y. Jing, Y. Liu, Y. Yang, Z. Feng, Y. Yu, D. Tao, and M. Song, "Stroke controllable fast style transfer with adaptive receptive fields," in *ECCV*, 2018, pp. 238–254.

[165] X. Li, S. Liu, J. Kautz, and M.-H. Yang, "Learning linear transformations for fast image and video style transfer," in *CVPR*, 2019, pp. 3809–3817.

[166] D. Y. Park and K. H. Lee, "Arbitrary style transfer with style-attentional networks," in *CVPR*, 2019, pp. 5880–5888.

[167] R. Abdal, Y. Qin, and P. Wonka, "Image2styleGAN: How to embed images into the stylegan latent space?" in *ICCV*, 2019, pp. 4432–4441.

[168] Y. Jing, X. Liu, Y. Ding, X. Wang, E. Ding, M. Song, and S. Wen, "Dynamic instance normalization for arbitrary style transfer," in *AAAI*, vol. 34, no. 04, 2020, pp. 4369–4376.

[169] D. Kotovenko, M. Wright, A. Heimbrecht, and B. Ommer, "Rethinking style transfer: From pixels to parameterized brushstrokes," *CVPR*, pp. 12 196–12 205, 2021.

[170] T. Lin, Z. Ma, F. Li, D. He, X. Li, E. Ding, N. Wang, J. Li, and X. Gao, "Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer," in *CVPR*, 2021, pp. 5141–5150.

[171] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in style: a stylegan encoder for image-to-image translation," in *CVPR*, 2021, pp. 2287–2296.

[172] H. Dong, S. Yu, C. Wu, and Y. Guo, "Semantic image synthesis via adversarial learning," in *ICCV*, 2017, pp. 5706–5714.

[173] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *ICCV*, 2017, pp. 2849–2857.

[174] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *ICML*, 2017, pp. 1857–1865.

[175] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," in *ICLR*, 2017.

[176] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *NeurIPS*, 2017, pp. 465–476.

[177] S. Benaim and L. Wolf, "One-sided unsupervised domain mapping," in *NeurIPS*, 2017, pp. 752–762.

[178] Z. Gan, L. Chen, W. Wang, Y. Pu, Y. Zhang, H. Liu, C. Li, and L. Carin, "Triangle generative adversarial networks," in *NeurIPS*, 2017, pp. 5253–5262.

[179] S. Ma, J. Fu, C. W. Chen, and T. Mei, "Da-gan: Instance-level image translation by deep attention generative adversarial networks," in *CVPR*, 2018, pp. 5657–5666.

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

[181] B. Zhao, B. Chang, Z. Jie, and L. Sigal, "Modular generative adversarial networks," in *ECCV*, 2018, pp. 150–165.

[182] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "Ganimation: Anatomically-aware facial animation from a single image," in *ECCV*, 2018, pp. 818–833.

[183] M. Li, H. Huang, L. Ma, W. Liu, T. Zhang, and Y. Jiang, "Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks," in *ECCV*, 2018, pp. 184–199.

[184] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, 2018, pp. 172–189.

[185] T. Xiao, J. Hong, and J. Ma, "Elegant: Exchanging latent encodings with gan for transferring multiple face attributes," in *ECCV*, 2018, pp. 168–184.

[186] L. Ma, X. Jia, S. Georgoulis, T. Tuytelaars, and L. Van Gool, "Exemplar guided unsupervised image-to-image translation with semantic consistency," in *ICLR*, 2019.

[187] R. Zhang, T. Pfister, and J. Li, "Harmonic unpaired image-to-image translation," in *ICLR*, 2019.

[188] W. Cho, S. Choi, D. K. Park, I. Shin, and J. Choo, "Image-to-image translation via group-wise deep whitening-and-coloring transformation," in *CVPR*, 2019, pp. 10 639–10 647.

[189] W. Wu, K. Cao, C. Li, C. Qian, and C. C. Loy, "Transgaga: Geometry-aware unsupervised image-to-image translation," in *CVPR*, 2019, pp. 8012–8021.

[190] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *CVPR*, 2019, pp. 1429–1437.

[191] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, "Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation," in *CVPR*, 2019, pp. 2417–2426.

[192] H. Tang, H. Liu, D. Xu, P. H. Torr, and N. Sebe, "Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks," *IEEE TNNLS*, 2021.

[193] O. Sendik, D. Cohen-Or, and D. Lischinski, "Crossnet: Latent cross-consistency for unpaired image translation," in *WACV*, 2020, pp. 3043–3051.

[194] I. Anokhin, P. Solovev, D. Korzhenkov, A. Kharlamov, T. Khakhulin, A. Silvestrov, S. Nikolenko, V. Lempitsky, and G. Sterkin, "High-resolution daytime translation without domain labels," in *CVPR*, 2020, pp. 7488–7497.

