

Deep Learning for Free-Hand Sketch: A Survey

Peng Xu

Abstract—Free-hand sketches are highly hieroglyphic and illustrative, which have been widely used by humans to depict objects or stories from ancient times to the present. The recent prevalence of touchscreen devices has made sketch creation a much easier task than ever and consequently made sketch-oriented applications increasingly more popular. The prosperity of deep learning has also immensely promoted the research for the free-hand sketch. This paper presents a comprehensive survey of the free-hand sketch oriented deep learning techniques. The main contents of this survey include: (i) The intrinsic traits and domain-unique challenges of the free-hand sketch are discussed, to clarify the essential differences between free-hand sketch and other data modalities, *e.g.*, natural photo. (ii) The development of the free-hand sketch community in the deep learning era is reviewed, by surveying the existing datasets, research topics, and the state-of-the-art methods via a detailed taxonomy. (iii) Moreover, the bottlenecks, open problems, and potential research directions of this community have also been discussed to promote the future works.

Index Terms—Free-Hand Sketch, Deep Learning, Survey, Introductory, Taxonomy.



1 INTRODUCTION

FREE-HAND sketch is a kind of vivid minimalism data formation, and easy to obtain, which has appeared before words/texts. **Anybody** can express his/her ideas by sketching even before he/she learned to speak and write. **Anything** can be depicted with the help of sketching if it is difficult to describe by words. **Anywhere** can allow sketching to assist communications, no matter what language is spoken there. Consequently, free-hand sketches are widely involved in human daily life from ancient times to date.

Free-hand sketch is illustrative, in spite of highly concise and abstract, which can be regarded as a form of human brain's visual perception of the real world. Drawing free-hand sketches is one of humans natural behavior, and sketches are useful in various scenarios, *e.g.*, communication, design. Therefore, free-hand sketch has been studied in computer vision field [1], [2], [3], [4], [5], [6], [7], [8] and computer human interaction community [9], [10], [11], [12]. In particular, early research can be traced back to the 1960s and 1970s [13], [14].

However, free-hand sketch is essentially different to natural photo¹. It stands as a special data modality/domain, having some domain-unique challenges, *e.g.*, highly sparse, abstract, noisy, arbitrary. Moreover, free-hand sketching is also a dynamic processing, so that free-hand sketch can be understood/represented in multiple formations/modalities, *e.g.*, static pixel space, stroke coordinate based temporal space, Euclidean space, geometrical graph space (see detailed discussion in Section 2). These intrinsic traits of free-hand sketch decide that solving sketch-related

problems needs to propose **sketch-specific** designs to overcome the sketch-specific challenges.

In recent years, the prevalence of smart touchscreen devices (*e.g.*, smart phone, tablet) has made sketch creation a much easier task than ever, and consequently made sketch-oriented applications increasingly more popular. This motivates both academia and industry to study free-hand sketches more extensively. Meanwhile, deep learning techniques are developing rapidly, and have achieved the state-of-the-art in various artificial intelligence tasks. Thus, an unprecedented boom of free-hand sketch research has emerged, evidenced by the increasing publications in the top venues (as illustrated in Figure 1): (i) Some classic research topics (*e.g.*, sketch recognition, sketch-based image retrieval, sketch-based 3D shape retrieval) have been re-studied in deep learning manner [3], [4], [5], [6], [15] and obtained significant performance improvements. (ii) Some brand new topics are proposed based on deep learning, *e.g.*, deep learning based sketch generation/synthesis [16], sketch-based deep model generation [17], deep reinforcement learning based sketch abstraction [18], deep sketch hashing [7]. (iii) Beyond the global representation based tasks (*e.g.*, sketch recognition), more fine-grained tasks have been further studied or proposed, *e.g.*, instance-level sketch-based image retrieval [5], deep stroke-level sketch segmentation [19]. (iv) Compared with representing sketches as static pictures in previous, there is a significant progress on the representations for free-hand sketches in recent years. Various deep neural network architectures have been designed for sketch. Especially, thanks to the groundbreaking work SketchRNN [16], the temporal traits of free-hand sketches are modeled by recurrent neural network (RNN). (v) More sketch-based applications have appeared, *i.e.*, on-line sketch game QuickDraw² [16], sketch-based commodity

• This work was done in BUPT before Peng Xu graduated.
Homepage: <http://www.pengxu.net>, GitHub: <https://github.com/PengBoXiangShang>

1. Image includes free-hand sketch, natural photo, *etc.* In this survey, natural photo denotes the image obtained by camera, *e.g.*, the photos from ImageNet. Unless specified instructions, photo refers to natural photo throughout this survey.

2. <https://quickdraw.withgoogle.com>

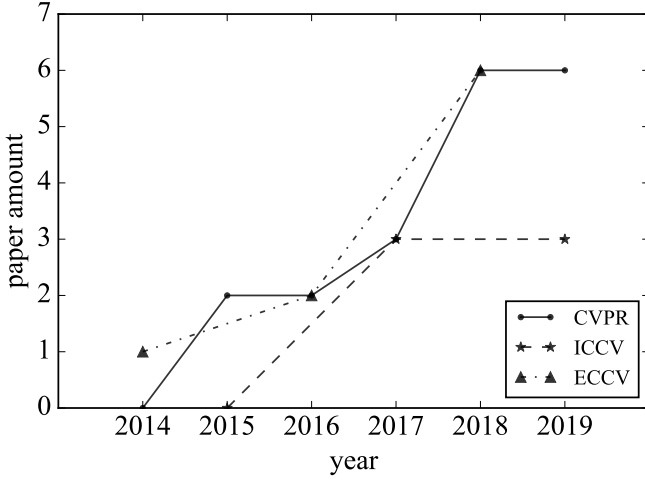


Fig. 1. Amount of the free-hand sketch papers on top conferences in computer vision area over the years.

search engine³ [5], [20]. (vi) Some large-scale sketch datasets have been collected, *e.g.*, Google QuickDraw⁴ [16] a million-scale sketch dataset (50M+).

The simultaneous prosperity of touchscreen devices and deep learning brings both opportunities and challenges to the free-hand sketch research community. In particular, there are still a lot of problems that are under-studied, although a great deal of deep learning models are proposed for free-hand sketches in recent years. This survey aims to review the state of the free-hand sketch community in deep learning era, hoping to bring insights to the related researchers. Please note that this survey focuses on free-hand sketches, not including professional (forensic) facial sketch [21], [22], [23], [24], professional pencil sketch [25], human/face sketch [26], professional landscape sketch [27], photo-like edge-maps (artificially rendered sketch) [28], [29], [30], cartoon/manga [31], [32], [33], well-drawn 3D sketch [34], clip arts [32]. In this survey, “sketch” refers to “free-hand sketch”, if without special instructions.

Different to the existing sketch-related survey papers [35], [36], [37] that focus on free-hand sketch based recognition and image/3D retrieval, this survey aims to introduce the state of free-hand sketch community in deep learning era by the detailed discussion and taxonomy for all the representative sketch-related research topics.

This survey are organized as follows. Section 2 provides some background knowledge of free-hand sketches, including intrinsic traits, domain-unique challenges, milestone techniques of the existing sketch-oriented deep learning, *etc.* Section 3 summarizes the representative free-hand sketch datasets. In Section 4, we provide a comprehensive taxonomy for sketch-based tasks, and the representative deep learning models will be described in detail. The bottlenecks and some open problems of the free-hand sketch community in deep learning era are discussed in Section 5 and Section 6, respectively. Section 7 gives some promising and

TABLE 1
Notation definitions in this survey.

Notations	Descriptions
\mathbf{M}, \mathbf{M}^T	matrix \mathbf{M} and its transpose
$\mathcal{X} = \{\mathbf{X}_n\}_{n=1}^N$	sketch sample set
\mathbf{X}_n	n -th sketch sample in the sketch sample set \mathcal{X}
$\mathcal{Y} = \{y_n\}_{n=1}^N$	associated label set of \mathcal{X}
y_n	label of \mathbf{X}_n
\cap	intersect two sets
\emptyset	empty set
$ \cdot $	the length of the set.
$\ \cdot\ _p$	p norm
\mathcal{L}	loss function
$\mathbf{W}, \mathbf{b}, \Theta$	learnable parameters of neural network
$\mathcal{F}(\cdot)$	function mapping or feature extraction
$\mathcal{F}_\Theta(\cdot)$	neural network feature extraction, parameterized by Θ
$\mathcal{D}(\cdot)$	distance metric, <i>e.g.</i> , ℓ_2 distance
λ	weighting factor
\sum	summation
\odot	element-wise product
$\sigma(\cdot)$	sigmoid activation function
$\text{ReLU}(\cdot)$	ReLU activation function
\mathbb{R}	set of real numbers
T	time step
t	time step
α, β, γ	hyper parameters by manually setting

interesting topics for future work. The conclusion of this survey is presented in Section 8.

Throughout this survey, bold uppercase characters and bold lowercase characters denote matrices and vectors, respectively. Unless specified instructions, the mathematical symbols follow the conventions in Table 1.

2 BACKGROUND

This section will present some background knowledge, including: (i) The intrinsic traits and domain-unique challenges of free-hand sketch. (ii) A brief development history of the deep learning techniques engineered for free-hand sketch. In particular, the essential differences between free-hand sketch and natural photo will be discussed, and the milestone deep learning techniques proposed for free-hand sketch will be introduced.

2.1 Intrinsic Traits and Domain-Unique Challenges

Free-hand sketch is a special modality in visual domain, essentially different to natural photos that are the pixel-perfect copies of the real world. For faster calculation, free-hand sketches are saved as black background picture in the computer, which is a highly sparse matrix (see Figure 2). Free-hand sketching is a dynamic processing. Hence, (i) as shown in Figure 2, sketches are not only the static 2D matrix in pixel space, but also can be represented as temporal vectors that consist of the stroke coordinate sequences. Both of the pixel space and stroke coordinate space belong to Euclidean space/gridding. From another perspective, free-hand sketching is also a processing of connecting the key points of strokes into lines. Thus, a sketch can be further extracted as a more sparse representation, *i.e.*, a sparsely connected graph that consists of the key stroke points, see Figure 2. Compared with the well-regulated Euclidean representations, graph representation in geometrical space is more flexible. Graph representation is the higher level

3. <http://sketchx.eecs.qmul.ac.uk/demos/>

4. <https://github.com/googlecreativelab/quickdraw-dataset>

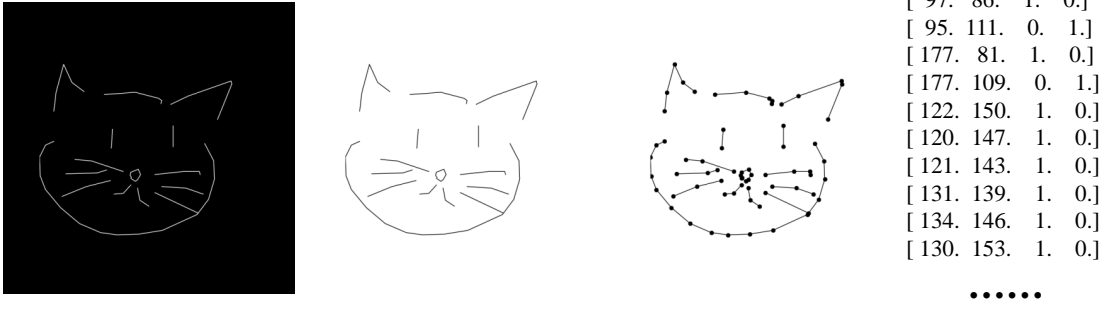


Fig. 2. Sketch-specific representations. Representations from left to right: picture (black background with white lines), picture (white background with black lines), graph, stroke sequence. For both graph and stroke sequence representations are based on the key stroke points. In stroke sequence, each key point is denoted as a four-bit vector, where the first two bits and the last two bits represent the coordinates and pen state, respectively. See details in text.

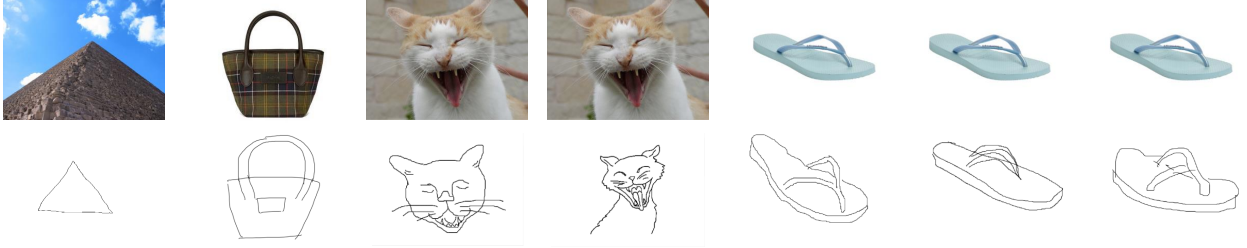


Fig. 3. Illustrations of the domain-unique challenges of free-hand sketches. Each column is a photo-sketch pair.

of abstraction for free-hand sketch. Consequently, in deep learning recipes, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Graph Neural Network (GNN) can be used to model and represent sketch. These are the major domain-specific traits of free-hand sketch, resulting in the domain-unique challenges.

The domain-unique challenges of free-hand sketch can be summarized as follows. (i) **Highly abstract**. Humans use sketch to depict a object or a event by as few strokes as possible, which are preprocessed by human brain. As shown in Figure 3, a pyramid can be depicted as a triangle in sketch, and several strokes depict a fancy handbag. (ii) **Highly diverse**. Different persons have different drawing styles, *e.g.*, exaggerated, artistic. Different persons have different drawing abilities. For instance, given a semantic concept of “cat”, persons draw really diverse cats with/without body (see Figure 3). In Figure 3, it can be observed that different persons draw distinctive sketches when given the identical slipper. (iii) **Highly noisy**. For one hand, free-hand sketches are always noisy with some redundant and noisy strokes. For another hand, in sketch domain, some semantically different object categories have high-level of visual similarity, *e.g.*, dog and pig. Sometimes, the noisy outlier of dog category looks more like a pig, due to the **intra-class variation/diversity and inter-class similarity**. Specifically, as demonstrated by Xu *et al.* [7], the image entropy values of each object category of sketches is a Gaussian distribution. (iv) **Highly sparse**. If represent free-hand sketch in pixel space, completely different to the pixel-dense natural photo, free-hand sketch is a kind of highly sparse signal. This makes representing sketch more challenging. If use CNN to model sketch, many unnecessary zeros will be involved

via convolutional operations.

Moreover, there are some unique challenges when collect sketch, which will be discussed in detail in following (see Section 3.1).

The aforementioned intrinsic traits and domain-unique challenges decides that **the free-hand sketch problems need the sketch-specific designs and thinkings!**

2.2 Development History in Deep Learning Era

In recent five years, free-hand sketch community has gone through a rapid development. Figure 4 presents the milestones of the free-hand sketch research community from 2015, mainly from three perspectives: deep learning oriented tasks, datasets, representations (network architectures). (i) In 2015, Sketch-a-Net [4] was proposed by SketchX lab. Sketch-a-Net is the first deep convolutional neural network engineered for free-hand sketch, and has achieved the accurate recognition rate that beats humans. This inspiring work demonstrates the availability of CNN for sketch analysis, and motivates researchers start to design deep learning models to solve sketch tasks. (ii) In 2016, three fine-grained sketch-based image retrieval (FG-SBIR) datasets are released, *i.e.*, SketchX Shoe [5], SketchX chair [5], Sketchy [15]. Combining with deep triplet ranking [38], these fine-grained cross-modal datasets have led a research boom of FG-SBIR, motivating various fine-grained sketch tasks. (iii) In 2017, Google company released a million-scale sketch dataset, *i.e.*, Google QuickDraw, via the online game “QuickDraw”. QuickDraw dataset is collected from the game users around the global world, containing more than 50M sketches. Thus, the data diversity of QuickDraw

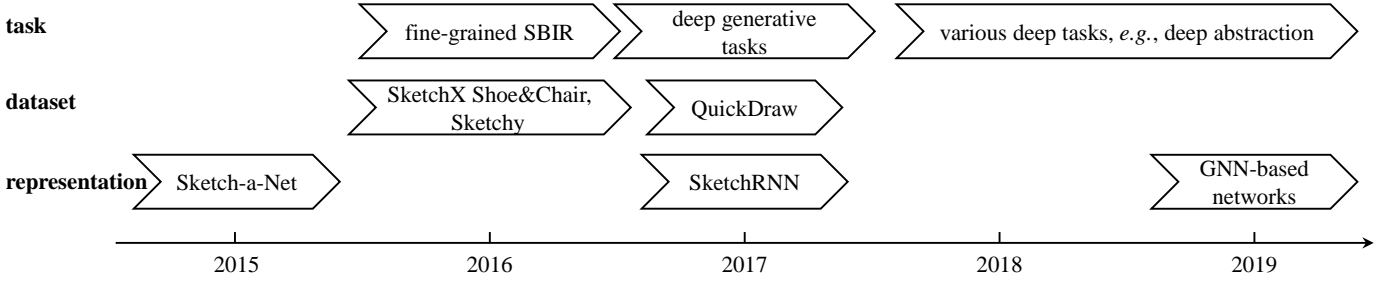


Fig. 4. Milestones of deep learning based free-hand sketch research in recent years, from the perspectives of task, dataset, and representation. Various representations for free-hand sketches are based on different deep neural network architectures, *i.e.*, CNN, RNN, hybrid of CNN and RNN, GNN.

