

# FS-COCO: Towards Understanding of Freehand Sketches of Common Objects in Context

Pinaki Nath Chowdhury<sup>1, 2</sup> Aneeshan Sain<sup>1, 2</sup> Ayan Kumar Bhunia<sup>1</sup>  
Tao Xiang<sup>1, 2</sup> Yulia Gryaditskaya<sup>1, 3</sup> Yi-Zhe Song<sup>1, 2</sup>

<sup>1</sup>SketchX, CVSSP, University of Surrey, United Kingdom.

<sup>2</sup>iFlyTek-Surrey Joint Research Centre on Artificial Intelligence.

<sup>3</sup>Surrey Institute for People Centred AI, CVSSP, University of Surrey.

**Abstract.** We advance sketch research to scenes with the first dataset of freehand scene sketches, FS-COCO. With practical applications in mind, we collect sketches that convey scene content well but can be sketched within a few minutes by a person with any sketching skills. Our dataset comprises 10,000 freehand scene vector sketches with per point space-time information by 100 non-expert individuals, offering both object- and scene-level abstraction. Each sketch is augmented with its text description. Using our dataset, we study for the first time the problem of fine-grained image retrieval from freehand scene sketches and sketch captions. We draw insights on: (i) Scene salience encoded in sketches using the strokes temporal order; (ii) Performance comparison of image retrieval from a scene sketch and an image caption; (iii) Complementarity of information in sketches and image captions, as well as the potential benefit of combining the two modalities. In addition, we extend a popular vector sketch LSTM-based encoder to handle sketches with larger complexity than was supported by previous work. Namely, we propose a hierarchical sketch decoder, which we leverage at a sketch-specific “pre-text” task. Our dataset enables for the first time research on freehand scene sketch understanding and its practical applications. We release the dataset under CC BY-NC 4.0 license: [FS-COCO dataset](https://fscoco.github.io)<sup>1</sup>.

## 1 Introduction

As research on sketching thrives [21, 41, 16, 5], the focus shifts from an analysis of quick single-object sketches [7, 6, 40, 8] to an analysis of scene sketches [61, 17, 29, 12], and professional [19] or specialised [53] sketches. In the age of data-driven computing, conducting research on sketching requires representative datasets. For instance, the inception of object-level sketch datasets [21, 45, 58, 41, 16, 20] enabled and propelled research in diverse applications [5, 4, 13]. Recently, increasingly more attempts are conducted towards not only collecting the data but also understanding how humans sketch [20, 5, 22, 57, 54]. We extend these efforts to scene sketches by introducing FS-COCO (Freehand Sketches of Common Objects in COntext), the first dataset of 10,000 unique freehand scene sketches, drawn

---

<sup>1</sup> <https://fscoco.github.io>

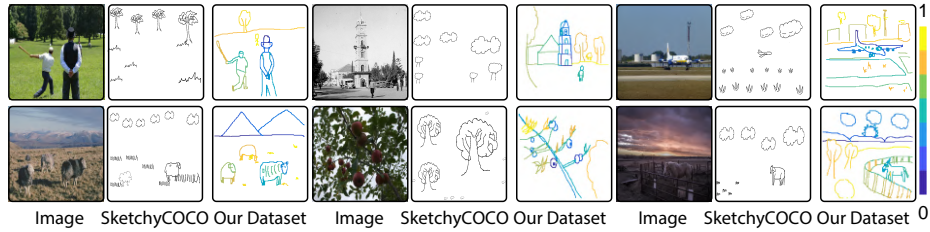


Fig. 1: Comparison of our sketches to the scene sketches from **SketchyCOCO**, the latter are obtained by combining together sketches of individual objects. Our freehand scene sketches contain abstraction at the object and scene level and better capture the content of reference scenes. This figure demonstrates a large domain gap between freehand scene sketches and available scene sketches, motivating the need for new datasets. Our sketches contain stroke temporal order information, which we visualize using the “Parula” color scheme: strokes in “blue” are drawn first, strokes in “yellow” are drawn last.

by 100 non-expert participants. We envision this dataset to permit a multitude of novel tasks and to contribute to the fundamental understanding of visual abstraction and expressivity in scene sketching. With our work, we make the first stab in this direction: We study fine-grained image retrieval from freehand scene sketches and the task of scene sketch captioning.

Thus far, research on scene sketches leveraged semi-synthetic [17, 29, 61] datasets that are obtained by combining together sketches and clip-arts of individual objects. Such datasets lack the holistic scene-level abstraction that characterises real scene sketches. Fig. 1 shows a visual comparison between the existing semi-synthetic [17] scene sketch dataset and ours FS-COCO. It shows interactions between scene elements in our sketches and diversity of objects depictions. Moreover, our sketches contain more object categories than previous datasets: Our sketches contain more than 92 categories from the COCO-stuff [9], while sketches in SketchyScene [61] and SketchyCOCO [17] contain 45 and 17 object categories, respectively.

Our dataset collection setup is practical applications-driven, such as the retrieval of a video frame given a quick sketch from memory. This is an important task because, while the text-based retrieval achieved impressive results in recent years, it might be easier to communicate via sketching fine-grained details. However, this will only be practical if users can provide a quick sketch and are not expected to be good sketchers. Therefore, we collect *easy to recognize but quick to create* freehand scene sketches from recollection (similar to object sketches collected previously [16, 41]). As reference images, we select photos from the MS-COCO [28], a benchmark dataset for scene understanding that ensures diversity of scenes and is complemented with rich annotations in a form of semantic segmentation and image captions.