[195] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, "Sean: Image synthesis with semantic region-adaptive normalization," in *CVPR*, 2020, pp. 5104–5113.

[196] P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen, "Cross-domain correspondence learning for exemplar-based image translation," in *CVPR*, 2020, pp. 5143–5153.

[197] L. Jiang, C. Zhang, M. Huang, C. Liu, J. Shi, and C. C. Loy, "TSIT: A simple and versatile framework for image-to-image translation," in *ECCV*, 2020.

[198] Y. Zhao, R. Wu, and H. Dong, "Unpaired image-to-image translation using adversarial consistency loss," in *ECCV*, 2020, pp. 800–815.

[199] S. Ramasinghe, K. Ranasinghe, S. Khan, N. Barnes, and S. Gould, "Conditional generative modeling via learning the latent space," in *ICLR*, 2021.

[200] Y. Xu, Y. Shen, J. Zhu, C. Yang, and B. Zhou, "Generative hierarchical features from synthesizing images," *CVPR*, 2021.

[201] R. Liu, Y. Ge, C. L. Choi, X. Wang, and H. Li, "Divco: Diverse conditional image synthesis via contrastive generative adversarial network," in *CVPR*, 2021, pp. 16 377–16 386.

[202] X. Zhou, B. Zhang, T. Zhang, P. Zhang, J. Bao, D. Chen, Z. Zhang, and F. Wen, "CoCosNet v2: Full-resolution correspondence learning for image translation," in *CVPR*, 2021, pp. 11 465–11 475.

[203] C. Meng, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Image synthesis and editing with stochastic differential equations," *arXiv*, 2021.

[204] L. Zhang, L. Lin, X. Wu, S. Ding, and L. Zhang, "End-to-end photo-sketch generation via fully convolutional representation learning," in *ICMR*, 2015, pp. 627–634.

[205] M. Zhu, N. Wang, X. Gao, and J. Li, "Deep graphical feature learning for face sketch synthesis," in *IJCAI*, 2017, pp. 3574–3580.

[206] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *CVPR*, 2017, pp. 5400–5409.

[207] M. Zhang, N. Wang, Y. Li, R. Wang, and X. Gao, "Face sketch synthesis from coarse to fine," in *AAAI*, 2018, pp. 7558–7565.

[208] W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, and J. Hays, "TextureGAN: Controlling deep image synthesis with texture patches," in *CVPR*, 2018, pp. 8456–8465.

[209] J. Song, K. Pang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Learning to sketch with shortcut cycle consistency," in *CVPR*, 2018, pp. 801–810.

[210] Y. Lu, S. Wu, Y.-W. Tai, and C.-K. Tang, "Image generation from sketch constraint using contextual GAN," in *ECCV*, 2018, pp. 205–220.

[211] S. Zhang, R. Ji, J. Hu, Y. Gao, and C.-W. Lin, "Robust face sketch synthesis via generative adversarial fusion of priors and parametric sigmoid," in *IJCAI*, 2018, pp. 1163–1169.

[212] M. Zhang, N. Wang, X. Gao, and Y. Li, "Markov random neural fields for face sketch synthesis," in *IJCAI*, 2018, pp. 1142–1148.

[213] L. Wang, V. Sindagi, and V. Patel, "High-quality facial photo-sketch synthesis using multi-adversarial networks," in *ICAFGR*, 2018, pp. 83–90.

[214] M. Zhang, R. Wang, X. Gao, J. Li, and D. Tao, "Dual-transfer face sketch–photo synthesis," *IEEE TIP*, vol. 28, no. 2, pp. 642–657, 2018.

[215] S. Zhang, R. Ji, J. Hu, X. Lu, and X. Li, "Face sketch synthesis by multidomain adversarial learning," *IEEE TNNLS*, vol. 30, no. 5, pp. 1419–1428, 2018.

[216] H. Kazemi, M. Iranmanesh, A. Dabouei, S. Soleymani, and N. M. Nasrabadi, "Facial attributes guided deep sketch-to-photo synthesis," in *WACVW*, 2018, pp. 1–8.

[217] H. Kazemi, F. Taherkhani, and N. M. Nasrabadi, "Unsupervised facial geometry learning for sketch to photo synthesis," in *BIOSIG*, 2018, pp. 1–5.

[218] S. You, N. You, and M. Pan, "Pi-rec: Progressive image reconstruction network with edge and color domain," *arXiv preprint arXiv:1903.10146*, 2019.

[219] M. Zhang, N. Wang, Y. Li, and X. Gao, "Deep latent low-rank representation for face sketch synthesis," *IEEE TNNLS*, vol. 30, no. 10, pp. 3109–3123, 2019.