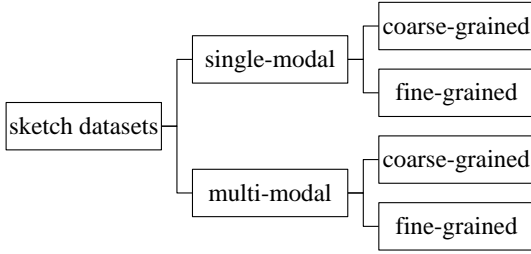


Fig. 5. A tree diagram of the sketch dataset taxonomy.

dataset is extremely rich. Furthermore, based on QuickDraw dataset, Ha *et al.* propose “SketchRNN”, a RNN-based deep Variational AutoEncoder (VAE) model that can generate diverse sketches. Before 2017, all the previous deep learning works regard sketch as static picture, so that only use CNN to represent sketch. SketchRNN opens a new door to the sketch community, showing the amazing idea that use the coordinate of key stroke points as input, by grasp the temporal traits of sketch stroke. This seminal idea motivates the sketch researchers to analysis sketch in temporal space, beyond the static 2D space, considering more intrinsic traits of sketching processing. In 2017, with the prosperity of Generative Adversarial Networks (GAN), some sketch-based deep generative works [39] gradually began to be presented in the top conferences in computer vision area. (iv) From 2018 to date, based on deep learning techniques, various novel tasks have been proposed, *e.g.*, deep sketch hashing [7], deep sketch abstraction [18], deep sketch-based photo classifier generation [17], deep zero-shot sketch-photo hashing [40], and some classical tasks are re-studied, *e.g.*, deep sketch perceptual grouping [41]. See Figure 4 for more detailed chronological orders.

3 FREE-HAND SKETCH DATASETS

This section will discuss the unique-challenges of free-hand sketch collection, and introduce the recently representative deep learning oriented datasets by a detailed taxonomy.

As shown in Figure 5, according to the data modalities involved, free-hand sketch based datasets can be categorized as two kinds: (i) single-modal and (ii) multi-modal datasets. Single-modal sketch datasets only consist of sketches, which are mainly used for sketch recognition, retrieval/fast retrieval, perceptual grouping, segmentation,

generation, *etc.* Multi-modal sketch datasets consist of both sketches and the samples from other modalities (*e.g.*, natural photo, 3D shape, text, cartoon, video), which are mainly used for the retrieval/matching, generation across sketch and other modalities. Moreover, both single-modal and multi-modal sketch datasets can be divided in two kinds: (a) coarse-grained and (b) fine-grained.

The coarse-grained single-modal sketch datasets (*e.g.*, TU-Berlin [1], QuickDraw [16]) are usually used for sketch recognition, sketch retrieval. The fine-grained single-modal sketch datasets [19], [41] have fine-grained annotations, which can serve for the fine-grained sketch tasks, *e.g.*, perceptual grouping, segmentation, parse.

The fine-grained multi-modal sketch datasets (*e.g.*, SketchX Shoe [5]) provide the instance-level pairing information, usually with fine-grained annotations, *i.e.*, local attributes. Hence, the fine-grained multi-modal sketch datasets can be used for fine-grained sketch based cross-modal retrieval and generation tasks.

Existing collection approaches mainly include: (i) crowd-sourcing selecting and matching on existing datasets, *e.g.*, Doodle2Sketch QuickDraw-Extended [8], (ii) crowd-sourcing drawing from scratch, *e.g.*, SketchX Shoe [5], Sketchy [15]. (iii) collecting via online drawing game, *e.g.*, Google QuickDraw [16]. In particular, for fine-grained multi-modal sketch datasets, crowd-sourcing drawing is more suitable, since fine-grained selecting and matching takes more time.

In addition, some novel concepts and tasks on learning fine-grained sketch patterns are widely studied in recent years, leading to some new sketch datasets, *e.g.*, SketchX Shoe [5], Sketchy [15], QuickDraw [16]. This section will brief the most representative datasets in current stage, which are also the most frequently-used datasets. Some well-known but overly small-scale datasets will not be discussed in this section, *e.g.*, the dataset proposed by Hirata and Kato [54].

3.1 Unique-Challenge of Sketch Collection

Free-hand sketch stands as a special data domain, its collection has some unique-challenges: (i) Sketching is a dynamic and temporal processing, thus saving as the static picture (*e.g.*, JPEG, PNG) will loss too much useful information. In order to enable more extensive research, it is better to record the stroke position and timing information. This is to

TABLE 2

Summary for the representative sketch datasets. “✓” denotes “yes/available/provided”. Both of “grouping” and “segmentation” annotations are in stroke-level. “K” and “M” means “thousand” and “million”, respectively. “cat.” means “category”. The stroke column will be marked as available if the sketches are provided in SVG files or coordinate arrays.

single-modal datasets	public	modalities & sample amount	cat.	stroke	object/scene	instance pairing	annotations	remarks
TU-Berlin [1]	✓	20K sketches	250	✓	o	-	class	
QuickDraw [16]	✓	50M+ sketches	345	✓	o	-	class	
QuickDraw-5-step [42]		38M+ sketches	345		o	-	class	
Li <i>et al.</i> SPG [41]	✓	20K sketches	25	✓	o	-	class, grouping	
SketchSeg-150K [19]		150K sketches	20	✓	o	-	class, segmentation	57 semantic labels
SketchSeg-10K [43]	✓	10K sketches	10		o	-	class, segmentation	24 semantic labels
SketchFix-160 [44]	✓	3904 sketches	160	✓	o	-	class, eye fixation	
multi-modal datasets	public	modalities & sample amount	cat.	stroke	object/scene	instance pairing	annotations	remarks
SketchX Shoe [5]	✓	419 sketches, 419 photos	1		o	✓	pairing, triplet, attribute	21 binary attributes
SketchX Shoe V2	✓	6648 sketches, 2K photos	1	✓	o	✓	pairing	
SketchX Chair [5]	✓	297 sketches, 297 photos	1		o	✓	pairing, triplet, attribute	15 binary attributes
SketchX Handbag [20]	✓	568 sketches, 568 photos	1		o	✓	pairing	
Sketchy [15]	✓	75K sketches, 12K photos	125	✓	o		class	12K objects
sketch&UI [9]		1998 sketches, 1998 photos	23		o	✓	class, pairing	UI
QuickDrawExtended [8]	✓	330K sketches, 204K photos	110		o		class	
TU-Berlin Extended [45]		20K sketches, 191K photos	250		o		class	
sketch Flickr15K [2]	✓	330 sketches, 15K photos	33		o		class	
Aerial-SI [46], [47]		400 sketches, 3.3K photos	10		o, s		class	aerial scene
HUST-SI [48]	✓	20K sketches, 31K photos	250	✓	o		class	
SBSR [49]	✓	1814 sketches, 1814 3D models	161		o		class	
SHREC’13 [35]	✓	7200 sketches, 1258 3D models	90		o		class	
SHREC’14 [50]	✓	12680 sketches, 8987 3D models	171		o		class	
PACS DG [51]	✓	9991 (sketches, photos, cartoons, paintings)	7		o		class	domain generalization
Flickr1M [52]		500 sketches, 1.3M photos	100		o		class	
Cross-Modal Places [53]	✓	16K sketches, 11K descriptions, 458K spatial texts, 12K clip arts, 1.5M photos	205		s		class	

say that generally sketch collection needs specific designed interactive interface to record the sketch domain-unique information. In some cases, for further analysis, it also needs to record some extra information of the drawers, *e.g.*, gender, nationality. More complicated than natural photo collection (crawling and saving as picture directly), sketch collection generally needs to save sketch as Scalable Vector Graphics (SVG) ⁵ and record other additional information, *e.g.*, stroke position/timing information. (ii) Although, with popularization of smart touch-screen devices, more and more free-hand sketches are available in the internet, it is still difficult to collection sketches via crawling. The main reasons are: (a) The ready-made sketches in the internet are saved as static pictures, without stroke temporal and timing information, so that only support to model sketch statically. (b) The quality of sketch is exceedingly dependent on the status of collection device, *e.g.*, screen resolution, predefined stroke thickness. Moreover, some sketches in the internet are too messy to recognize even by humans. Thus, it is difficult to obtain sketches in the acceptable drawing quality. This is distinctively different to photo collection, which is mainly based on crawling in the internet. (c) When collect fine-grained cross-modal datasets, it needs to collect the fine-grained/instance-level cross-modal sample pairs. In this case, selecting and then matching the cross-modal sample pairs on the fine-grained criterion is time-consuming, while drawing sketches referring to the

counterpart in other modality could be more efficient. In addition, it often happens that collectors can not search for a adequately well-matched sketches for the target sample from other modality. (iii) As aforementioned, sketch is full of the diversity of drawing styles. Thus, in ideal conditions, any sketch dataset should involve as many drawing participants and styles as possible. It is better to ensure that the drawing participants vary in as many ages and regions as possible. However, during crawling, we can not know any extra information about the sketchers in the internet. (iv) There are two extreme cases that always happen: (a) The crowd-sourcing participants often draw deliberately to obtain the well-drawn sketches. (b) The online game participants often draw overly arbitrarily resulting in the highly noisy sketches.

3.2 Representative Dataset Taxonomy

In the last decade, the prosperity of both deep learning and touch-screen based human-computer interaction, motivates the fast development of free-hand sketch research. Some brand new topics related to sketch have been proposed. New topics often need new datasets. Moreover, nowadays, almost all the sketch topics are studied in deep learning techniques, which need large-scale datasets. Therefore, in recent ten years, a number of sketch datasets [55], [56], [57], [58], [59] are proposed. This part will introduce and compare the recently representative sketch datasets via a detailed taxonomy, and describe what tasks/topics each dataset can be used for. Please see details in Table 2.

5. https://en.wikipedia.org/wiki/Scalable_Vector_Graphics

Table 2 summarizes the representative sketch datasets of both single-modal and multi-modal, where the comparison is based on the aspects of modalities, sample amount, sample category amount, stroke information, annotations, *etc.* Note that SVG file is able to generate picture files, *e.g.*, JPEG, PNG (not vice versa).

4 TASK AND METHODOLOGY TAXONOMY

According to the data modalities involved, free-hand sketch related tasks can be divided into single-modal and multi-modal tasks. The single-modal sketch research serves as the foundation for the multi-modal research. This section will detail the definitions of the existing tasks and introduce the correspondingly representative deep learning methods, providing a detailed taxonomy. Figure 6 provides a tree diagram of the existing free-hand sketch tasks.

4.1 Single-Modal

Single-modal sketch tasks need only to study sketch without other data modalities involved. In recent years, various single-modal sketch tasks are studied in deep learning framework, including recognition, retrieval/hashing, generation, grouping, segmentation, abstraction, *etc.* These interesting tasks need the machine to understand sketches from global to local.

4.1.1 Recognition

Sketch recognition [1] is to predict the class label for the given sketch, which is one of the most fundamental tasks in free-hand sketch community, with a lot of practical applications, *e.g.*, interactive drawing system that provides some feedback to users [60], sketch-based science education system [61]. Both of object sketches [1], [16] and scene sketches [62], [63] have been studied to date. The data augmentations, network architectures, loss functions that are engineered for sketch recognition can be used for other sketch tasks. Moreover, the deep learning based sketch recognition techniques have been already widely used in some sketch-based human-computer interaction applications, *e.g.*, online drawing game QuickDraw, WeChat mini-app Caihua Xiaoge that was released by Google.

Sketch recognition can be categorised as (i) offline recognition and (ii) online recognition.

Offline recognition systems take the whole sketch as input and predict a class label, after the sketching was done. Online recognition systems take the accumulated sketch strokes and continuously predict the class label, during sketching. Most of the existing deep learning based sketch recognition works are offline recognition models. Online sketch recognition can be used for real-time drawing guidance [42], tracing, and sketch retrieval, providing better interactive experience.

According to the sketching content, sketch recognition can also be categorised as object recognition and scene recognition/understanding.

Comments: With the development of deep learning, the understanding of researchers on sketch is gradually deepening, the main progressive trends including: (i) from CNN-based static representation to RNN-based dynamic

representation, (ii) from explicit picture space to implicit sequence space; (iii) from global representation to local understanding (*i.e.*, semantic spatial attention [20]), from Euclidean space (CNN, RNN based) to geometric space (GNN based), (iv) from conventional supervised training to adversarial training.

Benefit from the booming progress of deep learning, various deep models [45], [64], [65], [66], [67], [68], [69], [70], [71] are proposed for free-hand sketch recognition. In the following, the deep learning based sketch recognition works will be introduced from three perspectives, *i.e.*, (i) network architectures, (ii) data augmentations, (iii) loss functions.

Network Architecture:

Figure 7 demonstrates the evolution of the existing deep learning-based sketch representations. Moreover, Table 3 summarized the existing network architectures that are engineered for free-hand sketches and capable of sketch recognition. The following will introduce some representative architectures.

(i) Sketch-a-Net [4], [6] is the first deep CNN designed for free-hand sketch. Compared with classic photo-oriented CNN architecture [78], the sketch-specific aspects of its architecture mainly include: (a) Considered the highly sparsity of sketch, larger size of 15×15 is used for the first layer filters. This is because sketches generally lack texture information. Larger convolutional filters are more likely to capture more patterns from the structured/sparse context. (b) Local response normalization (LRN) [78] layers are removed for faster learning without sacrificing performance, since LRN works for “pixel brightness normalization”. The brightness of sketches is 0 or 1. Moreover, Yu *et al.* also proposed two novel sketch-specific data augmentation strategies to synthesises new sketches to enable Sketch-a-Net to learn the stroke ordering information. Given an ensemble of Sketch-a-Net trained by varying the mini-batch orders, joint Bayesian fusion [79] is adopted to improve the recognition performance.

(ii) Sarvadevabhatla *et al.* [72] also propose a sketch recognition network to explore the sequential process of sketching, where each training sketch needs to be plotted as a continuous sequence of cumulative stroke pictures and the corresponding AlexNet [78] based deep features will be sent into Gated Recurrent Unit (GRU) [80] network in sequence. This network is able to work in online recognition mode, since it involves the intermediate status of the sketch.

(iii) Similarly, He *et al.* [74] proposed the Deep Visual-Sequential Fusion (DVSF) net to capture spatial and temporal patterns of sketches simultaneously. For each training sketches, its three accumulation sub-pictures (with 60%, 80%, 100% of strokes) go through three-way CNNs (ResNet-18 [81]) to produce deep features, which are parallelly fed into the visual and sequential networks. In particular, the visual and sequential networks are implemented by residual fully-connected (R-FC) layer and Residual Long Short Term Memory (R-LSTM) [82] layer, respectively. The visual and sequential representations are integrated into the fusion layer to conduct sketch recognition.