Equipped with our FS-COCO dataset, we for the first time study the problem of a fine-grained image retrieval from freehand scene sketches. First, we show the presence of a domain gap between freehand sketches and semi-synthetic ones [61, 17], which are easier to collect, on the example of fine-grained sketch-

based image retrieval. Then, in our work we aim at understanding how scene-sketch-based retrieval compares to text-based retrieval, and what information sketch captures. To obtain a thorough understanding, we collect for each sketch its text description. The text description makes the subject who created the sketch, eliminating the noise due to sketch interpretation. By comparing sketch text descriptions with image text descriptions from the MS-COCO [28] dataset, we draw conclusions on the complementary nature of the two modalities: sketches and image text descriptions.

Our dataset of freehand scene sketches enables analysis towards insights into how humans sketch scenes, not possible with earlier datasets [17]. We continue the recent trend on understanding and leveraging strokes order [20, 5, 19, 54] and observe the same trends of coarse-to-fine sketching in scene sketches: We study stroke order as a factor of its salience for retrieval. Finally, we study sketch-captioning as an example of a sketch understanding task.

Collecting human sketches is costly, and despite our dataset being relatively large-scale, it is hard to reach the scale of the existing datasets of photos [47, 43, 33]. To tackle this known problem of sketch data, recent work [34, 4] to improve the performance of the encoder-decoder-based architectures on the downstream tasks proposed to pre-train the encoder relying on some auxiliary task. In our work, we build on [4] and consider the auxiliary task of raster sketch to vector sketch generation. Since our sketches are more complex than those of single objects considered before, we propose a dedicated hierarchical RNN decoder. We demonstrate the efficiency of the pre-training strategy and our proposed hierarchical decoder on fine-grained retrieval and sketch-captioning.

In summary, our contributions are: (1) We propose the first dataset of freehand scene sketches and their captions; (2) We study for the first time fine-grained freehand-scene-sketch-based image retrieval (3) and the relations between sketches, images and their captions. (4) Finally, to address the challenges of scaling sketch datasets and complexity of scene sketches, we introduce a novel hierarchical sketch decoder that exploit temporal stroke order available for our sketches. We leverage this decoder at the pre-training stage for fine-grained retrieval and sketch captioning.

## 2 Related Work

*Single-Object Sketch Datasets* Most freehand sketch datasets contain sketches of individual objects, annotated at the category level [16, 21] or part level [18], paired to photos [58, 41, 45] or 3D shapes [38]. Category-level and part-level annotations enable tasks such as sketch recognition [59, 42] and sketch generation [18, 5]. *Paired* datasets allow to study practical tasks such as sketch-based image retrieval [58] and sketch-based image generation [52].

However, collecting fine-grained paired datasets is time-consuming since one needs to ensure accurate, fine-grained matching while keeping the sketching task natural for the subjects [24]. Hence, such paired datasets typically contain a few thousand sketches per category, *e.g.*, QMUL-Chair-V2 [58] consists of 1432

Table 1: Properties of scene sketch datasets.

Dataset	Abstraction Object Scene	# pho- tos	Stroke temporal order	Cap- tions	Free- hand
SketchyScene [61]	✓	✗	7,264	✗	✗
SketchyCOCO [17]	✗	✓	14,081	✗	✗
<b>FS-COCO</b>	✓	✓	10,000	✓	✓

sketch-photo pairs on a single ‘chair’ category, Sketchy [41] has an average of 600 sketches per category, albeit over 125 categories.

Our dataset contains *10,000 scene sketches*, each paired with a ‘reference’ photo and text description. It contains scene sketches rather than sketches of individual objects and excels the existing fine-grained datasets of single-object sketches in the amount of paired instances.

*Scene Sketch Datasets* Probably the first dataset of 8,694 freehand scene sketches was collected within the multi-modal dataset [2]. It contains sketches of 205 scenes, but the examples are not paired between modalities. Scene sketch datasets with the pairing between modalities [61, 17] have started to appear, however they are ‘*semi-synthetic*’. Thus, the SketchyScene [61] dataset contains 7,264 sketch-image pairs. It is obtained by providing participants with a reference image and clip-art like object sketches to drag-and-drop for scene composition. The augmentation is performed by replacing object sketches with other sketch instances belonging to the same object category. SketchyCOCO [17] was generated automatically relying on the segmentation maps of photos from COCO-Stuff [9] and leveraging freehand sketches of single objects from [41, 16, 21].

Leveraging the semi-synthetic datasets, previous work studied scene sketch semantic segmentation [61], scene-level fine-grained sketch based image retrieval [29], and image generation [17]. Nevertheless, sketches in the existing datasets are not representative of freehand human sketches as shown in Fig. 1, and therefore the existing results can be only considered preliminary. Unlike existing semi-synthetic datasets, our dataset of freehand scene sketches captures abstraction at the object level and holistic scene level, and contains stroke temporal information. We provide a comparative statistics with previous datasets in Tab. 1, discussed in Sec. 4.1. We demonstrate the benefit and importance of the newly proposed data on two problems: image retrieval and sketch captioning.

### 3 Dataset Collection

Targeting practical applications, such as sketch-based image retrieval, we aimed to collect representative freehand scene sketches with object- and scene-levels of abstraction. Therefore, we define the following requirements towards collected sketches: (1) created by non-professionals, (2) fast to create, (3) recognizable, (4) paired with images, and (5) supplemented with sketch-captions.

*Data preparation* We randomly select 10k photos from MS-COCO [28], a standard benchmark dataset for scene understanding [39, 11, 10]. Each photo in this

dataset is accompanied by image captions [28] and semantic segmentation [9]. Our selected subset of photos includes 72 “*things*” instances (well-defined foreground objects) and 78 “*stuff*” instances (background instances with potentially no specific or distinctive spatial extent or shape: e.g., “trees”, “fence”), according to the classification introduced in [9]. We present detailed statistics in Sec. 4.1.