[220] M. Zhu, J. Li, N. Wang, and X. Gao, "A deep collaborative framework for face photo–sketch synthesis," *IEEE TNNLS*, vol. 30, no. 10, pp. 3096–3108, 2019.

[221] M. Zhang, Y. Li, N. Wang, Y. Chi, and X. Gao, "Cascaded face sketch synthesis under various illuminations," *IEEE TIP*, vol. 29, pp. 1507–1521, 2019.

[222] M. Zhu, N. Wang, X. Gao, J. Li, and Z. Li, "Face photo-sketch synthesis via knowledge transfer," in *IJCAI*, 2019, pp. 1048–1054.

[223] Y. Li, C. Fang, A. Hertzmann, E. Shechtman, and M.-H. Yang, "Im2pencil: Controllable pencil illustration from photographs," in *CVPR*, 2019, pp. 1525–1534.

[224] A. Ghosh, R. Zhang, P. K. Dokania, O. Wang, A. A. Efros, P. H. Torr, and E. Shechtman, "Interactive sketch & fill: Multiclass sketch-to-image translation," in *ICCV*, 2019, pp. 1171–1180.

[225] X. Wang and J. Yu, "Learning to cartoonize using white-box cartoon representations," in *CVPR*, 2020, pp. 8090–8099.

[226] C. Gao, Q. Liu, Q. Xu, L. Wang, J. Liu, and C. Zou, "Sketchycoco: image generation from freehand scene sketches," in *CVPR*, 2020, pp. 5174–5183.

[227] S. Yang, Z. Wang, J. Liu, and Z. Guo, "Deep plastic surgery: Robust and controllable image editing with human-drawn sketches," in *ECCV*, 2020, pp. 601–617.

[228] S.-Y. Chen, W. Su, L. Gao, S. Xia, and H. Fu, "DeepFaceDrawing: Deep generation of face images from sketches," *ACM TOG*, vol. 39, no. 4, pp. 72–1, 2020.

[229] J. Yu, X. Xu, F. Gao, S. Shi, M. Wang, D. Tao, and Q. Huang, "Toward realistic face photo-sketch synthesis via composition-aided gans," *IEEE TCYB*, vol. 51, no. 9, pp. 4350–4362, 2020.

[230] Y. Fang, W. Deng, J. Du, and J. Hu, "Identity-aware CycleGAN for face photo-sketch synthesis and recognition," *PR*, vol. 102, p. 107249, 2020.

[231] Y. Lin, S. Ling, K. Fu, and P. Cheng, "An identity-preserved model for face sketch-photo synthesis," *IEEE SPL*, vol. 27, pp. 1095–1099, 2020.

[232] C. Peng, N. Wang, J. Li, and X. Gao, "Universal face photo-sketch style transfer via multiview domain translation," *IEEE TIP*, vol. 29, pp. 8519–8534, 2020.

[233] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[234] S. Duan, Z. Chen, Q. J. Wu, L. Cai, and D. Lu, "Multi-scale gradients self-attention residual learning for face photo-sketch transformation," *IEEE TIFS*, vol. 16, pp. 1218–1230, 2020.

[235] S.-Y. Wang, D. Bau, and J.-Y. Zhu, "Sketch your own GAN," in *ICCV*, 2021.

[236] A. K. Bhunia, S. Khan, H. Cholakkal, R. M. Anwer, F. S. Khan, J. Laaksonen, and M. Felsberg, "Doodleformer: Creative sketch drawing with transformers," *arXiv preprint arXiv:2112.03258*, 2021.

[237] Y. Deng, F. Tang, X. Pan, W. Dong, C. Xu *et al.*, "Stytr$\hat{2}$: Unbiased image style transfer with transformers," *arXiv preprint arXiv:2105.14576*, 2021.

[238] S. Saxena and M. N. Teli, "Comparison and analysis of image-to-image generative adversarial networks: A survey," 2021.

[239] M. Mirza and S. Osindero, "Conditional generative adversarial nets," in *NeurIPSW*, 2014.

[240] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.

[241] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *ICLR*, 2021.

[242] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *NeurIPS*, 2014, pp. 2672–2680.

[243] Y. Song, C. Yang, Y. Shen, P. Wang, Q. Huang, and C.-C. J. Kuo, "Spg-net: Segmentation prediction and guidance network for image inpainting," in *BMVC*, 2018.

[244] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[245] X. Zheng, Y. Guo, H. Huang, Y. Li, and R. He, "A survey of deep facial attribute analysis," *IJCV*, vol. 128, no. 8, pp. 2002–2034, 2020.

[246] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *FRLIW*, 2008.

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

[248] E. M. Hand and R. Chellappa, "Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification," in *AAAI*, 2017.

[249] H. Han, A. K. Jain, F. Wang, S. Shan, and X. Chen, "Heterogeneous face attribute estimation: A deep multi-task learning approach," *IEEE TPAMI*, vol. 40, no. 11, pp. 2597–2609, 2017.