(iv) In 2017, Ha and Eck proposed a groundbreaking work, *i.e.*, SketchRNN [16], which contributes a novel paradigm to model the sketching ordering patterns. SketchRNN brings significant insights to the sketch repre-

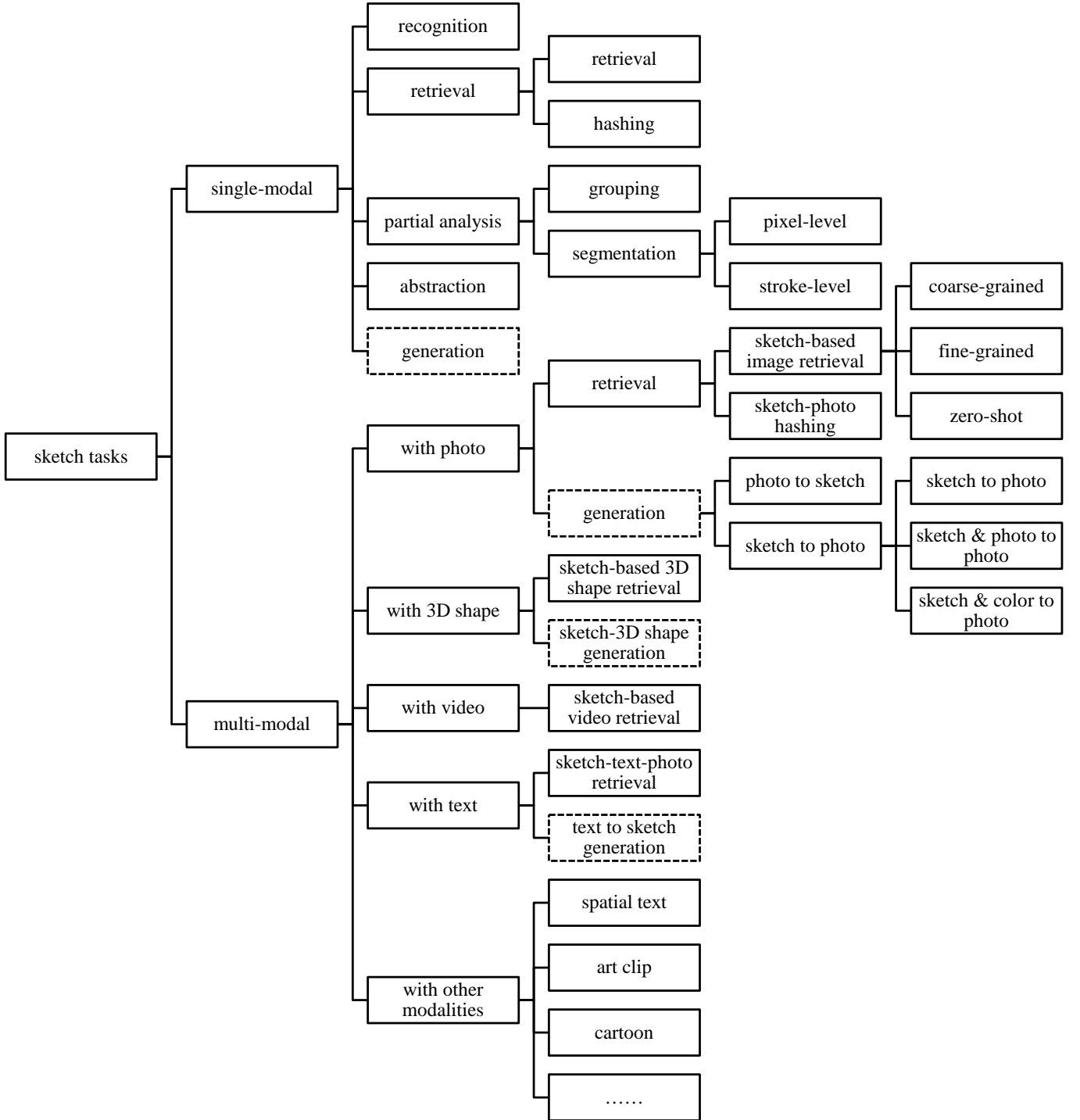


Fig. 6. A tree diagram of the sketch task taxonomy. Generative tasks are framed by dashed lines.

sent learning, although it is a Variational Inference (VI) [83] based sketch generation model. Distinctively different to the stroke accumulated sub-picture based RNN inputs, RNN backbone of SketchRNN is directly fed into the key point coordinates of sketching stroke. In particular, as illustrated in Figure 2, each key point is denoted as a vector consisting of two coordinate bits (*i.e.*, horizontal and vertical coordinates) and the corresponding flag bits. Flag bits of the key stroke points depict the pen states during sketching, telling

when to stop a stroke. Specifically, within each stroke, its ending point will have a distinctive flag bit representation to those of other key points.

(v) Xu *et al.* proposed a sketch hashing network termed as SketchMate [7], where the backbone is a CNN-RNN dual branch architecture, utilizing CNN to extract abstract visual concepts and RNN to model human sketching temporal orderings. The CNN branch takes in the raster pixel sketch pictures; and the RNN branch takes in the vector sketch (*i.e.*,

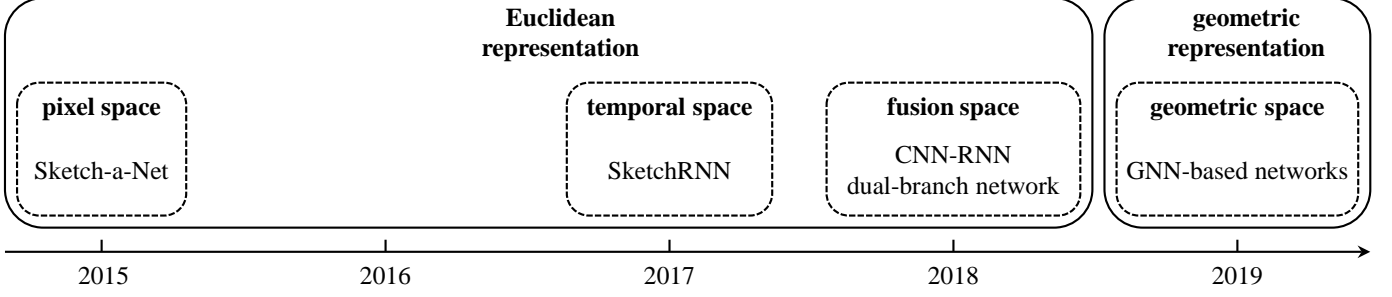


Fig. 7. Evolution of the existing deep learning-based sketch representations. Various modeling spaces are separated by dotted lines. The representative network architectures are provided.

TABLE 3

Comparison for the representative sketch-oriented network architectures. “—” means that it is vague or not mentioned in the original paper. The reported performances are the recognition accuracy of “acc.@top1”. Abbreviations in this table are defined as: “para.”: parameters; “ens.”: ensemble; “stroke accu. pic.”: stroke accumulated pictures; “R-FC”: residual fully-connected layer; “cat.”: category; “pad.”: padding; “tru.”: truncation; “augm.”: specific augmentations; “tran.”: transformer.

Year	Model	Architecture	Layers	para.	ens.	Pretrain	Input	Preprocess	Dataset	Performance
2015	Sketch-a-Net [4]	CNN	5 conv.	8.5M	✓		picture	augm. [4]	TU-Berlin [1] 250 cat.	74.90%
2016	Alexnet-FC-GRU [72]	CNN-to-RNN cascaded	—	—			stroke accu. pic.	—	TU-Berlin 160 cat.	85.10%
2018	SketchMate [7]	RNN	2 GRU	—			stroke vector	tru. & pad. [7]	QuickDraw 3.8M [7]	77.88%
2018	SketchMate [7]	CNN-RNN dual-branch	5 conv. & 2 GRU	—			picture & stroke vector	tru. & pad. [7]	QuickDraw 3.8M [7]	79.49%
2017	Jia <i>et al.</i> [73]	RNN-RNN dual branch	—	—	✓	CNN needs pretraining	CNN features of stroke accu. pic.	reflection, rotation, etc.	TU-Berlin	92.20%
2017	DVSF [74]	R-FC and RNN dual branch	—	—		CNN needs pretraining	CNN features of stroke accu. pic.	—	TU-Berlin	79.60%
2018	FBin DAB-Net [75]	binary CNN	—	—			picture	—	TU-Berlin	73.70%
2018	RNN→CNN [76]	RNN-to-CNN cascaded	2 LSTM & 5 conv.	—		CNN needs pretraining	stroke vector	augm. [6]	TU-Berlin	78.49%
2019	multi-graph tran. [77]	GNN	4 tran.	10M			stroke vector	—	QuickDraw subset [77]	70.70%

key stroke point coordinates), respectively. The dual-branch interaction is implemented by a concatenation-based late-fusion layer. This architecture demonstrates the complementarity of visual and temporal embedding spaces on sketch representation learning. This CNN-RNN dual-branch modeling idea has been widely applied to other sketch tasks, *e.g.*, SPFusionNet [43] for sketch semantic segmentation. Addition to the parallel architectures of CNN and RNN, some cascaded architectures (*e.g.*, CNN-to-RNN [72], RNN-to-CNN [76]) also have been studied.

Sketch Data Augmentation:

Please note that all the sketch-specific data augmentation methods discussed in this subsection can be applied to both sketch recognition and all the other sketch involved tasks, *e.g.*, sketch-based image retrieval, sketch-related generation.

(i) Sketch can be formatted in visual domain as picture, so that most of the common data augmentations designed for natural photos (*e.g.*, ImageNet [84] photos) can be applied to sketch, *e.g.*, horizontal reflection/mirroring, rotation, horizontal shift, vertical shift, central zoom. These augmentations have already been evaluated by the early sketch-oriented deep learning works [4], [85]. However, the random cropping is unsuitable to sketch due to that partial sketches are too sparse and abstract to recognize even for humans.

(ii) Stroke thickening/dilation is another kind of augmentation for free-hand sketch. As discussed in some pre-

vious works on the spatially-sparse convolutional neural networks [86], the subtle details of sparse strokes of sketches will be lost during the layer-by-layer convolutions. Thus, the stroke thickening/dilation is useful for the neural networks to learn the details of sketches.

(iii) Yu *et al.* [6] propose to remove the strokes to obtain more diverse sketches, which rely on the temporal order information of human sketching. Note that this stroke remove is not a random remove. Its main idea comes from a general rule [1] that during sketching human like draw main outline first and then draw details. Thus, Yu *et al.* progressively remove detail from each training sketch to generate new sketches at multiple abstraction levels.

(iv) Zheng *et al.* [87] propose a Bezier pivot based deformation (BPD) strategy and a mean stroke reconstruction (MSR) approach. Both of these two augmentations do not need any temporal cues of sketches. The main idea of MSR is to generate novel sketches with smaller intra-class variance.

(v) Some sketch-related generation works [88] also propose some sketch data augmentation strategies. Liu *et al.* propose two sketch-specific data augmentation strategies: (a) Manually extract some strokes from sketch SVG files to construct the noise stroke masks. Then, randomly apply the noise stroke masks to the original sketches to synthesize augmented sketches. (b) Randomly extract a patch from another sketch, and attach it upon a given sketch.

Comments: Compared with augmentations on the full picture basis (e.g., rotation, shift), making full use of the stroke information to design sketch-specific data augmentations is a significant progress. This benefits from that researchers gradually analyse sketches from global to local details. However, all the aforementioned stroke modification based augmentations are merely random perturbations, without the considering how to mimic the real drawing styles of humans.

Loss Function:

Most of the previous deep sketch recognition methods use cross-entropy softmax loss to train the deep neural networks. However, as stated in Section 2.1, free-hand sketch has its intrinsic traits and domain-unique challenges, so that it's better to study the sketch-specific loss function to drive the recognition models to learn more fine-grained sketch patterns. Xu *et al.* proposed the sketch-specific center loss [7] for million-scale sketches, based on the staged-training strategy. The basis is that the image entropy distribution of each sketch category is a truncated Gaussian distribution.

Inspired by the classical Bayesian decision theory [89] and Bayesian risk [90], Mishra *et al.* proposed a deep metric loss to drive the pretrained deep neural network to minimize the Bayesian risk $\mathcal{R}(\mathcal{F}_\Theta)$ of misclassifying the sketch pairs that were randomly selected within each mini-batch. Assume $\mathcal{F}_\Theta(\mathbf{X}_i)$ and $\mathcal{F}_\Theta(\mathbf{X}_j)$ are the normalized feature embeddings of the random selected sketches \mathbf{X}_i and \mathbf{X}_j . c^+ and c^- denote all the possible positive and negative class pairs that could be predicted, respectively.

$$\mathcal{R}(\mathcal{F}_\Theta) = \int_{-1}^{-1} \int_{-1}^{z=\mathcal{D}(\mathcal{F}_\Theta(\mathbf{X}_i), \mathcal{F}_\Theta(\mathbf{X}_j))} P(c^+ | \mathcal{D}(\mathcal{F}_\Theta(\mathbf{X}_i), \mathcal{F}_\Theta(\mathbf{X}_j))) P(c^- | \mathcal{D}(\mathcal{F}_\Theta(\mathbf{X}_i), \mathcal{F}_\Theta(\mathbf{X}_j))) d^2 z, \quad (1)$$

where $\mathcal{D}(\cdot, \cdot) (\in [-1, 1])$ denotes cosine distance function. This risk is the probability integral of classifying positive sketches as negative and negative sketches as positive. Based on this Bayesian risk loss, sketch recognition needs a two-stage training. After obtained the features, it needs to train a linear SVM [91] as the classifier.

Comments: The practical human-computer interaction motivates a new trend of sketch online recognition. Moreover, the previous deep learning models for sketch recognition work in the supervised manner, leaving the semi-supervised and unsupervised method under-studied. Considering from the practical application scenarios, zero-shot [63] and few-shot sketch recognition methods are also worth studying in the future. Another interesting topic that proposed by recent works [70], [92] is using multi-task learning framework to simultaneously solve sketch recognition with other tasks.

4.1.2 Retrieval and Hashing

Sketch retrieval [42], [93], [94] means that use sketch as query/target to retrieve the similar samples from sketch gallery. This is also a fundamental task for sketch understanding, with several practical applications. With the popularization of touch-screen devices, more and more sketches are spreading in the internet, activating the study of sketch fast retrieval. Sketch hashing [7] is a kind of sketch fast retrieval that all the sketches are encoded and searched as binary hashing code vectors.

Sketch retrieval is a challenging task due to the highly abstract and intra-class variations, so that it's difficult to explicitly localizing the feature points across sketches (e.g., the manner of SIFT [95]). After entering the deep learning era, the end-to-end feature learning manners of CNNs outperform the shallow feature extractions on various computer vision tasks. CNNs have also been used for sketch retrieval.

In particular, the common practice is using CNNs to learn embedding representations and then conduct the retrieval/matching as k Nearest Neighbors (kNN) problem. Therefore, most existing deep sketch retrieval models work as metric learning manners, which focus on the loss function designs to learn metrics for sketch matching in CNN-based embedding spaces. Wang *et al.* [93] proposed a representative sketch retrieval framework, which has two key components, i.e., pure convolutional layer based Siamese CNN backbone, ℓ_1 norm distance based pair-wise loss. The idea is two-fold: (i) Use convolutional feature map to preserve the spatial information for sketches without point correspondence. (ii) Pair-wise loss aims to drive the similar pairs closer in the feature space.

In addition to low-level feature based spatial matching, higher level semantic features also work well on sketch retrieval. Xu *et al.* propose a dual-branch CNN-RNN architecture to represent sketches via dual-branch late-fusion.

The aforementioned sketch retrieval/hashing methods work in supervised learning manner. If the class labels are unavailable, adversarial training can be used to force the networks learn how to capture the features for sketches. Based on Generative Adversarial Network (GAN) [96] framework, Creswell *et al.* [94] propose the Sketch-GAN to solve sketch retrieval in unsupervised manner, where both the query and gallery sketches are represented by the output features of the discriminator network.

4.1.3 Generation

Sketch generation is a booming topic in deep learning era, which has several practical application scenarios, e.g., synthesising novel pictures, assisting artist design, finishing incomplete sketches. Sketch generation can be solved in various frameworks, e.g., variational autoencoder (VAE) [16], [59], [97], Generative Adversarial Network (GAN) [98], VAE-GAN [98], reinforcement learning (RL) [98], [99]. OK

The seminal model SketchRNN [16] is a sequence-to-sequence VAE model for conditional and unconditional sketch generation in the vector format. Its encoder and decoder are implemented by bidirectional RNN [100] and unidirectional RNN, respectively. Specifically, as stated earlier, free-hand sketching is a dynamic process of connecting key points into lines, so that the main idea of SketchRNN is to simulate the human sketching in a point-by-point manner. This is to say that SketchRNN produces sketch strokes in a point-by-point manner, and provides both the coordinates and pen state for each point. Pen state enables machine to know when to stop a stroke.