*Task* We built a custom web application<sup>2</sup> to engage 100 participants, each annotating a distinct subset of 100 photos. Our objective is to collect easy-to-recognize freehand scene sketches drawn from memory, alike single-object sketches collected previously [16, 41]. To imitate real world scenario of sketching from memory, following the practice of single object dataset collection, we showed a reference scene photo to a subject for a limited duration of 60 seconds, determined through a series of pilot studies. To ensure recognizable but not overly detailed drawings, we also put time limits on the duration of the sketching. We determined the optimal time limits through a series of pilot studies with 10 participants, which showed that 3 minutes were sufficient for participants to comfortably sketch recognizable scene sketches. We allow repeated sketching attempts, with the subject making an average of 1.7 attempts. Each attempt repeats the entire process of observing an image and drawing on a blank canvas. Upon satisfaction with their sketch, we ask the same subject to describe their sketch in text. The instructions to write a sketch caption are similar to that of Lin *et al.* [28] and are provided in supplemental materials. To reduce fatigue that can compromise data quality, we encourage participants to take frequent breaks and complete the task over multiple days. Thus, each participant spent 12 – 13 hours to annotate 100 photos over an average period of 2 days.

*Quality check* We check the quality of sketches. We hired as a *human judge* one appointed person (1) with experience in data collection and (2) non-expert in sketching. The human judge instructed to “mark sketches of scenes that are *too difficult to understand or recognize*.” The tagged photos were sent back to their assigned annotator. This process guarantees the resulting scene sketches are recognizable by a human, and therefore, should be understood by a machine.

*Participants* We recruited 100 non-artist participants from the age group 22–44, with an average age of 27.03, including 72 males and 28 females.

## 4 Dataset composition

Our dataset consists of 10,000 (a) unique freehand scene sketches, (b) textual descriptions of the sketches (sketch captions), (c) reference photos from the MSCOCO [28] dataset. Each photo in [28] contains 5 associated text descriptions (image captions) by different subjects [28]. Figs. 1 and 3 show samples from our dataset, and supplemental materials visualize more sketches from our dataset.

<sup>2</sup> <https://github.com/pinakinathc/SketchX-SST>

Table 2: Comparison of scene sketch datasets based on the distribution of categories in sketch-image pairs. ‘FG’ denotes subsets of datasets that are recommended for use in Fine-Grained tasks, such as fine-grained retrieval.  $e_l/e_c$  denotes estimates based on semantic segmentation labels in images and based on the occurrence of a word in a sketch caption, respectively. See Sec. 4 for details.

Dataset	# photos	# categories	# categories per sketch				# sketches per category			
			Mean	Std	Min	Max	Mean	Std	Min	Max
SketchyScene [61]	7,264	45	7.88	1.96	4	20	1079.76	1447.47	31	5723
SketchyCOCO [17]	14,081	17	3.33	0.9	2	7	1932.41	3493.01	33	9761
SketchyScene FG	2,724	45	7.71	1.88	4	20	394.51	540.30	3	2154
SketchyCOCO FG	1,225	17	3.28	0.89	2	6	164.71	297.79	5	824
FS-COCO ( $e_c$ )	10,000	92	1.37	0.57	1	5	99.42	172.88	1	866
FS-COCO ( $e_l$ )	10,000	150	7.17	3.27	1	25	413.18	973.59	1	6789

#### 4.1 Comparison to existing datasets

Tab. 2 provides comparison with previous dataset and statistics on distribution of object categories in our sketches, which we discuss in more detail below.

*Categories* First, we obtain a joint set of labels from the labels in [61, 17] and [9]. To compute statistics on the categories present in [61, 17], we use the semantic segmentation labels available in these datasets. For our dataset, we compute two estimates of the category distribution across our data: (1)  $e_l$ , based on semantic segmentation labels in images and (2)  $e_c$ , based on the occurrence of a word in a sketch caption. As can be seen from Fig. 3, the participants do not exhaustively describe in the caption all the objects present in sketches. Our dataset contains  $e_c/e_l = 92/150$  categories, which is more than double the number of categories in previous scene sketch datasets (Tab. 2). On average, each category is present in  $e_c/e_l = 99.42/413.18$  sketches. Among the most common category in all three datasets are ‘cloud’, ‘tree’ and ‘grass’ common to outdoor scenes. In our dataset ‘person’ is also among one of the most frequent categories along with common animals such as ‘horse’, ‘giraffe’, ‘dog’, ‘cow’ and ‘sheep’. Our dataset, according to lower/upper estimates, contains 33/71 indoor categories and 59/79 outdoor categories. We provide detailed statistics in supplemental materials.

*Sketch complexity* Existing datasets of freehand sketches [16, 41] contain sketches of single objects. The complexity of scene sketches is unavoidably higher than the one of single-object sketches. Sketches in our dataset have a median stroke count of 64. For comparison, a median strokes count in the popular Tu-Berlin [16] and Sketchy [41] datasets is 13 and 14, respectively.

## 5 Towards scene sketch understanding

### 5.1 Semi-synthetic versus freehand sketches

To study the domain gap between existing ‘semi-synthetic’ and our freehand scene sketches, we evaluate the state-of-the-art methods for Fine Grained Sketch Based Image Retrieval (FG-SBIR) on the three datasets: SketchyCOCO [17], SketchyScene [61] and FS-COCO (ours) (Tab. 3).

Table 3: Evaluation of a domain gap between ‘semi-synthetic’ sketches [61, 17] and freehand sketches FS-COCO. The details on the compared methods are in Sec. 5.1. Top-1/Top-10 accuracy (R@1/R@10) is the percentage of test sketches for which the ground-truth image is among the first 1/10 ranked retrieval results.