[250] Y. Jang, H. Gunes, and I. Patras, "Smilenet: registration-free smiling face detection in the wild," in *ICCVW*, 2017, pp. 1581–1589.

[251] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-in-one convolutional neural network for face analysis," in *FG*, 2017, pp. 17–24.

[252] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE TAC*, 2020.

[253] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev, "Panda: Pose aligned networks for deep attribute modeling," in *CVPR*, 2014, pp. 1637–1644.

[254] M. Kan, S. Shan, H. Chang, and X. Chen, "Stacked progressive auto-encoders (spae) for face recognition across poses," in *CVPR*, 2014, pp. 1883–1890.
[255] Y. Wu, Z. Wang, and Q. Ji, "Facial feature tracking under varying facial expressions and face poses based on restricted boltzmann machines," in *CVPR*, 2013, pp. 3452–3459.
[256] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning GAN for pose-invariant face recognition," in *CVPR*, 2017, pp. 1415–1424.
[257] U. Toseeb, D. R. Keeble, and E. J. Bryant, "The significance of hair for face recognition," *PloS one*, vol. 7, no. 3, p. e34144, 2012.
[258] S. J. Bartel, K. Toews, L. Gronhovd, and S. L. Prime, "Do i know you? altering hairstyle affects facial recognition," *VC*, vol. 26, no. 3, pp. 149–155, 2018.
[259] N. Kumar, P. Belhumeur, and S. Nayar, "Facetracer: A search engine for large collections of images with faces," in *ECCV*, 2008, pp. 340–353.
[260] L. Huai-Yu, D. Wei-Ming, and B.-G. Hu, "Facial image attributes transformation via conditional recycle generative adversarial networks," *JCST*, vol. 33, no. 3, pp. 511–521, 2018.
[261] J.-S. Pierrard and T. Vetter, "Skin detail analysis for face recognition," in *CVPR*, 2007, pp. 1–8.
[262] S. Z. Li, *Encyclopedia of Biometrics: I-Z.*, 2009, vol. 2.
[263] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE SPL*, vol. 23, no. 10, pp. 1499–1503, 2016.
[264] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
[265] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NeurIPSW*, 2017.
[266] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2014.
[267] Q. Yu, F. Liu, Y.-Z. Song, T. Xiang, T. M. Hospedales, and C.-C. Loy, "Sketch me that shoe," in *CVPR*, 2016, pp. 799–807.
[268] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
[269] Y. Wang, C. Wu, L. Herranz, J. van de Weijer, A. Gonzalez-Garcia, and B. Raducanu, "Transferring gans: generating images from limited data," in *ECCV*, 2018, pp. 218–234.
[270] L. Yu, J. van de Weijer *et al.*, "Deepi2i: Enabling deep hierarchical image-to-image translation by transferring from gans," *NeurIPS*, pp. 11 803–11 815, 2020.
[271] A. Shocher, Y. Gandelsman, I. Mosseri, M. Yarom, M. Irani, W. T. Freeman, and T. Dekel, "Semantic pyramid for image generation," in *ICCV*, 2020, pp. 7457–7466.
[272] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *ICLR*, 2017.
[273] O. Chapelle, B. Scholkopf, and A. Zien, Eds., "Semi-supervised learning (chapelle, o. et al., eds.; 2006) [book reviews]," *IEEE TNN*, vol. 20, no. 3, pp. 542–542, 2009.
[274] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free? - weakly-supervised learning with convolutional neural networks," in *CVPR*, 2015, pp. 685–694.
[275] X. Wang, K. He, and A. Gupta, "Transitive invariance for self-supervised visual representation learning," in *ICCV*, 2017, pp. 1329–1338.
[276] R. Pinto, T. Mettler, and M. Taisch, "Managing supplier delivery reliability risk under limited information: Foundations for a human-in-the-loop DSS," *DSS*, vol. 54, no. 2, pp. 1076–1084, 2013.
[277] Y. LeCun *et al.*, "Generalization and network design strategies," *CIP*, vol. 19, pp. 143–155, 1989.
[278] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, D. Keysers, J. Uszkoreit, M. Lucic *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *arXiv preprint arXiv:2105.01601*, 2021.
[279] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.
[280] K. Lee, H. Chang, L. Jiang, H. Zhang, Z. Tu, and C. Liu, "ViTGAN: Training GANs with vision transformers," *arXiv preprint arXiv:2107.04589*, 2021.
[281] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "Fsim: A feature similarity index for image quality assessment," *IEEE TIP*, vol. 20, no. 8, pp. 2378–2386, 2011.
[282] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM TOG*, vol. 26, no. 3, p. 10, 2007.
[283] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE TPAMI*, vol. 38, no. 2, pp. 295–307, 2015.