As shown in Figure 8, the VAE encoder of SketchRNN takes vector format sketches as input, and encodes each sketch as a hidden vector \mathbf{h} , which is the RNN hidden state of the last time step. This hidden vector will be further encoded as two parameters μ and σ to model a Gaussian distribution $N(\mu, \sigma)$, where a latent vector \mathbf{z} will be sampled

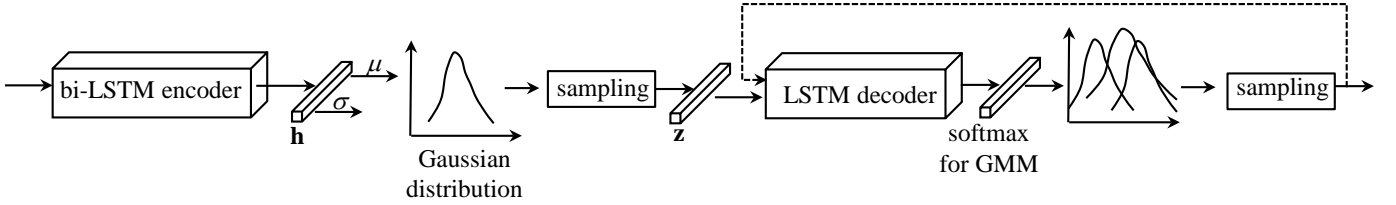


Fig. 8. Architecture of SketchRNN model. The dotted arrow line denotes the recurrent processing of LSTM decoder. For simplicity, the recurrent processing of bi-LSTM encoder is not shown here.

out. Then, the LSTM based VAE decoder will generate the pen coordinates and pen states of the key stroke points, conditionally on z . In particular, the pen coordinates and state for each key stroke point is sampled from a Gaussian mixture model, also as the input for the next time step for decoder.

SketchRNN is able to work in conditional and unconditional modes, corresponding to two practical scenarios: (i) Conditional Reconstruction: Neural network follows the human sketching styles to imitate the inputted sketches. (ii) Unconditional Generation: Neural network draws a sketch of the given class label, without any references. In unconditional generation scenario, it only needs to train the decoder of VEA, without any input or latent vectors. The unconditional generation has various practical applications, *e.g.*, helping human to design novel graphic patterns.

To improve SketchRNN to deal with multi-class generation, Cao *et al.* [101] propose a generative model named as “AI-Sketcher”, which is also a VAE based network.

There are several development trends of sketch generation: (i) Fine-grained sketching [59] is proposed by researchers. (ii) Recently, a novel evaluation metric termed as “Ske-score” is proposed in [98], which quantifies the goodness of generated vector sketches.

4.1.4 Grouping, Segmentation, and Parse

Compared with sketch recognition, retrieval, and generation, there are several more fine-grained single-modal sketch understanding tasks, *e.g.*, perceptual grouping, segmentation, parse. These fine-grained tasks need the corresponding models are able to analyse semantically meaningful sketch parts on local-level, *i.e.*, stroke-level. Moreover, these local-level understanding techniques also benefit other global-level tasks, *i.e.*, sketch-based image retrieval, sketch-based video retrieval [102], sketch generation/synthesis. In the following, we will overview the recent advance of deep learning on sketch perceptual grouping, segmentation, and parse.

Sketch Perceptual Grouping (SPG):

Humans have the ability to perceptually group visual cues into semantically object parts/components, which has been widely researched in Gestalt psychology area [103], [104]. As shown in Figure 9, humans are able to perceptually group sketches into semantic parts, *e.g.*, airplane grouped into fuselage and wings. Thus, sketch perceptual grouping (SPG) is to imitate human intuition to perceptually group sketch strokes into semantic parts, which has already been studied before deep learning era [105], [106], [107]. One representative application of SPG is to simplify sketches [108].

Moreover, SPG can also be used for sketch recognition [109], sketch semantic segmentation, synthesis [110], retrieval, fine-grained sketch-based image retrieval (FG-SBIR), sketch-based video retrieval [102], *etc.*

Li *et al.* [41], [111] contribute the largest SPG dataset to date of 20,000 manually-annotated sketches across 25 object categories, and propose a universal deep grouper that can be applied to sketches of any category in any domain. Specifically, this deep universal grouper is also a sequence-to-sequence variational autoencoder with both generative and discriminative objectives: (i) Its generative loss enables the grouper have the ability to handle unseen object categories and datasets. (ii) Its discriminative loss consists of a local grouping loss and a novel global grouping loss, to guarantee both local and global perceptual grouping consistency.

Sketch Semantic Segmentation (SSS):

Sketch semantic segmentation has drawn a lot of attention [105], [112], [113] in free-hand sketch community as a classic topic before the eruption of deep learning. The goal of sketch semantic segmentation is to predict the semantic labels for sketch strokes. The major difference between sketch semantic segmentation and sketch grouping is that the semantic segmentation results explicitly provide the part labels (*i.e.*, the concrete semantic category names) for each stroke, while the grouping results only provide the aggregation relationships for the strokes. As argued in [41], the similarity between sketch perceptual grouping and sketch semantic segmentation is analogous to that between unsupervised image segmentation [114] and semantic segmentation [115].

According to the network architectures, the existing deep learning models for sketch semantic segmentation can be mainly divided into several manners, CNN, RNN, CNN-RNN dual-branch, VAE based models, *etc.* The representative models mainly contain Stroke-RNN [116], SketchSegNet [117], SketchSegNet+ [19], SPFusionNet [43], *etc.*

Li *et al.* [118] train a CNN-based network to transfer the well-annotated segmentations and labellings from 3D dataset to sketch domain. They use annotated 3D data [112], [119] to produce edge-map with partial annotations as the fake sketch to train the segmentation networks. Qi *et al.* propose SketchSegNet [117] and SketchSegNet+ [19]. SketchSegNet+ [19] considers the sketch stroke orderings and is able to work over multiple object categories. In particular, SketchSegNet and SketchSegNet+ work in RNN-based VEA framework, where the Gaussian mixture model (GMM) layers of SketchRNN are replaced with the fully-connected layers combining with softmax layer to predict

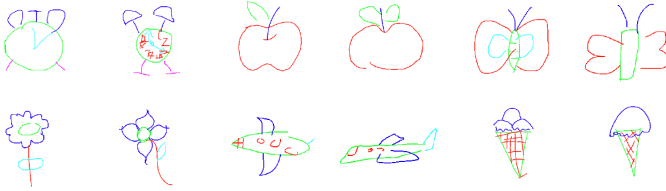


Fig. 9. Sketch samples of SPG dataset [111] (alarm clock, apple, butterfly, flower, airplane, ice cream). Semantically meaningful stroke groups are annotated by colors. Best viewed in color.

the part labels. The architecture of Stroke-RNN [116] equals to the encoder of SketchRNN. SPFusionNet [43] uses the late fusion of CNN-RNN dual-branch to represent sketch to conduct segmentation training.

Comments: Essentially, sketch perceptual grouping (SPG) is a stroke-level clustering, however, sketch semantic segmentation (SSS) is a explicitly semantic based stroke-level classification. Considering the supervision during training, SSS needs more stronger supervision than SPG. Specifically, during training, SSS system needs to know not only which strokes are connected together to denote a semantic part but also the meaningful name of this part.

Comments: Even though sketch semantic segmentation can be modeled by CNNs. However, CNNs lose the stroke orderings and temporal patterns of sketch, learning sketch representations on the patch basis. This CNN-based representation makes SSS fall into the same workflow as that of natural photo semantic segmentation.

Comments: In some cases, the non-deep grouping methods mainly rely on thresholding low-level geometric properties among the strokes. This often results in that strokes with equal geometric properties but different semantics will be grouped. Thanks to RNN-based representation, deep SPG methods consider more high-level semantic and temporal information, from the perspective of sketching.

Sketch Parse:

In recent years, a new deep learning concept “sketch parse” [58], [120], [121], [122] appears in free-hand sketch community. As a kind of fine-grained semantic understanding of sketch, sketch parse has already been applied to assist other sketch tasks [58], *e.g.*, sketch-based image retrieval (SBIR). Sketch parse is a special case of sketch semantic segmentation. In particular, as shown in Figure 10, completely different to the stroke-level parse of sketch semantic segmentation models, sketch parse models work in the natural photo semantic segmentation manner, which parse sketches in area-level/part-level/patch-level ignoring the stroke traits of sketches thoroughly. The existing sketch parse models use only CNN-base architecture to represent sketch, *e.g.*, SFSegNet [120] using Deep Fully Convolutional Networks (FCN) [123].

4.1.5 Simplification, and Abstraction

Before deep learning era, sketch simplification has been widely studied [108], [124], [125] by computer graphic researchers to obtain the sparse yet discriminative representation, which aims to determine which strokes of a given scribbled sketch should be **merged** [125] by some artificial principles, *e.g.*, Gestalt principle. A typical pipeline is the

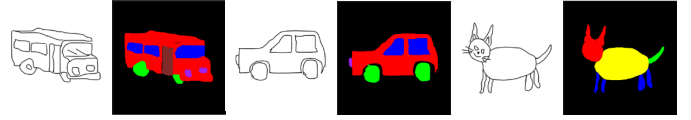


Fig. 10. Sketches (bus, car, cat) and ground truth annotations selected from sketch parse paper [122]. The semantic parts and background are annotated by colors. Best viewed in color.

two-stage sketch simplification [126], which geometrically clusters the stroke lines into groups, and then generates a new line for each group. This solution to sketch simplification is based on the computer graphic techniques. This is to say that in some cases the sketch simplification is a geometric calculation problem, but not a machine learning problem.

With the prevalence of deep learning, CNN-based sketch representations have been used for the sketch simplification [127]. Furthermore, a brand new concept “stroke-level sketch abstraction” appears in sketch community recently.

Note that “stroke-level sketch abstraction” is completely different to “sketch simplification”. Stroke-level sketch abstraction aims to **remove** the redundant information on stroke basis yet keeping the recognizability of the sketch, which can be used for various sketch understanding tasks, including: (i) modeling stroke saliency for sketch and (ii) synthesizing sketches of variable abstraction for a given category. Moreover, stroke-level sketch abstraction is a stroke-level fine-grained sketch understanding task, so that it can not be modeled by CNN efficiently. This is its task-unique challenge. Removing the unnecessary strokes from the given sketch is intrinsically a dynamic process of trading off between the recognizability of a sketch and the number of strokes used to draw it [18]. However, it is difficult to write a concrete math formulations as optimization objective to train the models to abstract free-hand sketch. Thus, reinforcement learning framework is an useful tool for this task.

Muhammad *et al.* [18] proposed a deep reinforcement learning (RL) based RNN model that learns to predict which strokes can be removed without affecting recognizability. Following this, Muhammad *et al.* [128] proposed a novel goal-driven sequential abstractor trained by RL that can abstract sketches depending on different abstraction goals, *e.g.*, category, attribute.

Comments: Generally, the CNN-based deep learning and the non-deep models for sketch simplification work from the perspective of graphics. The deep reinforcement learning based stroke-level sketch abstraction methods focus on how to understand the logic of sketching.

4.2 Multi-Modal

Free-hand sketch stands itself as a special data domain, and it has a lot of cross-modal applications with other modalities. In this section, sketch related cross-modal deep learning topics/models will be discussed, involving visual domains (*e.g.*, natural photo, 3D shape, video) and textual domains. For each sketch-based cross-modal task, its task-specific challenges, typical deep networks, and applications will be introduced and compared. In particular, the sketch-based

cross-modal retrieval and generation will be discussed in detail.

Nowadays, most existing visual retrieval approaches work in the “query-by-example” (QBE) [129] framework that users provide examples of the content that they seek. Compared with the queries from other modalities (*e.g.*, photo, video, text), sketch has its unique superiorities. In some scenarios, it’s difficult/unpractical for user to provide the photos or videos to describe what object or content they want to search. Moreover, sometimes, users do not know the exact name of the object/content that they seek. In particular, editing suitable key words or textual description become more difficult, when users want to search some fine-grained contents. However, every user is able to sketch out the target object/content with the visual details as much as he/she like. Sketch can be used as query example to retrieve natural photo, manga [130], 3D shape, video, *etc.* The advantage of sketch queries will become more significantly when users want to search something in fine-grained/instance-level.

4.2.1 Sketch-Photo Retrieval

Sketch-photo retrieval is also termed as sketch-based image retrieval (SBIR) [2], [5], [131], [132]. Note that in sketch-based image retrieval scenarios, the retrieval is mutual retrieval that both of sketch and photo can be used as query to search target in the gallery of the other modality.

Sketch-photo retrieval is a challenging task. Free-hand sketch is highly abstract and arbitrary, so that it is absolutely not the edge-map of the natural photo. The domain gap between sketch and photo means: (i) the intrinsic difference between sparse line drawing and dense pixel spaces. (ii) the significant distinction between human abstraction expression and the pixel-perfect copy of the real world.

Figure 6 provides a taxonomy for SBIR. From the perspective of evaluation criterion, SBIR can be divided into conventional/coarse-grained SBIR (*i.e.*, category-level SBIR), mid-grained [133], and fine-grained SBIR (*i.e.*, instance-level SBIR). FG-SBIR is essentially a kind of instance-level retrieval [134]. From the perspective of retrieval embedding space, SBIR can be divided into Euclidean/common retrieval and hashing/fast retrieval. From the perspective of supervision involved in training, SBIR can be divided into fully-supervised and zero-shot retrievals.

In coarse-grained SBIR, given a target sketch as query, a ranking list will be returned based on the similarity (*e.g.*, Euclidean distance, Hamming distance). The retrieval will be judged as correct, if the photo ranked at the top has the identical class label as the query. However, in fine-grained SBIR, the evaluation metric is based on the instance-level sketch-photo pairings. The retrieval will be judged as correct only when the returned photo is from the instance pair that contains the query sketch.

With the development of deep learning and popularity of the smart touch-screen devices, it is more convenient to obtain free-hand sketches, so that more and more sketch-based human-computer interaction (SBHCI) applications are able to serve for our daily life. Several sketch-based commodity search engines have already been implemented, *e.g.*, sketch-based skirt image retrieval system [135], fine-grained sketch-based shoe [5], [20]/chair [5], [20]/hand-bag [20] retrieval systems.

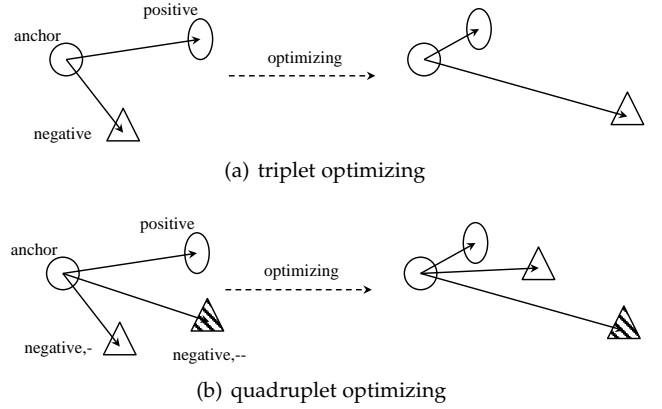


Fig. 11. Illustration of triplet and quadruplet ranking based optimizing objectives. The lengths of solid arrows denote the distances in embedding spaces. In quadruplet illustration, the hollow triangle denotes the negative sample from the category of anchor, while the shaded one denotes the negative sample from the remaining categories.

The following discussion will be divided into three main parts, *i.e.*, category and instance level SBIR, zero-shot SBIR, and SBIR hashing.

Category and Instance Level SBIR:

Sketch-based image retrieval (SBIR) has been widely studied [5], [15], [136], [137], [138], [139], [140], [141], [142], [143], [144], [145], [146] in recent years. The existing SBIR solutions mainly include (i) discriminative methods and (ii) discriminative-generative hybrid methods. The discriminative methods mainly contains: (a) contrastive comparison based methods (implemented by pair-wise loss), (b) ranking based methods [5], [15], (c) deep canonical correlation analysis (DCCA) [147] based methods [148], (d) cross-domain dictionary learning [149], *etc.* The most widely-studied method is ranking based methods, including triplet ranking [140], [150], [151], quadruplet ranking [152].

The common motivation of the existing discriminative SBIR methods is using the discriminative loss terms to learn a joint embedding space where sketches and photos can be compared.

Some previous SBIR works [153], [154], [155] use edge maps (image contours) as sketch approximation to conduct the cross-modal matching. Canny edge detector [156], Edge Boxes toolbox [157], and holistically-nested edge detection (HED) [158] are usually used to extract the edges from natural photos. However, this kind of process does not make full use of the end-to-end feature extraction ability of deep neural network.