Methods	Trained On																	
	SketchyScene (S-Scene) [61]						SketchyCOCO (S-COCO) [17]						FS-COCO (Ours)					
	Evaluate on						Evaluate on						Evaluate on					
	S-Scene		S-COCO		FS-COCO		S-Scene		S-COCO		FS-COCO		S-Scene		S-COCO		FS-COCO	
	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10
Siam-VGG16 [58]	22.8	43.5	1.1	4.1	1.8	6.6	0.3	2.1	37.6	80.6	<0.1	0.4	5.8	24.5	2.4	11.6	23.3	52.6
HOLEF [46]	22.6	44.2	1.2	3.9	1.7	5.9	0.4	2.3	38.3	82.5	0.1	0.4	6.0	24.7	2.2	11.9	22.8	53.1
CLIP zero-shot [39]	1.26	9.70	—	—	—	—	—	—	1.85	9.41	—	—	—	—	—	—	1.17	6.07
CLIP*	8.6	24.8	1.7	6.6	2.5	8.2	1.3	5.1	15.3	43.9	0.6	3.1	1.6	11.9	2.6	12.5	5.5	26.5

*Methods and training details.* *Siam.-VGG16* adapts the pioneering method of Yu et al. [58] by replacing the Sketch-a-Net [59] feature extractor with VGG16 [44] trained using triplet loss [50, 55], as we observed that this increases retrieval performance. *HOLEF* [46] extends *Siam.-VGG16* by using spatial attention to better capture fine-scale details and introducing a novel trainable distance function in the context of triplet loss.

We also explore CLIP [39], a recent method that has shown an impressive ability to generalize across multiple photo datasets [28, 37]. *CLIP (zero-shot)* uses the pre-trained photo encoder, trained on 400 million text-photo pairs that do not include photos from the MS-COCO dataset. In our experiments, we use the publicly available ViT-B/32 version<sup>3</sup> of CLIP, which uses the visual transformer backbone as a feature extractor. Finally, *CLIP\** means CLIP fine-tuned on the target data. Since we found training CLIP to be very unstable, we train only the layer normalization [3] modules and add a fully connected layer to map the sketch and photo representations to a shared 512 dimensional feature space. We train *CLIP\** using triplet loss [50, 55] with a margin value set to 0.2 with a batch size 256 and a low learning rate of 0.000001.

*Train and test splits.* We train *Siam.-VGG16* and *HOLEF*, and fine-tune *CLIP\** on the sketches from one of three datasets: SketchyCOCO [17], SketchyScene [61] and FS-COCO. For our FS-COCO dataset 70% of each user sketches are used for training and the remaining 30% for testing. This results in a training/testing sets of 7,000 and 3,000 sketch-image pairs. For [17, 61] we use subsets of sketch-image pairs, since both datasets contain noisy data, which leads to performance degradation when used for the fine-grained tasks such as fine-grained retrieval. For SketchyCOCO [17], following Liu et al. [29], we sort the sketches based on the number of the foreground objects and select the top 1,225 scene sketch-photo pairs. We then randomly split those into training and test sets of 1,015 and 210 pairs, respectively. For SketchyScene [61] we follow their approach used to evaluate image retrieval, and manually select sketch-photo pairs that have same categories present in images and sketches. We obtain training and test sets of 2,472 and 252 pairs, respectively. The statistics on object categories in these subsets are given in Tab. 2 (‘FG’). Note that in each experiment, the image

<sup>3</sup> <https://github.com/openai/CLIP>



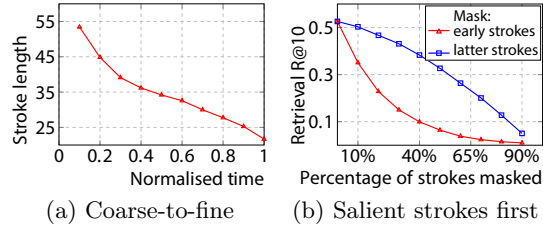


Fig. 2: Sketching strategies in our freehand scene sketches: Sec. 5.2. (a) Humans follow a coarse-to-fine sketching strategy, drawing longer strokes first. (b) Humans draw strokes more salient for the retrieval task early on. We plot the Top-10 (R@10) retrieval accuracy when certain strokes during testing are masked out. Top-10 accuracy calculates the percentage of test sketches for which the ground-truth image is among the first 10 ranked retrieval results.

gallery size is equal to the test set size. Therefore, in the case of our dataset, the retrieval is performed among the largest number of images.

*Evaluation.* Tab. 3 shows that training on ‘semi-synthetic’ sketch datasets like SketchyCOCO [17] and SketchyScene [61] does not generalize to freehand scene sketches from our dataset: training on FS-COCO / SketchyCOCO / SketchyScene and testing on our data results in  $R@1$  of 23.3 /  $< 0.1$  / 1.8. Training with the sketches from [61] rather than from [17] results in better performance on our sketches, probably due to the larger variety of categories in [61] (46 categories) than in [17] (17 categories). Tab. 3 also shows a large domain gap between all three datasets.

As the image gallery is larger when tested on our sketches than for other datasets, the performance on our sketches in Tab. 3 is lower, even when trained on our sketches. For a fairer comparison, we create 10 additional test sets consisting of 210 sketch-image pairs (the size of the SketchyCOCO dataset’s image gallery) by randomly selecting them from the initial set of 3000 sketches. For Siam-VGG16, the average retrieval accuracy and its standard deviation over ten splits are: Top-1 is  $50.39\% \pm 2.15\%$  and Top-10 is  $89.38\% \pm 2.0\%$ . For  $CLIP^*$ , the average retrieval accuracy and its standard deviation over ten splits are: Top-1 is  $42.53\% \pm 3.16\%$  and Top-10 is  $87.93\% \pm 2.14\%$ . These high performance numbers show the high quality of the sketches in our dataset.