Some previous SBIR methods work in two-stage pipeline, where the initial retrieval combines a post-processing re-ranking [153], [155], [159]. Actually, re-ranking is able to refine the initial learned embedding spaces.

The following will introduce the triplet-ranking and quadruplet-ranking SBIR methods in detail.

As shown in Figure 11, given a sketch anchor \mathbf{X}_n and its positive and negative candidates ($\mathbf{X}_{n,+}$, $\mathbf{X}_{n,-}$), the goal of triplet ranking is

$$\mathcal{D}(\mathcal{F}(\mathbf{X}_n), \mathcal{F}(\mathbf{X}_{n,+})) < \mathcal{D}(\mathcal{F}(\mathbf{X}_n), \mathcal{F}(\mathbf{X}_{n,-})). \quad (2)$$

where $\mathcal{D}(\cdot, \cdot)$ is a distance metric (*e.g.*, ℓ_2 distance). In common practice [5], [140], negative sample is usually selected

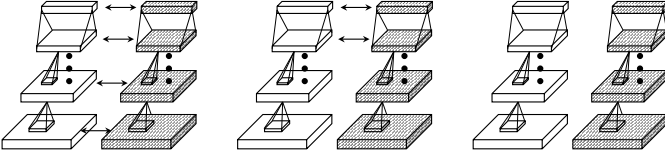


Fig. 12. Different weight sharing manners (left: Siamese, middle: semi-heterogeneous, right: heterogeneous) for CNN-based cross-modal triplet architectures. The hollow and shaded networks denote the branches for sketches and photos, respectively. The double sided arrows indicate sharing of weights.

from the identical class of the anchor. Specifically, the loss function of triplet ranking goes as

$$\mathcal{L}_{triplet} = \sum_{n=1}^N \max(0, \Delta + \|\mathcal{F}_{\Theta}(\mathbf{X}_n) - \mathcal{F}_{\Theta}(\mathbf{X}_{n,+})\|_2^2 - \|\mathcal{F}_{\Theta}(\mathbf{X}_n) - \mathcal{F}_{\Theta}(\mathbf{X}_{n,-})\|_2^2), \quad (3)$$

where Δ is the margin to guarantee the minimum distance between the embedding pairs of $\{\mathcal{F}(\mathbf{X}_n), \mathcal{F}(\mathbf{X}_{n,+})\}$ and $\{\mathcal{F}(\mathbf{X}_n), \mathcal{F}(\mathbf{X}_{n,-})\}$.

For quadruplet ranking [152], the input atom is a quadruplet of anchor \mathbf{X}_n , positive candidate $\mathbf{X}_{n,+}$, negative candidate $\mathbf{X}_{n,-}$ from the class of anchor, negative candidate $\mathbf{X}_{n,-,-}$ from a different class to anchor. As illustrated in Figure 11, the goal of quadruplet ranking is to ensure

$$\mathcal{D}(\mathcal{F}(\mathbf{X}_n), \mathcal{F}(\mathbf{X}_{n,+})) < \mathcal{D}(\mathcal{F}(\mathbf{X}_n), \mathcal{F}(\mathbf{X}_{n,-})) < \mathcal{D}(\mathcal{F}(\mathbf{X}_n), \mathcal{F}(\mathbf{X}_{n,-,-})). \quad (4)$$

Based on this, quadruplet ranking is essentially multi-task or multiple triplet ranking by constructing two extra triplet relationships, so that encode more semantic information into the embedding space. In particular, Seddati *et al.* [152] construct three triplets from each quadruplet, including $triplet_a = \{\mathbf{X}_n, \mathbf{X}_{n,+}, \mathbf{X}_{n,-}\}$, $triplet_b = \{\mathbf{X}_n, \mathbf{X}_{n,+}, \mathbf{X}_{n,-,-}\}$, and $triplet_c = \{\mathbf{X}_n, \mathbf{X}_{n,-}, \mathbf{X}_{n,-,-}\}$. Therefore, the quadruplet ranking loss is defined as

$$\mathcal{L}_{quadruplet} = \mathcal{L}_{triplet_a} + \lambda_b \mathcal{L}_{triplet_b} + \lambda_c \mathcal{L}_{triplet_c}, \quad (5)$$

where λ_b, λ_c are the weights.

Generally, for SBIR tasks, in each triplet or quadruplet, the anchor is from sketch domain, and other components are from photo domain. Both of triplet and quadruplet ranking can be widely used for both category-level and instance-level SBIR tasks.

As shown in Figure 12, for both triplet and quadruplet ranking based SBIR models, the backbone network architecture can adopt (i) Siamese network, (ii) semi-heterogeneous, and (iii) heterogeneous networks: (i) Siamese networks [5] have the fully-shared weight/parameter sharing manners. (ii) Semi-heterogeneous network [160], [161] is in the half-shared manner, where the cross-modal shared layers are located in the high levels of the networks. Thus, semi-heterogeneous networks also can be treated as a kind of late-fusion architecture. (iii) Generally, for heterogeneous networks [162], sketch anchor will have a independent sub-network, and the positive and negative photos will share another sub-network. Various weigh sharing manners are discussed in [163].

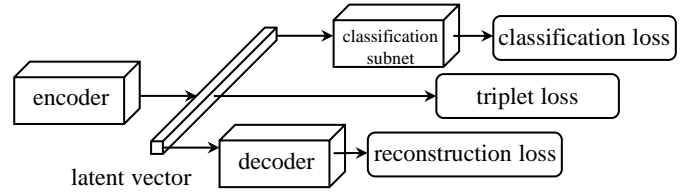


Fig. 13. Architecture of the discriminative-generative hybrid model for FG-SBIR proposed in [165].

Furthermore, triplet or quadruplet ranking based SBIR models can be improved by multi-task training, via combining with classification [15], [163], [164].

Comments: The essential principle of triplet loss is using the local partial orderings to establish the global ordered relationship in the embedding space, so that triplet ranking [38] can be understood as Topological Sorting⁶. The triplet annotations work as a partially ordered set. Compared with other loss functions, the main advantages of triplet loss on sketch-related retrieval are: (i) It helps to involve more locally partial orderings/annotations to learn more fine-grained embedding space. (ii) Given limited training samples of N , their triplet orderings have C_N^3 combinations, producing significant data augmentations. This is benefit to train deep neural networks on small scale sketch datasets. It should be noted that the performance of triplet loss is heavily dependent on (i) choice of margin parameter and (ii) the triplet construction strategies.

Pang *et al.* [165] propose a novel discriminative-generative hybrid model to solve FG-SBIR, which is also intrinsically a multi-task learning network, involving discriminative task and generative task. In particular, as shown in Figure 13, this hybrid model uses a Siamese encoder-decoder architecture with a classification subnet, taking triplets as input. Thus, there are totally three subnets in this model, *i.e.*, encoder subnet, decoder subnet, classification subnet. Given a triplet input, the outputs of encoder subnet will be used for two discriminative tasks, *i.e.*, triplet ranking, object category-level classification. The output of decoder subnet will be used for generative task. Therefore, its loss function has three terms:

$$\mathcal{L} = \mathcal{L}_{triplet} + \lambda_{cla.} \mathcal{L}_{cla.} + \lambda_{rec.} \mathcal{L}_{rec.}, \quad (6)$$

where $\lambda_{cla.}, \lambda_{rec.}$ are the weights, and “cla.” and “rec.” are short for “classification” and “reconstruction”, respectively. Specifically, in (6), $\mathcal{L}_{triplet}$, $\mathcal{L}_{cla.}$, and $\mathcal{L}_{rec.}$ are triplet, cross-entropy softmax, and auto-encoder reconstruction loss terms, respectively.

Sharing some similar motivations with [165], generative adversarial network based SBIR models also have been studied. For instance, Guo *et al.* [166] propose a SBIR model based on conditional generative adversarial network (cGAN) [167].

Zero-Shot SBIR:

In recent years, motivated by the zero-shot validation criterion for supervised photo retrieval [168], zero-shot sketch-based image retrieval (ZS-SBIR) has been studied as a new

6. https://en.wikipedia.org/wiki/Topological_sorting

topic in free-hand sketch community [8], [169], [170], [171], [172], [173], [174], [175], [176], [177]. Similar to the natural photo zero-shot learning/recognition [178], [179], [180], the ZS-SBIR system aims to search the candidate photos for the sketch query that is from the *unseen* categories. “*unseen* categories” denote the categories that have not been involved in the training stage. Thus, addition to the SBIR specific challenges, ZS-SBIR also needs to solve the zero-shot learning challenges, *e.g.*, domain knowledge transfer, domain alignment.

However, even in the single-modal zero-shot learning, the domain shift [181] problems are difficult to solve. In the sketch-photo cross-modal case, the sketch-photo domain shift issues will become more serious, due to the large sketch-photo domain gap and the high-level abstraction of sketch.

Comments: Domain knowledge transfer based domain alignment is key to zero-shot learning problem. Hence, it is important to look for a set of reasonable prior domain knowledge to solve ZS-SBIR. The existing methods use the identical set of prior domain knowledge for both sketch and photo domains. This can not make sense.

Comments: The domain alignment in ZS-SBIR is distinctly different to that in natural photo based zero-shot learning. Photo zero-shot learning only needs align photo domain to the prior domain knowledge. However, ZS-SBIR is inherently a multi-modal problem, so that ZS-SBIR model needs to align two domains of sketch and photo to the prior domain(s). This makes the cross-modal retrieval more challenging.

Comments: In natural photo zero-shot learning, the common semantic knowledge works well, *i.e.*, word vector [182], attribute [183]. However, we should rethink about the semantic knowledge in ZS-SBIR. The domain-gap between sketch and photo is huge, thus it can not be guaranteed that the suitable prior semantic knowledge for natural photo still make sense to sketch. A main bottleneck is that the existing ZS-SBIR always directly “borrow” the semantic knowledge/space from natural photo zero-shot learning models. It is really need to propose the new prior knowledge that is suitable/reasonable for sketch domain.

SBIR Hashing:

In order to achieve more efficient/faster sketch-based image retrieval, in the recent years, researchers start to study the novel task of sketch-based fast image retrieval. The existing ideas to speed up the SBIR retrieval include optimizing the feature coding (*e.g.*, sketch-image hashing [184], [185]), optimizing the feature map (*e.g.*, Asymmetric Feature Maps [186]), *etc.*

In particular, sketch-image hashing has gained more attention, which can be termed as sketch-photo fast retrieval, sketch-photo hashing, fast SBIR, or SBIR hashing. Liu *et al.* [184], for the first time, propose a deep hashing model for SBIR, which is in a classical deep hashing pipeline including: (i) feature extractor network, (ii) hashing layer with binary constraints, and (iii) hashing loss (see Figure 14). This classical pipeline illustrated in Figure 14 has been widely studied in natural photo oriented deep hashing [187], [188], where generally the hashing layer is fully-connected layer using sigmoid or tanh functions as activation, and with discrete binary constraints. The loss functions of deep hash-

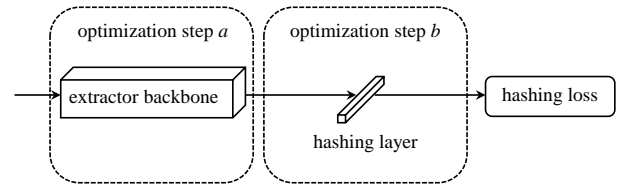


Fig. 14. Illustration of a classical deep hashing pipeline, where the extractor backbone and hashing layer are alternatively optimized in two separate steps.

ing models are often non-differentiable, due to the discrete binary constraints. The common practice is that the feature extractor network/backbone and hashing layer are alternatively optimized in two separate steps. The parameters of hashing layer are fixed, when optimize the parameters of the backbone (vice versa).

Moreover, sketch-photo hashing has also been studied in zero-shot learning setting, termed as “zero-shot sketch-photo hashing” [40].

Comments: The existing SBIR deep hashing models work on the SBIR benchmarks, *i.e.*, Sketchy [15] (75K sketches), TU-Berlin Extended [45] (20K). The scales of these SBIR benchmarks are not large enough, so that they are inappropriate for large-scale deep hashing exploration. This is the bottleneck of the existing SBIR hashing studies.

4.2.2 Sketch-Photo Generation

Sketch and photo based mutual generation (translation/synthesis) is a classical cross-modal topic of sketch research, and often termed as “sketch-photo generation” mainly including: (i) sketch-to-photo generation. (ii) photo-to-sketch generation [189], [190], [191]. In particular, sketch-to-photo generation can be categorized into: (a) sketch to photo, (b) sketch & photo to photo [192], [193], (c) sketch/edge & color to photo [39]. Here, photo mainly refers to natural photo, painting, *etc.* (see Figure 6) Sketch and photo based generation can be used to help users to create or design novel images in a lot of practical application scenarios: sketch-based photo editing [192], [193], sketch to painting generation [194], cloth design [195], [196], sketch to natural photo generation [197], *etc.* In some cases, sketch-photo generations usually involve style transfer [198].

Note that: (i) Sketch-to-photo generation aims to solve the cross-modal translation from the abstract and sparse line drawings to pixel space, different to the well-drawn sketch colorization [199], [200], [201]. (ii) Photo-to-sketch generation does not refer to extracting edge-map from natural photo [157], [202], which needs the model to learn the human sketching patterns to mimic the drawing styles and logic of human.

With the prosperity of Generative Adversarial Net (GAN) [96], sketch and photo based generation are further widely studied [88], [203], [204], [205], [206] based on various generative adversarial networks, *e.g.*, conditional GAN [207], cycle GAN [203], texture GAN [208]. In the GAN framework, both CNN and RNN can be used as backbone networks for the generations across sketch and photo domains. The existing deep sketch-photo generation



Fig. 15. Illustration of view matching across sketch and 3D shape. Images (from left to right: sketch, 3D shape, three random views of the 3D shape) are selected from [212].

models are able to work based on sketch-photo pairing or unpaired samples.

Sketch-photo generation is distinctively different to photo-to-photo generation/translation. The photo-to-photo generation models generally fail to work for sketch-photo generation. The reason is that, the domain gap between sketch and photo is huge, so that it is impossible to conduct the pixel-wise alignment in visual space directly. One main challenge is that, even if sketch-photo pairing information is available, this information only provides weak supervision signals to train a cross-modal translation model. For instance, as stated in [209], conditional generative adversarial networks incorporate hard condition like pixel-wise correspondence alongside the translation process, thus it can not be directly used for sketch-photo generation.

Lu *et al.* [209] propose a contextual GAN that uses sketch as weak constraint to enable that the output photo contours do not necessarily to follow the input edges/sketches. Specifically, sketches and photos are organized as the joint sketch-photo pairs.

4.2.3 Sketch-3D Retrieval

Using sketch as query to retrieve the target 3D model is termed as sketch based 3D retrieval or sketch-3D retrieval. Compared with the sketch based image retrieval, sketch based 3D retrieval is more challenging due to the larger domain gap across the 2D space to the 3D space. For more background knowledge, please read the survey paper [35] on sketch-based 3D retrieval.

Before deep learning era, sketch-3D retrieval is often solved in a two-stage manner, which involves two separated steps [210], *i.e.*, (i) “best view” selection for 3D model: use the automatic procedure to select the representative viewpoints of a given 3D model, hoping that one of the selected viewpoints is similar to that of the query sketches, and (ii) matching between sketches and the 2D projection of the selected best views, based on manually predefined features, *e.g.*, SIFT [95]. More specifically, based on the selected views, 3D models can be projected into 2D space by the line rendering algorithm [211]. Note that each 3D model can be projected as multiple 2D views, and allowed to have multiple best views. Thus, a sketch matches a given 3D model, if it matches one of the views of the 3D model (see Figure 15).

However, as argued in some previous works [3], the view selection is the bottleneck to the two-stage approach, since selecting best views is subjective and ambiguous. Moreover, selecting hand-crafted features manually will also bring biases.

Gradually, sketch based 3D model retrieval has been studied in the end-to-end pipeline based on deep learning [3], [212], [213], [214], [215], [216]. Representatively,

Wang *et al.* [3] propose to use two Siamese networks to learn the sketch and projected views directly in the end-to-end manner, which takes a quadruplet of sketches and projected viewpoints as input, and uses multiple pair-wise losses. In each quadruplet atoms, two sketches ($\mathbf{X}_1, \mathbf{X}_2$) and two viewpoints ($\mathbf{V}_1, \mathbf{V}_2$) are randomly selected from sketch and 3D domains, respectively. For simplicity, Wang *et al.* [3] assume that \mathbf{X}_1 and \mathbf{X}_2 are from the same category sharing a Siamese network, while \mathbf{V}_1 and \mathbf{V}_2 are also from the same category sharing another Siamese network. The loss of a quadruplet is defined as

$$\mathcal{L}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{V}_1, \mathbf{V}_2) = \mathcal{L}_{pair}(\mathbf{X}_1, \mathbf{X}_2) + \mathcal{L}_{pair}(\mathbf{V}_1, \mathbf{V}_2) + \mathcal{L}_{pair}(\mathbf{X}_1, \mathbf{V}_1), \quad (7)$$

Where $\mathcal{L}_{pair}(\cdot, \cdot)$ is the pair-wise loss function that will be detailed in the following. In (7), $\mathcal{L}_{pair}(\mathbf{X}_1, \mathbf{X}_2)$ and $\mathcal{L}_{pair}(\mathbf{V}_1, \mathbf{V}_2)$ will enable the network to learn the category-level similarity within each domain, and $\mathcal{L}_{pair}(\mathbf{X}_1, \mathbf{V}_1)$ will force the network to learn the cross-modal similarity in the end-to-end procedure. Given input samples a and b , one of their pair-wise loss functions can be defined as:

$$\mathcal{L}_{pair}(a, b) = \begin{cases} \alpha \mathcal{D}(\mathcal{F}_a(a), \mathcal{F}_b(b)), & \text{if } y_a \neq y_b, \\ \beta e^{\gamma \mathcal{D}(\mathcal{F}_a(a), \mathcal{F}_b(b))}, & \text{otherwise,} \end{cases} \quad (8)$$

where y_a and y_b are the corresponding class labels, and $\mathcal{F}_a(\cdot)$ and $\mathcal{F}_b(\cdot)$ denotes the feature extractions that have been applied to a and b , respectively.

Addition to the pair-wise deep metric learning, other deep metric learning methods also can be applied to the cross-domain matching across sketch and 3d model, *e.g.*, triplet ranking [217], [218], deep correlation metric learning [219], [220].

Moreover, some previous works also studied how to represent 3D models more comprehensively in sketch based 3D retrieval tasks. For instance, Xie *et al.* [221] propose to represent 3D models by computing the Wasserstein distance [222] based barycenters of multiple projections of 3D models.

Comments: Due to the significant discrepancy across sketches and 3D models, it is unpractical to force sketch domain and 3D domain share the backbone network in sketch-based 3D retrieval models. This is different to that of the sketch-based image retrieval models.

4.2.4 Sketch-3D Generation

The generation across sketch and 3D model is also an interesting cross-modal research topic. Using sketch to generate 3D models/shapes [223] is an extremely challenging and application-driven task, which has some application scenarios, *e.g.*, sketch-based product design [224]. Due to the task-unique challenges, this task is under-studied to date. The existing deep learning based sketch-to-3D generation models are engineered for the highly well-drawn even or professional pencil sketches [225], [226]. Recently, 3D-to-sketch [227] generation also has been studied in deep learning manner.

4.2.5 Sketch-video retrieval

Sketch based video retrieval (SBVR) [228] has been studied before the boom of deep learning. SBVR is also an extremely

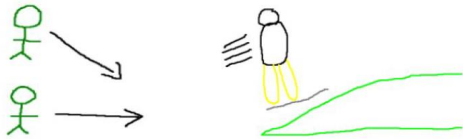


Fig. 16. Sketch samples randomly selected from sketch based video retrieval dataset TSF [228]. The left sketch depicts that two persons are approaching each other. The right one depicts that a person is gliding up the hillside.

challenging task due to the huge domain gap between free-hand sketch and video. In SBVR application scenarios, serving as the query, the advantage of sketch is that humans can use lines or arrow vectors to describe the moving or other dynamic scenes, as illustrated in Figure 16. This is to say that sketch can not only depict static object/scene but also motion information.

The task-specific challenge of SBVR is that both of appearance matching and motion matching across free-hand sketch and video should be considered. The appearance matching and motion matching need to consider the static and dynamic information, respectively.

Furthermore, motion sketch based crowd video retrieval [229], [230] has been studied recently, which is useful for the video surveillance analysis.

4.2.6 Other Sketch-Related Multi-Modal Tasks

Recently, based on deep learning techniques, some other interesting sketch based multi-modal tasks have also been studied, *e.g.*, text-to-sketch generation [231], sketch-based photo classifier generation [17], sketch-to-text pictorial game [232], [233], sketch to photo contour transfer [234]. Similarly, the challenges of these tasks are also mainly due to the large domain gap between free-hand sketches and other modalities.

5 CURRENT BOTTLENECKS

This section will discuss the current bottlenecks of the free-hand sketch community, mainly considering from the perspective of deep learning.

5.1 Lacking of Data and Annotations

For single-modal sketch tasks, QuickDraw [16] is the largest dataset containing more than 50M sketches. However, the sketches in QuickDraw are highly abstract and noisy, due to that QuickDraw online game allows the users to use only 20 seconds to draw a sketch. On the other hand, only the class labels are provided without any other fine-grained annotations (*i.e.*, attributes, groupings), thus, for single-modal tasks, QuickDraw dataset is mainly used for sketch recognition or pre-training.

For multi-modal sketch tasks, most existing methods are proposed in supervised manner. As discussed in Section 4.2.2, the cross-modal pairing information only provides weak supervision signals to train the cross-modal model, due to the large domain gap, *e.g.*, sketch-photo gap, sketch-3D gap. Hence, fine-grained/local annotations

can bring more opportunities to study the subtle cross-modal alignment and matching, *e.g.*, attribute annotations provided in SketchX Shoe dataset [5].

The main application of sketch research is the sketch-based touch-screen interaction that should be capable for any possible users with various drawing styles and abilities. Hence, for both single-modal and multi-modal sketch tasks, sketches selected from any drawing styles and abilities should be studied. However, all the existing sketch datasets are mainly selected by crowd-sourcing and online game. These two approaches are corresponding to two extreme cases: (i) the crowd-sourcing participants draw every stroke deliberately. (ii) the gamers draw too casually, *i.e.*, QuickDraw. This led to the selected sketches are either excessively well-drawn or messily poorly-drawn. Therefore, both single-modal and multi-modal free-hand sketch related tasks need more diverse sketches.

5.2 Lacking of Sketch-Specific Thinking

Free-hand sketch is distinctively different from other visual data, *e.g.*, natural photo, cartoon, professional painting. Therefore, sketch has its domain-unique challenge, and **really needs the sketch-specific designs**. Addition to the visual information, the temporal and geometry information of sketch also needs to be explored. The style diversity, abstraction, and noise of sketch also needs to be taken into account. **However, nowadays, some sketch papers have no sketch-specific designs, while directly copy the techniques that are engineered for natural photo domain. This kind of fake sketch papers hinder the development of sketch community.**

For instance, in photo zero-shot learning, photo domain is aligned with the common semantic domain, *e.g.*, word vector, whereas this domain alignment design recipe does not necessarily translate from photo ZSL setting to sketch ZSL setting. Some semantically dissimilar categories look highly similar in sketch domain, *i.e.*, apple and ball, so that it can not make sense that align sketch domain to the common semantic domain constrainedly. Unfortunately, almost of the existing sketch related zero-shot learning methods borrow the domain alignment techniques directly from the photo ZSL methods.

Another noticeable problem is that sketch is routinely represented by CNNs. It is intuitive that the pixel-by-pixel scan of the CNN procedure involves useless information, due to the highly sparsity in pixel domain of sketch. Sketch can be regarded as temporal sequence or graph, thus sketch needs more sketch-specific feature representations. There is still a lot of room for progress in RNN and GNN based sketch representation/modeling.

6 OPEN PROBLEMS

6.1 Rethinking the Category-Level Validation Criterion for Sketch-Based Cross-Modal Retrieval

As aforementioned in this survey, the coarse-grained sketch-based cross-modal retrieval systems adopt category-level validation criterion, which will determine a returned sample as a true match if this returned sample has the identical class label to the query. This criterion is suitable for semantic

retrieval, yet unsuitable for visual similarity or content based retrieval.

As a kind of query, sketch is able to depict the visual details that are difficult to describe in key words or sentences. For example, you can draw sketches as queries to search the beautiful shoe, which has been found by you in the street yet you don't know how to describe its intricate decorations. This is the advantage of sketch query.

However, if the retrieval is evaluated in category-level, sketch-based cross-modal retrieval will degenerate into classification problem, and using category-based key words is more convenient than sketching. Thus, using sketch as query for the visual/content detail based retrieval will making more sense, *i.e.*, fine-grained retrieval, instance-level retrieval.

6.2 Rethinking the Pre-Training

For some specific sketch tasks, it often happens that existing datasets are not large enough to train the deep models. Many previous works use sketches to fine-tune the deep models pre-trained on ImageNet. However, the domain gap between sketch and photo domains is so large. Compared with the natural photos, edge-map is more close to sketch domain. Therefore, using edge-maps to pre-train deep neural networks for free-hand sketch tasks is a better solution.

7 POTENTIAL RESEARCH DIRECTIONS

Nowadays, not only deep learning is developing rapidly, but also various data collection equipment is constantly updated. This brings many new opportunities and challenges to free-hand sketch community.

There are some potential research directions in future:

(i) With the developments of data collection equipment, collecting 3D sketches [235] is easier than previous. There are many interesting 3D sketch related research topics, *e.g.*, combining virtual reality (VR) [236] and augmented reality (AR) [11], [12], [237]. 3D sketch studies will help to bring the sketch-based human-computer interaction from the 2D planes of the touch-screens to the 3D spaces, enabling users have immersive experience.

(ii) Existing sketch works focus on the sketches that depict objects and scenes. In the practical application scenarios, users want to machines understand the input sketching contents as many as possible. Thus, beyond object and scene, other practical sketch contents also need to be studied, *e.g.*, sheets, curves, histograms [238], maps [239], user interface (UI) prototype drawings [240]. Sketch also can be studied with hand-written character together.

(iii) All the existing free-hand sketches are collected by common touch-screen devices, *e.g.*, mobile phone, ipad. Hence, the position information of strokes is the main feature in previous works. Currently, professional hand-drawn collection equipment is able to sense the pressure along strokes, thus more features for sketch stroke analysis can be collected. This is also worth studying in future.

(iv) Sketch is particularly diverse, abstract, and noisy, so that the sketch-oriented deep models need more robustness. However, the deep model robustness is really under-studied in sketch community to date. In future, sketch can be studied from the perspective of model robustness. Furthermore,

adversarial attack learning also can be involved to sketch study.

(v) Sketch is a special data domain. Beyond human-computer interaction related applications, sketch has some other potential uses. Recently, sketch-like images (*e.g.*, ImageNet-Sketch [29]) are used to evaluate the generalization ability of deep models. Thus, for both uni-modal and cross-modal cases, free-hand sketch can be used as a more difficult/challenging test-bed to evaluate the domain adaptation/transferability/robustness of the deep models. For instance, essentially speaking, both of fine-grained sketch-base image retrieval (FG-SBIR) and person re-identification (Re-ID) are instance-level matching problems. Therefore, researchers can evaluate the generalization ability of person re-identification models on FG-SBIR tasks.

8 CONCLUSION

This survey focuses on reviewing the developments of deep learning techniques for the free-hand sketch. Firstly, background knowledge of sketch is introduced briefly, including the intrinsic traits and domain-unique challenges. The taxonomies of the existing datasets and tasks are provided. For each specific task, deep learning based solutions are summarized and compared in detail. The current bottlenecks, open problems, and potential research directions of the free-hand sketch field have also been discussed.

Sincerely hope this survey will: (i) help the researchers to see the state of the free-hand sketch community within the deep learning background, and (ii) provide insights to the brave researchers to study the unsolved problems for sketch.

REFERENCES

- [1] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *TOG*, 2012.
- [2] R. Hu and J. Collomosse, "A performance evaluation of gradient field hog descriptor for sketch based image retrieval," *CVIU*, 2013.
- [3] F. Wang, L. Kang, and Y. Li, "Sketch-based 3d shape retrieval using convolutional neural networks," in *CVPR*, 2015.
- [4] Q. Yu, Y. Yang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-net that beats humans," in *BMVC*, 2015.
- [5] Q. Yu, F. Liu, Y.-Z. Song, T. Xiang, T. M. Hospedales, and C.-C. Loy, "Sketch me that shoe," in *CVPR*, 2016.
- [6] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-net: A deep neural network that beats humans," *IJCV*, 2017.
- [7] P. Xu, Y. Huang, T. Yuan, K. Pang, Y.-Z. Song, T. Xiang, T. M. Hospedales, Z. Ma, and J. Guo, "Sketchmate: Deep hashing for million-scale human sketch retrieval," in *CVPR*, 2018.
- [8] S. Dey, P. Riba, A. Dutta, J. Lladós, and Y.-Z. Song, "Doodle to search: Practical zero-shot sketch-based image retrieval," in *CVPR*, 2019.
- [9] F. Huang, J. F. Canny, and J. Nichols, "Swire: Sketch-based user interface retrieval," in *CHI*, 2019.
- [10] S. Suleri, V. P. Sermuga Pandian, S. Shishkovets, and M. Jarke, "Eve: A sketch-based software prototyping workbench," in *CHI*, 2019.
- [11] K. C. Kwan and H. Fu, "Mobi3dsketch: 3d sketching in mobile ar," in *CHI*, 2019.
- [12] D. Gasques, J. G. Johnson, T. Sharkey, and N. Weibel, "What you sketch is what you get: Quick and easy augmented reality prototyping with pintar," in *CHI*, 2019.
- [13] I. E. Sutherland, "Sketchpad a man-machine graphical communication system," *Simulation*, 1964.
- [14] C. F. Herot, "Graphical input through machine recognition of sketches," *TOG*, 1976.