## 5.2 What does a freehand sketch capture?

**Sketching strategy** We observe that humans follow a coarse-to-fine sketching strategy in scene sketches: in Fig. 2 (a) we show that the average stroke length decreases with time. Similarly, coarse-to-fine sketching strategies has previously been observed in single object sketch datasets [16, 41, 20, 54]. We also verify the hypothesis that humans draw salient and recognizable regions early [5, 16, 41]. We first train the classical SBIR method [58] on sketch-image pairs from our dataset: 70% of each user’s sketches are used for training and 30% for testing. During the evaluation, we follow two strategies: (i) We gradually mask out a



certain percentage of strokes drawn early, which is indicated by the red line in Fig. 2 (b). (ii) We then gradually mask out strokes drawn towards the end, which is indicated by the blue line in Fig. 2 (b). We observe that masking strokes towards the end has a smaller impact on the retrieval accuracy than masking early strokes. Thus we quantify that humans draw longer (Fig. 2a) and more salient for retrieval (Fig. 2b) strokes early on.

**Sketch captions vs. image captions** To gain insights into what information sketch captures, we compare sketch and image captions (Fig. 3 and 4). The vocabulary of our sketch captions matches 81.50% vocabulary of image captions. Specifically, comparing sketch and image captions for each instance reveals that on average 66.5% words in sketch captions are common with image captions, while 60.8% of words overlap among the 5 available captions of each image. This indicates that sketches preserve a large fraction of information in the image. However, the sketch captions in our dataset are on average shorter (6.55 words) than image captions (10.46). We explore this difference in more detail by visualizing the word clouds for sketch and image captions. From Fig. 4 we observe that, unlike image captions, sketch descriptions do not use “color” information. Also, we compute the percentage of nouns, verbs, and adjectives in sketch and image captions. Fig. 4(c) shows that our sketch captions are likely to focus more on objects (i.e., nouns like “horse”) and their actions (i.e., verbs like “standing”) instead of focusing on attributes (i.e., adjectives like “a brown horse”).

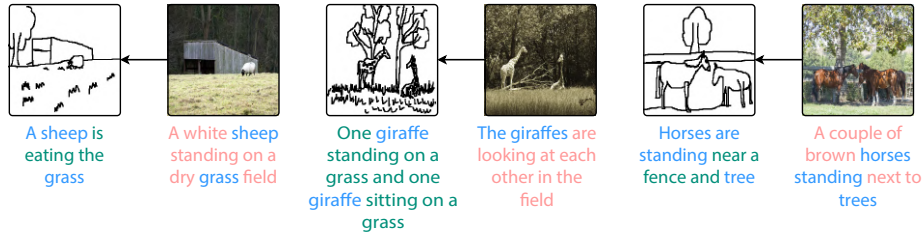


Fig. 3: A qualitative comparison of image and sketch captions. The overlapping words are marked in blue, the words present only in image-captions are marked in red, while the words present only in sketch-captions are marked in green.

**Freehand sketches vs. image captions** To understand the potential of quick freehand scene sketches in image retrieval, we compare freehand scene sketch with textual description as queries for fine-grained image retrieval (Tab. 4).

*Methods.* For text-based image retrieval, we evaluate two baselines: (1) *CNN-RNN* the simple and classic approach where text is encoded with an LSTM and images are encoded with a CNN encoder (VGG-16 in our implementation) [49, 25], and (2) CLIP [39] which is one of state-of-the-art methods alongside [26] in text-based image retrieval. For purity of experiments we evaluate here CLIP, as its training data did not include MS-COCO dataset from which the reference

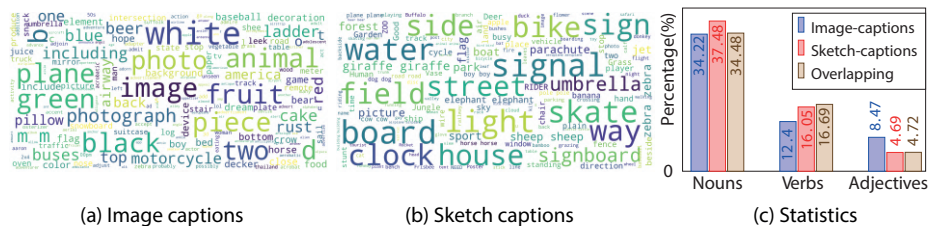


Fig. 4: (a,b) Word clouds show frequently occurring words in image and sketch captions, respectively. The larger the word, the more frequent it is. It shows that color information such as “white”, “green” is present in image captions but is missing from sketch captions. (c) Percentage of nouns, verbs, and adjectives in image and sketch captions, and their overlapping words.

images in our dataset are coming from. *CLIP zero-shot* uses off-the-shelf ViT-B/32 weights. *CLIP\** is fine-tuned on our sketch-captions by fine-tuning only layer normalization modules [3] with batch size 256 and learning rate  $1e-7$ .

*Training details.* *CNN-RNN* and *CLIP\** are trained with triplet loss [50, 55], with a margin value is set to 0.2. We use the same split to train/test sets as in Sec. 5.1. For retrieval from image captions, we randomly select one of 5 available caption versions.

*Evaluation.* Tab. 4 shows that image captions result in better retrieval performance compared to sketch captions, which we attribute to the color information in image captions. However, we observe that *CLIP\**-based retrieval from image captions is slightly inferior to *Siam.-VGG16*-based retrieval from sketches. Note that *CLIP\** is pre-trained on 400 million text-photo pairs, while *Siam.-VGG16* was trained on a much smaller set of 7000 sketch-photo pairs. Therefore, with even larger sketch datasets the retrieval accuracy from sketches will further increase. There is an intuitive explanation for this since scene sketches intrinsically encode fine-grained visual cues that are difficult to convey in text.

Table 4: Text-based versus sketch-based image retrieval.