- [15] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The sketchy database: learning to retrieve badly drawn bunnies," *TOG*, 2016.
- [16] D. Ha and D. Eck, "A neural representation of sketch drawings," in *ICLR*, 2018.
- [17] C. Hu, D. Li, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-classifier: Sketch-based photo classifier generation," in *CVPR*, 2018.
- [18] U. Riaz Muhammad, Y. Yang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Learning deep sketch abstraction," in *CVPR*, 2018.
- [19] Y. Qi and Z.-H. Tan, "Sketchsegnet+: An end-to-end learning of rnn for multi-class sketch semantic segmentation," *Access*, 2019.
- [20] J. Song, Q. Yu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Deep spatial-semantic attention for fine-grained sketch-based image retrieval," in *ICCV*, 2017.
- [21] S. Ouyang, T. M. Hospedales, Y.-Z. Song, and X. Li, "Forget-menot: Memory-aware forensic facial sketch matching," in *CVPR*, 2016.
- [22] C. Hu, D. Li, Y.-Z. Song, and T. M. Hospedales, "Now you see me: Deep face hallucination for unviewed sketches," in *BMVC*, 2017.
- [23] S. Nagpal, M. Singh, R. Singh, M. Vatsa, A. Noore, and A. Majumdar, "Face sketch matching via coupled deep transform learning," in *ICCV*, 2017.
- [24] 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.
- [25] M. Huang, J. Lin, N. Chen, W. An, and W. Zhu, "Reversed sketch: A scalable and comparable shape representation," *PR*, 2018.
- [26] L. Pang, Y. Wang, Y.-Z. Song, T. Huang, and Y. Tian, "Cross-domain adversarial feature learning for sketch re-identification," in *MM*, 2018.
- [27] S.-B. Chen, P.-C. Wang, B. Luo, C. H. Ding, and J. Zhang, "Sragan: Generating colour landscape photograph from sketch," in *IJCNN*, 2019.
- [28] M. R. Amer, S. Yousefi, R. Raich, and S. Todorovic, "Monocular extraction of 2.1 d sketch using constrained convex optimization," *IJCV*, 2015.
- [29] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, "Learning robust global representations by penalizing local predictive power," in *NeurIPS*, 2019.
- [30] K. Chen, I. Rabkina, M. D. McLure, and K. D. Forbus, "Human-like sketch object recognition via analogical learning," in *AAAI*, 2019.
- [31] X. Han, K. Hou, D. Du, Y. Qiu, Y. Yu, K. Zhou, and S. Cui, "Caricatureshop: Personalized and photorealistic caricature sketching," *arXiv preprint arXiv:1807.09064*, 2018.
- [32] C. Zou, Q. Yu, R. Du, H. Mo, Y.-Z. Song, T. Xiang, C. Gao, B. Chen, and H. Zhang, "Sketchyscene: Richly-annotated scene sketches," in *ECCV*, 2018.
- [33] L. Zhao, F. Han, X. Peng, X. Zhang, M. Kapadia, V. Pavlovic, and D. N. Metaxas, "Cartoonish sketch-based face editing in videos using identity deformation transfer," *Computers Graphics*, 2019.
- [34] J. Delanoy, M. Aubry, P. Isola, A. A. Efros, and A. Bousseau, "3d sketching using multi-view deep volumetric prediction," *CGIT*, 2018.
- [35] B. Li, Y. Lu, A. Godil, T. Schreck, B. Bustos, A. Ferreira, T. Furuya, M. J. Fonseca, H. Johan, T. Matsuda *et al.*, "A comparison of methods for sketch-based 3d shape retrieval," *CVIU*, 2014.
- [36] Y. Li and W. Li, "A survey of sketch-based image retrieval," *Machine Vision and Applications*, 2018.
- [37] X. Zhang, X. Li, Y. Liu, and F. Feng, "A survey on freehand sketch recognition and retrieval," *Image and Vision Computing*, 2019.
- [38] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015.
- [39] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *CVPR*, 2017.
- [40] Y. Shen, L. Liu, F. Shen, and L. Shao, "Zero-shot sketch-image hashing," in *CVPR*, 2018.
- [41] K. Li, K. Pang, J. Song, Y.-Z. Song, T. Xiang, T. M. Hospedales, and H. Zhang, "Universal sketch perceptual grouping," in *ECCV*, 2018.
- [42] J. Choi, H. Cho, J. Song, and S. M. Yoon, "Sketchhelper: Real-time stroke guidance for freehand sketch retrieval," *TMM*, 2019.
- [43] F. Wang, S. Lin, H. Wu, H. Li, R. Wang, X. Luo, and X. He, "Sp-fusionnet: Sketch segmentation using multi-modal data fusion," in *ICME*, 2019.
- [44] R. K. Sarvadevabhatla, S. Suresh, and R. V. Babu, "Object category understanding via eye fixations on freehand sketches," *TIP*, 2017.
- [45] H. Zhang, S. Liu, C. Zhang, W. Ren, R. Wang, and X. Cao, "Sketchnet: Sketch classification with web images," in *CVPR*, 2016.
- [46] T. Jiang, G.-S. Xia, and Q. Lu, "Sketch-based aerial image retrieval," in *ICIP*, 2017.
- [47] T.-B. Jiang, G.-S. Xia, Q.-K. Lu, and W.-M. Shen, "Retrieving aerial scene images with learned deep image-sketch features," *JCST*, 2017.
- [48] X. Wang, X. Duan, and X. Bai, "Deep sketch feature for cross-domain image retrieval," *Neurocomputing*, 2016.
- [49] M. Eitz, R. Richter, T. Boubekur, K. Hildebrand, and M. Alexa, "Sketch-based shape retrieval," *TOG*, 2012.
- [50] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, H. Fu, T. Furuya, H. Johan *et al.*, "Shrec14 track: Extended large scale sketch-based 3d shape retrieval," in *Eurographics workshop on 3D object retrieval*, 2014.
- [51] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *ICCV*, 2017.
- [52] C. Xiao, C. Wang, L. Zhang, and L. Zhang, "Sketch-based image retrieval via shape words," in *ICMR*, 2015.
- [53] L. Castrejon, Y. Aytar, C. Vondrick, H. Pirsiavash, and A. Torralba, "Learning aligned cross-modal representations from weakly aligned data," in *CVPR*, 2016.
- [54] T. Kato, T. Kurita, N. Otsu, and K. Hirata, "A sketch retrieval method for full color image database-query by visual example," in *ICPR*, 1992.
- [55] M. Eitz, K. Hildebrand, T. Boubekur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *TVCG*, 2010.
- [56] Y. Zhang, X. Qian, X. Tan, J. Han, and Y. Tang, "Sketch-based image retrieval by salient contour reinforcement," *TMM*, 2016.
- [57] R. K. Sarvadevabhatla *et al.*, "Analyzing structural characteristics of object category representations from their semantic-part distributions," in *MM*, 2016.
- [58] R. K. Sarvadevabhatla, I. Dwivedi, A. Biswas, S. Manocha *et al.*, "Sketchparse: Towards rich descriptions for poorly drawn sketches using multi-task hierarchical deep networks," in *MM*, 2017.
- [59] A. Jenal, N. Savinov, T. Sattler, and G. Chaurasia, "Rnn-based generative model for fine-grained sketching," *arXiv preprint arXiv:1901.03991*, 2019.
- [60] Y. Matsui, T. Shiratori, and K. Aizawa, "Drawfromdrawings: 2d drawing assistance via stroke interpolation with a sketch database," *TVCG*, 2016.
- [61] K. D. Forbus, B. Garnier, B. Tikoff, W. Marko, M. Usher, and M. McLure, "Sketch worksheets in stem classrooms: Two deployments," in *AAAI*, 2018.
- [62] Y. Ye, Y. Lu, and H. Jiang, "Human's scene sketch understanding," in *ICMR*, 2016.
- [63] Y. Xie, P. Xu, and Z. Ma, "Deep zero-shot learning for scene sketch," *arXiv preprint arXiv:1905.04510*, 2019.
- [64] O. Seddati, S. Dupont, and S. Mahmoudi, "Deepsketch: deep convolutional neural networks for sketch recognition and similarity search," in *CBMI*, 2015.
- [65] Y. Zhang, Y. Zhang, and X. Qian, "Deep neural networks for free-hand sketch recognition," in *Pacific Rim Conference on Multimedia*, 2016.
- [66] J. Guo, C. Wang, E. Roman-Rangel, H. Chao, and Y. Rui, "Building hierarchical representations for oracle character and sketch recognition," *TIP*, 2016.
- [67] P. Ballester and R. M. Araujo, "On the performance of googlenet and alexnet applied to sketches," in *AAAI*, 2016.
- [68] O. Seddati, S. Dupont, and S. Mahmoudi, "Deepsketch 2: Deep convolutional neural networks for partial sketch recognition," in *CBMI*, 2016.
- [69] H. Zhang, P. She, Y. Liu, J. Gan, X. Cao, and H. Foroosh, "Learning structural representations via dynamic object landmarks discovery for sketch recognition and retrieval," *TIP*, 2019.
- [70] O. Seddati, S. Dupont, and S. Mahmoudi, "Deepsketch2image: deep convolutional neural networks for partial sketch recognition and image retrieval," in *MM*, 2016.
- [71] K. Zhang, W. Luo, L. Ma, and H. Li, "Cousin network guided sketch recognition via latent attribute warehouse," in *AAAI*, 2019.

- [72] R. K. Sarvadevabhatla, J. Kundu *et al.*, “Enabling my robot to play pictionary: Recurrent neural networks for sketch recognition,” in *MM*, 2016.
- [73] Q. Jia, M. Yu, X. Fan, and H. Li, “Sequential dual deep learning with shape and texture features for sketch recognition,” *arXiv preprint arXiv:1708.02716*, 2017.
- [74] J.-Y. He, X. Wu, Y.-G. Jiang, B. Zhao, and Q. Peng, “Sketch recognition with deep visual-sequential fusion model,” in *MM*, 2017.
- [75] A. Prabhu, V. Batchu, S. A. Munagala, R. Gajawada, and A. Nambodiri, “Distribution-aware binarization of neural networks for sketch recognition,” in *WACV*, 2018.
- [76] L. Li, C. Zou, Y. Zheng, Q. Su, H. Fu, and C.-L. Tai, “Sketch-r2cnn: An attentive network for vector sketch recognition,” *arXiv preprint arXiv:1811.08170*, 2018.
- [77] P. Xu, C. K. Joshi, and X. Bresson, “Multi-graph transformer for free-hand sketch recognition,” *arXiv preprint arXiv:1912.11258*, 2019.
- [78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012.
- [79] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, “Bayesian face revisited: A joint formulation,” in *ECCV*, 2012.
- [80] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [81] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.
- [82] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, “Pixel recurrent neural networks,” *arXiv preprint arXiv:1601.06759*, 2016.
- [83] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [84] 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*, 2015.
- [85] R. K. Sarvadevabhatla and R. V. Babu, “Freehand sketch recognition using deep features,” *arXiv preprint arXiv:1502.00254*, 2015.
- [86] B. Graham, “Spatially-sparse convolutional neural networks,” *arXiv preprint arXiv:1409.6070*, 2014.
- [87] Y. Zheng, H. Yao, X. Sun, S. Zhang, S. Zhao, and F. Porikli, “Sketch-specific data augmentation for freehand sketch recognition,” *arXiv preprint arXiv:1910.06038*, 2019.
- [88] R. Liu, Q. Yu, and S. Yu, “An unpaired sketch-to-photo translation model,” *arXiv preprint arXiv:1909.08313*, 2019.
- [89] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.
- [90] A. Mishra and A. K. Singh, “Deep embedding using bayesian risk minimization with application to sketch recognition,” in *ACCV*, 2018.
- [91] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, 1995.
- [92] F. Liu, X. Deng, Y.-K. Lai, Y.-J. Liu, C. Ma, and H. Wang, “Sketchgan: Joint sketch completion and recognition with generative adversarial network,” in *CVPR*, 2019.
- [93] F. Wang and Y. Li, “Spatial matching of sketches without point correspondence,” in *ICIP*, 2015.
- [94] A. Creswell and A. A. Bharath, “Adversarial training for sketch retrieval,” in *ECCV*, 2016.
- [95] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, 2004.
- [96] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NIPS*, 2014.
- [97] N. Jaques, J. McCleary, J. Engel, D. Ha, F. Bertsch, R. Picard, and D. Eck, “Learning via social awareness: Improving a deep generative sketching model with facial feedback,” *arXiv preprint arXiv:1802.04877*, 2018.
- [98] S. Balasubramanian, V. N. Balasubramanian *et al.*, “Teaching gans to sketch in vector format,” *arXiv preprint arXiv:1904.03620*, 2019.
- [99] T. Zhou, C. Fang, Z. Wang, J. Yang, B. Kim, Z. Chen, J. Brandt, and D. Terzopoulos, “Learning to sketch with deep q networks and demonstrated strokes,” *arXiv preprint arXiv:1810.05977*, 2018.
- [100] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, 1997.
- [101] N. Cao, X. Yan, Y. Shi, and C. Chen, “Ai-sketcher: A deep generative model for producing high-quality sketches,” in *AAAI*, 2019.
- [102] J. P. Collomosse, G. McNeill, and L. Watts, “Free-hand sketch grouping for video retrieval,” in *ICPR*, 2008.
- [103] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt, “A century of gestalt psychology in visual perception: I. perceptual grouping and figure-ground organization,” *Psychological bulletin*, 2012.
- [104] K. Koffka, *Principles of Gestalt psychology*. Routledge, 2013.
- [105] Z. Sun, C. Wang, L. Zhang, and L. Zhang, “Free hand-drawn sketch segmentation,” in *ECCV*, 2012.
- [106] Y. Qi, J. Guo, Y.-Z. Song, T. Xiang, H. Zhang, and Z.-H. Tan, “Im2sketch: Sketch generation by unconflicted perceptual grouping,” *Neurocomputing*, 2015.
- [107] Y. Qi, Y.-Z. Song, T. Xiang, H. Zhang, T. Hospedales, Y. Li, and J. Guo, “Making better use of edges via perceptual grouping,” in *CVPR*, 2015.
- [108] X. Liu, T.-T. Wong, and P.-A. Heng, “Closure-aware sketch simplification,” *TOG*, 2015.
- [109] X. Wang, X. Chen, and Z. Zha, “Sketchpointnet: A compact network for robust sketch recognition,” in *ICIP*, 2018.
- [110] Y. Li, Y.-Z. Song, T. M. Hospedales, and S. Gong, “Free-hand sketch synthesis with deformable stroke models,” *IJCV*, 2017.
- [111] K. Li, K. Pang, Y.-Z. Song, T. Xiang, T. M. Hospedales, and H. Zhang, “Toward deep universal sketch perceptual grouper,” *TIP*, 2019.
- [112] Z. Huang, H. Fu, and R. W. Lau, “Data-driven segmentation and labeling of freehand sketches,” *TOG*, 2014.
- [113] R. G. Schneider and T. Tuytelaars, “Example-based sketch segmentation and labeling using crfs,” *TOG*, 2016.
- [114] C. Wang, B. Yang, and Y. Liao, “Unsupervised image segmentation using convolutional autoencoder with total variation regularization as preprocessing,” in *ICASSP*, 2017.
- [115] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *TPAMI*, 2017.
- [116] K. Kaiyrbekov and M. Sezgin, “Stroke-based sketched symbol reconstruction and segmentation,” *arXiv preprint arXiv:1901.03427*, 2019.
- [117] X. Wu, Y. Qi, J. Liu, and J. Yang, “Sketchsegnet: A rnn model for labeling sketch strokes,” in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2018.
- [118] L. Li, H. Fu, and C.-L. Tai, “Fast sketch segmentation and labeling with deep learning,” *IEEE computer graphics and applications*, 2018.
- [119] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas *et al.*, “A scalable active framework for region annotation in 3d shape collections,” *TOG*, 2016.
- [120] J. Jiang, R. Wang, S. Lin, and F. Wang, “Sfsegnet: Parse freehand sketches using deep fully convolutional networks,” in *IJCNN*, 2019.
- [121] K. Mukherjee, R. X. Hawkins, and J. E. Fan, “Communicating semantic part information in drawings,” in *Annual Conference of the Cognitive Science Society*, 2019.
- [122] Y. Zheng, H. Yao, and X. Sun, “Deep semantic parsing of freehand sketches with homogeneous transformation, soft-weighted loss, and staged learning,” *arXiv preprint arXiv:1910.06023*, 2019.
- [123] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015.
- [124] Y. Chien, W.-C. Lin, T.-S. Huang, and J.-H. Chuang, “Line drawing simplification by stroke translation and combination,” in *ICGIP*, 2014.
- [125] T. Ogawa, Y. Matsui, T. Yamasaki, and K. Aizawa, “Sketch simplification by classifying strokes,” in *ICPR*, 2016.
- [126] P. Barla, J. Thollot, and F. X. Sillion, “Geometric clustering for line drawing simplification,” in *TOG*, 2005.
- [127] R. K. Sarvadevabhatla *et al.*, “Eye of the dragon: Exploring discriminatively minimalist sketch-based abstractions for object categories,” in *MM*, 2015.
- [128] U. R. Muhammad, Y. Yang, T. M. Hospedales, T. Xiang, and Y.-Z. Song, “Goal-driven sequential data abstraction,” in *ICCV*, 2019.
- [129] Y.-P. Tan, S. R. Kulkarni, and P. J. Ramadge, “A framework for measuring video similarity and its application to video query by example,” in *ICIP*, 1999.
- [130] Y. Matsui, “Challenge for manga processing: Sketch-based manga retrieval,” in *MM*, 2015.

- [131] P. Xu, K. Li, Z. Ma, Y.-Z. Song, L. Wang, and J. Guo, "Cross-modal subspace learning for sketch-based image retrieval: A comparative study," in *IC-NIDC*, 2016.
- [132] P. Xu, Q. Yin, Y. Huang, Y.-Z. Song, Z. Ma, L. Wang, T. Xiang, W. B. Kleijn, and J. Guo, "Cross-modal subspace learning for fine-grained sketch-based image retrieval," *Neurocomputing*, 2018.
- [133] T. Bui, L. Ribeiro, M. Ponti, and J. Collomosse, "Deep manifold alignment for mid-grain sketch based image retrieval," in *ACCV*, 2018.
- [134] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *TPAMI*, 2017.
- [135] S.-i. Kondo, M. Toyoura, and X. Mao, "Sketch based skirt image retrieval," in *Proceedings of the 4th Joint Symposium on Computational Aesthetics, Non-Photorealistic Animation and Rendering, and Sketch-Based Interfaces and Modeling*, 2014.
- [136] S. Chopra, R. Hadsell, Y. LeCun *et al.*, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*, 2005.
- [137] P. Xu, Q. Yin, Y. Qi, Y.-Z. Song, Z. Ma, L. Wang, and J. Guo, "Instance-level coupled subspace learning for fine-grained sketch-based image retrieval," in *ECCV Workshops*, 2016.
- [138] Y. Qi, Y.-Z. Song, H. Zhang, and J. Liu, "Sketch-based image retrieval via siamese convolutional neural network," in *ICIP*, 2016.
- [139] J. Song, Y.-Z. Song, T. Xiang, T. M. Hospedales, and X. Ruan, "Deep multi-task attribute-driven ranking for fine-grained sketch-based image retrieval," in *BMVC*, 2016.
- [140] Q. Yu, X. Chang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "The devil is in the middle: Exploiting mid-level representations for cross-domain instance matching," *arXiv preprint arXiv:1711.08106*, 2017.
- [141] J. Collomosse, T. Bui, M. J. Wilber, C. Fang, and H. Jin, "Sketching with style: Visual search with sketches and aesthetic context," in *ICCV*, 2017.
- [142] J. Song, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Fine-grained image retrieval: the text/sketch input dilemma," in *BMVC*, 2017.
- [143] F. Huang, Y. Cheng, C. Jin, Y. Zhang, and T. Zhang, "Deep multimodal embedding model for fine-grained sketch-based image retrieval," in *SIGIR*, 2017.
- [144] S. Dey, A. Dutta, S. K. Ghosh, E. Valveny, J. Lladós, and U. Pal, "Learning cross-modal deep embeddings for multi-object image retrieval using text and sketch," in *ICPR*, 2018.
- [145] Y. Wang, F. Huang, Y. Zhang, R. Feng, T. Zhang, and W. Fan, "Deep cascaded cross-modal correlation learning for fine-grained sketch-based image retrieval," *PR*, 2019.
- [146] K. Pang, K. Li, Y. Yang, H. Zhang, T. M. Hospedales, T. Xiang, and Y.-Z. Song, "Generalising fine-grained sketch-based image retrieval," in *CVPR*, 2019.
- [147] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *ICML*, 2013.
- [148] F. Huang, C. Jin, Y. Zhang, and T. Zhang, "Towards sketch-based image retrieval with deep cross-modal correlation learning," in *ICME*, 2017.
- [149] D. Xu, X. Alameda-Pineda, J. Song, E. Ricci, and N. Sebe, "Cross-paced representation learning with partial curricula for sketch-based image retrieval," *TIP*, 2018.
- [150] C. Li, Y. Zhou, and J. Yang, "Sketch-based image retrieval via a semi-heterogeneous cross-domain network," in *ICME Workshops*, 2019.
- [151] J. Collomosse, T. Bui, and H. Jin, "Livesketch: Query perturbations for guided sketch-based visual search," in *CVPR*, 2019.
- [152] O. Seddati, S. Dupont, and S. Mahmoudi, "Quadruplet networks for sketch-based image retrieval," in *ICMR*, 2017.
- [153] S. D. Bhattacharjee, J. Yuan, W. Hong, and X. Ruan, "Query adaptive instance search using object sketches," in *MM*, 2016.
- [154] J. Lei, K. Zheng, H. Zhang, X. Cao, N. Ling, and Y. Hou, "Sketch based image retrieval via image-aided cross domain learning," in *ICIP*, 2017.
- [155] S. D. Bhattacharjee, J. Yuan, Y. Huang, J. Meng, and L. Duan, "Query adaptive multiview object instance search and localization using sketches," *TMM*, 2018.
- [156] J. Canny, "A computational approach to edge detection," *TPAMI*, 1986.
- [157] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014.
- [158] S. Xie and Z. Tu, "Holistically-nested edge detection," in *ICCV*, 2015.
- [159] F. Huang, C. Jin, Y. Zhang, K. Weng, T. Zhang, and W. Fan, "Sketch-based image retrieval with deep visual semantic descriptor," *PR*, 2018.
- [160] T. Bui, L. Ribeiro, M. Ponti, and J. Collomosse, "Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network," *CVIU*, 2017.
- [161] J. Lei, Y. Song, B. Peng, Z. Ma, L. Shao, and Y.-Z. Song, "Semi-heterogeneous three-way joint embedding network for sketch-based image retrieval," *TCSVT*, 2019.
- [162] H. Zhang, C. Zhang, and M. Wu, "Sketch-based cross-domain image retrieval via heterogeneous network," in *VCIP*, 2017.
- [163] T. Bui, L. Ribeiro, M. Ponti, and J. Collomosse, "Generalisation and sharing in triplet convnets for sketch based visual search," *arXiv preprint arXiv:1611.05301*, 2016.
- [164] H. Lin, Y. Fu, P. Lu, S. Gong, X. Xue, and Y.-G. Jiang, "Tc-net for isbir: Triplet classification network for instance-level sketch based image retrieval," in *MM*, 2019.
- [165] K. Pang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Cross-domain generative learning for fine-grained sketch-based image retrieval," in *BMVC*, 2017.
- [166] L. Guo, J. Liu, Y. Wang, Z. Luo, W. Wen, and H. Lu, "Sketch-based image retrieval using generative adversarial networks," in *MM*, 2017.
- [167] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
- [168] A. Sablayrolles, M. Douze, N. Usunier, and H. Jégou, "How should we evaluate supervised hashing?" in *ICASSP*, 2017.
- [169] P. Lu, G. Huang, Y. Fu, G. Guo, and H. Lin, "Learning large euclidean margin for sketch-based image retrieval," *arXiv preprint arXiv:1812.04275*, 2018.
- [170] W. Thong, P. Mettes, and C. G. Snoek, "Open cross-domain visual search," *arXiv preprint arXiv:1911.08621*, 2019.
- [171] T. Dutta and S. Biswas, "Style-guided zero-shot sketch-based image retrieval," in *BMVC*, 2019.
- [172] A. Dutta and Z. Akata, "Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval," in *CVPR*, 2019.
- [173] S. Kiran Yelamarthi, S. Krishna Reddy, A. Mishra, and A. Mittal, "A zero-shot framework for sketch based image retrieval," in *ECCV*, 2018.
- [174] J. Li, Z. Ling, L. Niu, and L. Zhang, "Bi-directional domain translation for zero-shot sketch-based image retrieval," *arXiv preprint arXiv:1911.13251*, 2019.
- [175] A. Pandey, A. Mishra, V. Kumar Verma, and A. Mittal, "Adversarial joint-distribution learning for novel class sketch-based image retrieval," in *ICCV Workshops*, 2019.
- [176] V. Kumar Verma, A. Mishra, A. Mishra, and P. Rai, "Generative model for zero-shot sketch-based image retrieval," in *CVPR Workshops*, 2019.
- [177] Q. Liu, L. Xie, H. Wang, and A. L. Yuille, "Semantic-aware knowledge preservation for zero-shot sketch-based image retrieval," in *ICCV*, 2019.
- [178] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Transductive multi-view zero-shot learning," *TPAMI*, 2015.
- [179] Y. Fu, T. Xiang, Y.-G. Jiang, X. Xue, L. Sigal, and S. Gong, "Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content," *IEEE Signal Processing Magazine*, 2018.
- [180] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *TIST*, 2019.
- [181] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *ICCV*, 2015.
- [182] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.
- [183] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *TPAMI*, 2013.
- [184] L. Liu, F. Shen, Y. Shen, X. Liu, and L. Shao, "Deep sketch hashing: Fast free-hand sketch-based image retrieval," in *CVPR*, 2017.
- [185] J. Zhang, F. Shen, L. Liu, F. Zhu, M. Yu, L. Shao, H. Tao Shen, and L. Van Gool, "Generative domain-migration hashing for sketch-to-image retrieval," in *ECCV*, 2018.
- [186] G. Toliás and O. Chum, "Asymmetric feature maps with application to sketch based retrieval," in *CVPR*, 2017.

- [187] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *CVPR Workshops*, 2015.
- [188] J. Wang, T. Zhang, N. Sebe, H. T. Shen *et al.*, "A survey on learning to hash," *TPAMI*, 2017.
- [189] J. Song, K. Pang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Learning to sketch with shortcut cycle consistency," in *CVPR*, 2018.
- [190] M. Li, Z. Lin, R. Mech, E. Yumer, and D. Ramanan, "Photo-sketching: Inferring contour drawings from images," in *WACV*, 2019.
- [191] Y. Zhang, G. Su, Y. Qi, and J. Yang, "Unpaired image-to-sketch translation network for sketch synthesis," in *VCIP*, 2019.
- [192] T. Portenier, Q. Hu, A. Szabo, S. A. Bigdeli, P. Favaro, and M. Zwicker, "Faceshop: Deep sketch-based face image editing," *TOG*, 2018.
- [193] Y. Jo and J. Park, "Sc-fegan: Face editing generative adversarial network with user's sketch and color," in *ICCV*, 2019.
- [194] J. Li, S. Liu, and M. Cao, "Line artist: A multiple style sketch to painting synthesis scheme," *arXiv preprint arXiv:1803.06647*, 2018.
- [195] M. Li, A. Sheffer, E. Grinspun, and N. Vining, "Foldsketch: Enriching garments with physically reproducible folds," *TOG*, 2018.
- [196] T. Y. Wang, D. Ceylan, J. Popovic, and N. J. Mitra, "Learning a shared shape space for multimodal garment design," *arXiv preprint arXiv:1806.11335*, 2018.
- [197] A. Ghosh, R. Zhang, P. K. Dokania, O. Wang, A. A. Efros, P. H. S. Torr, and E. Shechtman, "Interactive sketch & fill: Multiclass sketch-to-image translation," in *ICCV*, 2019.
- [198] J. Collomosse, T. Bui, M. J. Wilber, C. Fang, and H. Jin, "Sketching with style: Visual search with sketches and aesthetic context," in *ICCV*, 2017.
- [199] C. Zou, H. Mo, R. Du, X. Wu, C. Gao, and H. Fu, "Lucss: Language-based user-customized colourization of scene sketches," *arXiv preprint arXiv:1808.10544*, 2018.
- [200] L. Zhang, C. Li, T.-T. Wong, Y. Ji, and C. Liu, "Two-stage sketch colorization," in *TOG*, 2018.
- [201] C. Zou, H. Mo, C. Gao, R. Du, and H. Fu, "Language-based colorization of scene sketches," *TOG*, 2019.
- [202] X. Soria, E. Riba, and A. D. Sappa, "Dense extreme inception network: Towards a robust cnn model for edge detection," *arXiv preprint arXiv:1909.01955*, 2019.
- [203] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.
- [204] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.
- [205] W. Chen and J. Hays, "Sketchygan: Towards diverse and realistic sketch to image synthesis," in *CVPR*, 2018.
- [206] W. Xia, Y. Yang, and J.-H. Xue, "Cali-sketch: Stroke calibration and completion for high-quality face image generation from poorly-drawn sketches," *arXiv preprint arXiv:1911.00426*, 2019.
- [207] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
- [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.
- [209] Y. Lu, S. Wu, Y.-W. Tai, and C.-K. Tang, "Image generation from sketch constraint using contextual gan," in *ECCV*, 2018.
- [210] B. Li, Y. Lu, and J. Shen, "A semantic tree-based approach for sketch-based 3d model retrieval," in *ICPR*, 2016.
- [211] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella, "Suggestive contours for conveying shape," in *TOG*, 2003.
- [212] H. Li, H. Wu, X. He, S. Lin, R. Wang, and X. Luo, "Multi-view pairwise relationship learning for sketch based 3d shape retrieval," in *ICME*, 2017.
- [213] F. Zhu, J. Xie, and Y. Fang, "Learning cross-domain neural networks for sketch-based 3d shape retrieval," in *AAAI*, 2016.
- [214] Y. Ye, B. Li, and Y. Lu, "3d sketch-based 3d model retrieval with convolutional neural network," in *ICPR*, 2016.
- [215] J. Chen and Y. Fang, "Deep cross-modality adaptation via semantics preserving adversarial learning for sketch-based 3d shape retrieval," in *ECCV*, 2018.
- [216] J. Chen, J. Qin, L. Liu, F. Zhu, F. Shen, J. Xie, and L. Shao, "Deep sketch-shape hashing with segmented 3d stochastic viewing," in *CVPR*, 2019.
- [217] A. Qi, Y.-Z. Song, and T. Xiang, "Semantic embedding for sketch-based 3d shape retrieval," in *BMVC*, 2018.
- [218] S. Kuwabara, R. Ohbuchi, and T. Furuya, "Query by partially-drawn sketches for 3d shape retrieval," in *2019 International Conference on Cyberworlds*, 2019.
- [219] G. Dai, J. Xie, F. Zhu, and Y. Fang, "Deep correlated metric learning for sketch-based 3d shape retrieval," in *AAAI*, 2017.
- [220] G. Dai, J. Xie, and Y. Fang, "Deep correlated holistic metric learning for sketch-based 3d shape retrieval," *TIP*, 2018.
- [221] J. Xie, G. Dai, F. Zhu, and Y. Fang, "Learning barycentric representations of 3d shapes for sketch-based 3d shape retrieval," in *CVPR*, 2017.
- [222] V. I. Bogachev and A. V. Kolesnikov, "The monge-kantorovich problem: achievements, connections, and perspectives," *Russian Mathematical Surveys*, 2012.
- [223] L. Wang, C. Qian, J. Wang, and Y. Fang, "Unsupervised learning of 3d model reconstruction from hand-drawn sketches," in *MM*, 2018.
- [224] Y. Shen, C. Zhang, H. Fu, K. Zhou, and Y. Zheng, "Deepsketchhair: Deep sketch-based 3d hair modeling," *arXiv preprint arXiv:1908.07198*, 2019.
- [225] H. Huang, E. Kalogerakis, E. Yumer, and R. Mech, "Shape synthesis from sketches via procedural models and convolutional networks," *TVCG*, 2016.
- [226] X. Han, C. Gao, and Y. Yu, "Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling," *TOG*, 2017.
- [227] M. Ye, S. Zhou, and H. Fu, "Deepshapesketch: Generating hand drawing sketches from 3d objects," in *IJCNN*, 2019.
- [228] J. P. Collomosse, G. McNeill, and Y. Qian, "Storyboard sketches for content based video retrieval," in *ICCV*, 2009.
- [229] S. Wu, H. Su, S. Zheng, H. Yang, and Q. Zhou, "Motion sketch based crowd video retrieval via motion structure coding," in *ICIP*, 2016.
- [230] S. Wu, H. Yang, S. Zheng, H. Su, Q. Zhou, and X. Lu, "Motion sketch based crowd video retrieval," *Multimedia Tools and Applications*, 2017.
- [231] F. Huang and J. F. Canny, "Sketchforme: Composing sketched scenes from text descriptions for interactive applications," *arXiv preprint arXiv:1904.04399*, 2019.
- [232] R. K. Sarvadevabhatla, S. Surya, T. Mittal, and R. V. Babu, "Game of sketches: Deep recurrent models of pictionary-style word guessing," in *AAAI*, 2018.
- [233] R. K. Sarvadevabhatla, S. Surya, T. Mittal, and V. B. Radhakrishnan, "Pictionary-style word-guessing on hand-drawn object sketches: dataset, analysis and deep network models," *TPAMI*, 2020.
- [234] K. Pang, D. Li, J. Song, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Deep factorised inverse-sketching," in *ECCV*, 2018.
- [235] P. Xu, H. Fu, Y. Zheng, K. Singh, H. Huang, and C.-L. Tai, "Model-guided 3d sketching," *TVCG*, 2018.
- [236] B. Jackson and D. F. Keefe, "Lift-off: Using reference imagery and freehand sketching to create 3d models in vr," *TVCG*, 2016.
- [237] D. Giunchi, D. Degraen, A. Steed *et al.*, "Mixing realities for sketch retrieval in virtual reality," *arXiv preprint arXiv:1910.11637*, 2019.
- [238] J. C. Roberts, C. Headleand, and P. D. Ritsos, "Sketching designs using the five design-sheet methodology," *TVCG*, 2015.
- [239] F. Boniardi, A. Valada, W. Burgard, and G. D. Tipaldi, "Autonomous indoor robot navigation using a sketch interface for drawing maps and routes," in *ICRA*, 2016.
- [240] V. Jain, P. Agrawal, S. Banga, R. Kapoor, and S. Gulyani, "Sketch2code: Transformation of sketches to ui in real-time using deep neural network," *arXiv preprint arXiv:1910.08930*, 2019.