Methods	Retrieval accuracy					
	Image Captions		Sketch Captions		Sketches	
	R@1	R@10	R@1	R@10	R@1	R@10
Siam.-VGG16 [58]	—	—	—	—	<b>23.3</b>	<b>52.6</b>
CNN-RNN [45]	11.1	31.1	7.2	23.6	—	—
CLIP zero-shot [39]	21.0	50.9	11.5	35.3	1.17	6.07
CLIP*	<u>22.1</u>	<u>52.3</u>	14.8	36.6	5.5	26.5

**Text and sketch synergy** While we have shown that scene sketches have strong ability in expressing fine-grained visual cues, image captions convey additional information such as “color”. Therefore, we are exploring whether the

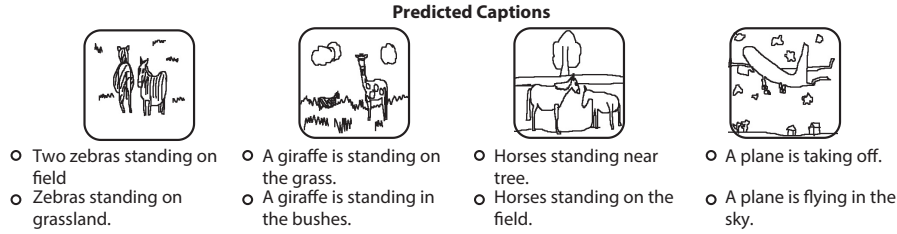


Fig. 5: Qualitative results showing predicted captions from LNFMM (H-Decoder) for scene sketches from our dataset.

two query modalities combined can improve fine-grained image retrieval. Following [30], we use two simple approaches to combine sketch and text: (-concat) we concatenate sketch and text features and (-add) we add sketch and text features. The combined features are then passed through a fully connected layer. Comparing the results in Tab. 5 and Tab. 4 shows that combining image captions and scene sketches improves fine-grained image retrieval. This confirms that the scene sketch complements the information conveyed by the text.

Table 5: Fine-grained image retrieval from the combined input of scene sketches and textual image descriptions.

Methods		R@1	R@10	Methods	R@1	R@10
CNN-RNN [45]	-add	<b>25.3</b>	<b>55.0</b>	CLIP* -add	23.9	53.5
CNN-RNN [45]	-concat	24.3	53.9	CLIP* -concat	23.3	52.6

Table 6: Sketch captioning (Sec. 5.3): our dataset enables captioning of scene sketches. We provide the results of the popular captioning methods developed for photos. For the evaluation, we use the standard metrics: BELU (B4) [35], METEOR (M) [14], ROUGE (R) [27], CIDEr (C) [48], SPICE (S) [1].

Methods	B4	M	R	C	S
Xu <i>et al.</i> [56]	13.7	17.1	44.9	69.4	14.5
AG-CVAE [51]	16.0	18.9	49.1	80.5	15.8
LNFMM [31]	16.7	21.0	52.9	90.1	16.0
LNFMM with pre-training (H-Decoder)	<b>17.3</b>	<b>21.1</b>	<b>53.2</b>	<b>95.3</b>	<b>17.2</b>

### 5.3 Sketch Captioning

While scene sketches are a pre-historic form of human communication, scene sketch understanding is nascent. Existing literature has solidified captioning as a hallmark task for scene understanding. The lack of paired scene-sketch and text datasets is the biggest bottleneck. Our dataset allows us to study this problem for the first time. We evaluate several popular and SOTA methods in Tab. 6: Xu *et al.* [56] is one of the first popular works to use the attention mechanism with an LSTM for image captioning. AG-CVAE [50] is a SOTA image captioning

model that uses a variational auto-encoder along with an additive gaussian prior. Finally, LNFMM [31] is a recent SOTA approach using normalizing flows [15] to capture the complex joint distribution of photos and text. We show qualitative results in Fig. 5 using the LNFMM model with the pre-training strategy we introduce in Sec. 6.

## 6 Efficient “pretext” task

Our dataset is large (10,000 scene sketches!) for a sketch dataset. However, scaling it up to millions of sketch instances paired with other modalities (photos/text) to match the size of the photo datasets [47] might be intractable in the short term. Therefore, when working with freehand sketches, it is important to find ways to go around the limited dataset size. One traditional approach to address this problem is to solve an auxiliary or “pretext” task [60,36,32]. Such tasks exploit self-supervised learning, allowing to pre-train the encoder for the ‘source’ domain leveraging unpaired/unlabeled data. In the context of sketching, solving jigsaw puzzles [34] and converting raster to vector sketch [4] “pretext” tasks were considered. We extend the state-of-the-art sketch-vectorization [4] “pretext” task to support the complexity of scene sketches, exploiting the availability of time-space information in our dataset. We pre-train a raster sketch encoder with the newly proposed decoder that reconstructs a sketch in a vector format as a sequence of stroke points. Previous work [4] leverages a single layer Recurrent Neural Network (RNN) for sketch decoding. However, it can only reliably model up to around 200 stroke points [21], while our scene sketches can contain more than 3000 stroke points, which makes modeling scene sketches challenging. We observe that, on average, scene sketches consist of only 74.3 strokes, with each stroke containing around 41.1 stroke points. Modeling such number of strokes or stroke points *individually* is possible using a standard LSTM network [23]. Therefore, we propose a novel 2-layered hierarchical LSTM decoder.

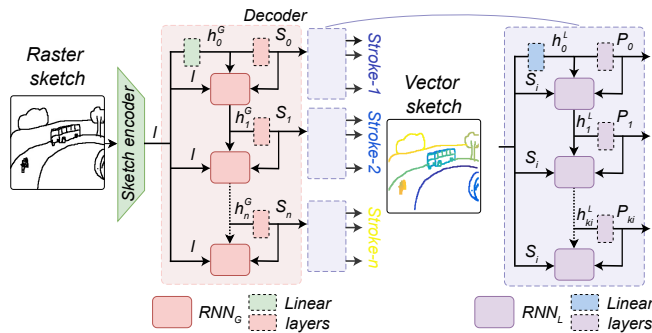


Fig. 6: The proposed hierarchical decoder used for pre-training a sketch encoder.

### 6.1 Proposed Hierarchical Decoder (H-Decoder)

We denote a raster sketch encoder that our proposed decoder pre-trains as  $E(\cdot)$ . Let the output feature map of  $E(\cdot)$  be  $F \in \mathbb{R}^{h' \times w' \times c}$ , where  $h'$ ,  $w'$  and  $c$  denotes height, width, and number of channels, respectively. We apply a global max pooling to  $F$ , with consequent flattening, to obtain a latent vector representation of the raster sketch,  $l_R \in \mathbb{R}^{512}$ .

Naively decoding  $l_R$  using a single layer RNN is intractable [21]. We propose a two-level decoder consisting of two LSTMs, referred to as global and local. The global LSTM ( $\text{RNN}_G$ ) predicts a sequence of feature vectors, each representing a stroke. The second local LSTM ( $\text{RNN}_L$ ) predicts a sequence of points for any stroke, given its predicted feature vector.

We initialize the hidden state of the global  $\text{RNN}_G$  using a linear embedding as follows:  $h_0^G = W_h^G l_R + b_h^G$ . The hidden state  $h_i^G$  of decoder  $\text{RNN}_G$  is updated as follows:  $h_i^G = \text{RNN}_G(h_{i-1}^G; [l_R, S_{i-1}])$ , where  $[\cdot]$  stands for a concatenation operation and  $S_{i-1} \in \mathbb{R}^{512}$  is the last predicted stroke representation computed as:  $S_i = W_y^G h_i^G + b_y^G$ .

Given each stroke representation  $S_i$ , the initial hidden state of local  $\text{RNN}_L$  is obtained as:  $h_0^L = W_h^L S_i + b_h^L$ . Next,  $h_j^L$  is updated as:  $h_j^L = \text{RNN}_L(h_{j-1}^L; [S_i, P_{t-1}])$ , where  $P_{t-1}$  is the last predicted point of the  $i$ -th stroke. A linear layer is used to predict a point:  $P_t = W_y^L h_j^L + b_y^L$ , where  $P_t = (x_t, y_t, q_t^1, q_t^2, q_t^3)$  is of size  $\mathbb{R}^{2+3}$  whose first two logits represent absolute coordinate  $(x, y)$ , and the later three denote the pen's state  $(q_t^1, q_t^2, q_t^3)$  [21].

We supervise the prediction of the absolute coordinate and pen state using the mean-squared error and categorical cross-entropy loss, as in [4].

### 6.2 Evaluation & Discussion

We use our proposed H-Decoder for pre-training a raster sketch encoder for fine-grained image retrieval (Tab. 7) and sketch captioning (Tab. 6).

*Training details* We start pre-training VGG-16 based *Siam.VGG16* (Tab. 7) and *LNFMM* (Tab. 6) encoders on QuickDraw [21], a large dataset of freehand object sketches, by coupling a VGG16 raster sketch encoder with our H-Decoder. For *CLIP\** we start from the model weights in ViT-B/32. We then train *CLIP\** and VGG-16-based encoders with our “pretext” task on *all* sketches from our dataset. We exploit here that the test data is available but does not have the paired data – captions, photos. After pre-training, training for downstream tasks starts with the weights learned during pre-training.

*Evaluation* Tab. 6 shows the benefit of the pre-training with the proposed decoder. With this pre-training strategy the performance of LNFMM [31] on sketches approaches the performance on images (CIDEr score of 98.4<sup>4</sup>), increasing, *e.g.*, the CIDEr score from 90.1 to 95.3.

<sup>4</sup> The performance of image captioning goes up to 170.5 when 100 generated captions are evaluated against the ground-truth instead of 1

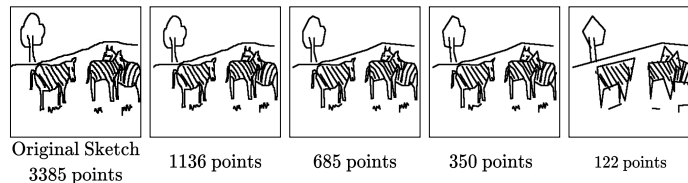


Fig. 7: Simplifying scene sketch with the RDP algorithm loses salient information. RNNs can reliably model around 200 points. The training of a single-layer RNN exploits the simplification level of the most right image.

Table 7: The role of pre-training with H-Decode in retrieval.

Method	Baseline		H-Decoder	
	R@1	R@10	R@1	R@10
Siam.-VGG16	23.3	52.6	<b>24.1</b>	<b>54.3</b>
CLIP*	5.5	26.5	5.7	27.1

This pre-training also slightly improves the performance of sketch-based retrieval (Tab. 7). Next, we compare pre-training with the proposed H-Decoder and a more naive approach. We simplify scene sketches with the Ramer-Douglas Peucker (RDP) algorithm (Fig. 7): On average, the simplified sketches contain 165 stroke points, while the original sketches contain 2437 stroke points. Then, we pre-train with a single layer RNN, as proposed in [4]. In this case *Siam.VGG16* achieves  $R@10$  of 52.1, which is lower than the performance without pre-training (Tab. 7). This further demonstrates the importance of the proposed hierarchical decoder to scene sketches.

## 7 Conclusion

We introduce the first dataset of freehand scene sketches with fine-grained paired text information. With the dataset, we took the first step towards freehand scene sketch understanding, studying tasks such as fine-grained image retrieval from scene sketches and scene sketches captioning. We show that relying on off-the-shelf methods and our data promising image retrieval and sketch captioning accuracy can be obtained. We hope that future work will leverage our findings to design dedicated methods exploiting the complementary information in sketches and image captions. In the supplemental materials, we provide a thorough comparison of modern encoders and state-of-the-art methods, and show how meta-learning can be used for few-shot sketch adaptation to an unseen user style. Finally, we proposed a new RNN-based decoder that exploits time-space information embedded in our sketches for a ‘pre-text’ task, demonstrating substantial improvement on sketch-captioning. We hope that our dataset will promote research on image generation from freehand scene sketches, sketch captioning, and novel sketch encoding approaches that are well suited for the complexity of freehand scene sketches.

## References

1. Anderson, P., Fernando, B., Johnson, M., Gould, S.: Spice: semantic propositional image caption evaluation. In: ECCV (2016) [11](#)
2. Aytar, Y., Castrejon, L., Vondrick, C., Pirsavash, H., Torralba, A.: Cross-modal scene networks. IEEE-TPAMI (2018) [4](#)
3. Ba, J., Kiros, J.R., Hinton, G.E.: Layer normalization. In: NIPS Deep Learning Symposium (2016) [7](#), [10](#)
4. Bhunia, A.K., Chowdhury, P.N., Yang, Y., Hospedales, T.M., Xiang, T., Song, Y.Z.: Vectorization and rasterization: Self-supervised learning for sketch and hand-writing. In: CVPR (2021) [1](#), [3](#), [12](#), [13](#), [14](#)
5. Bhunia, A.K., Das, A., Riaz Muhammad, U., Yang, Y., Hospedales, T.M., Xiang, T., Gryaditskaya, Y., Song, Y.Z.: Pixelor: A competitive sketching ai agent. so you think you can beat me? In: SIGGRAPH Asia (2020) [1](#), [3](#), [8](#)
6. Bhunia, A.K., Gajjala, V.R., Koley, S., Kundu, R., Sain, A., Xiang, T., Song, Y.Z.: Doodle it yourself: Class incremental learning by drawing a few sketches. In: CVPR (2022) [1](#)
7. Bhunia, A.K., Koley, S., Khilji, A.F.U.R., Sain, A., Chowdhury, P.N., Xiang, T., Song, Y.Z.: Sketching without worrying: Noise-tolerant sketch-based image retrieval. In: CVPR (2022) [1](#)
8. Bhunia, A.K., Sain, A., Shah, P., Gupta, A., Chowdhury, P.N., Xiang, T., Song, Y.Z.: Adaptive fine-grained sketch-based image retrieval. In: ECCV (2022) [1](#)
9. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: CVPR (2018) [2](#), [4](#), [5](#), [6](#)
10. Chen, J., Guo, H., Yi, K., Li, B., Elhoseiny, M.: Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. arXiv preprint arXiv:2102.10407 (2021) [4](#)
11. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915 (2016) [4](#)
12. Chowdhury, P.N., Bhunia, A.K., Gajjala, V.R., Sain, A., Xiang, T., Song, Y.Z.: Partially does it: Towards scene-level fg-sbir with partial input. In: CVPR (2022) [1](#)
13. Das, A., Yang, Y., Hospedales, T., Xiang, T., Song, Y.Z.: Béziersketch: A generative model for scalable vector sketches. In: ECCV (2020) [1](#)
14. Denkowski, M.J., Lavie, A.: Meteor universal: Language specific translation evaluation for any target language. In: WMT@ACL (2014) [11](#)
15. Dinh, L., Krueger, D., Bengio, Y.: Nice: non-linear independent components estimation. In: ICLR, Workshop Track Proc (2015) [12](#)
16. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Trans. Graph. (2012) [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [8](#)
17. Gao, C., Liu, Q., Wang, L., Liu, J., Zou, C.: Sketchycoco: Image generation from freehand scene sketches. In: CVPR (2020) [1](#), [2](#), [3](#), [4](#), [6](#), [7](#), [8](#)
18. Ge, S., Goswami, V., Zitnick, C.L., Parikh, D.: Creative sketch generation. In: ICLR (2021) [3](#)
19. Gryaditskaya, Y., Hähnlein, F., Liu, C., Sheffer, A., Bousseau: Lifting freehand concept sketches into 3d. In: SIGGRAPH Asia (2020) [1](#), [3](#)
20. Gryaditskaya, Y., Sypesteyn, M., Hoftijzer, J.W., Pont, S., Durand, F., Bousseau, A.: Opensketch: a richly-annotated dataset of product design sketches. ACM Trans. Graph. (2019) [1](#), [3](#), [8](#)



21. Ha, D., Eck, D.: A neural representation of sketch drawings. In: ICLR (2018) [1](#), [3](#), [4](#), [12](#), [13](#)
22. Hertzmann, A.: Why do line drawings work? *Perception* (2020) [1](#)
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* (1997) [12](#)
24. Holinaty, J., Jacobson, A., Chevalier, F.: Supporting reference imagery for digital drawing. In: ICCV Workshop (2021) [3](#)
25. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. *IEEE-TPAMI* (2017) [9](#)
26. Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., Gao, J.: Oscar: Object-semantics aligned pre-training for vision-language tasks. In: ECCV (2020) [9](#)
27. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: *Text Summarization Branches Out* (2004) [11](#)
28. Lin, T.Y., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: ECCV (2014) [2](#), [3](#), [4](#), [5](#), [7](#)
29. Liu, F., Zhou, C., Deng, X., Zuo, R., Lai, Y.K., Ma, C., Liu, Y.J., Wang, H.: Scenesketcher: Fine-grained image retrieval with scene sketches. In: ECCV (2020) [1](#), [2](#), [4](#), [7](#)
30. Liu, K., Li, Y., Xu, N., Natarajan, P.: Learn to combine modalities in multimodal deep learning. *arXiv preprint arXiv:1805.11730* (2018) [11](#)
31. Mahajan, S., Gurevych, I., Roth, S.: Latent normalizing flows for many-to-many cross-domain mappings. In: ICLR (2020) [11](#), [12](#), [13](#)
32. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016) [12](#)
33. Ordonez, V., Kulkarni, G., Berg, T.: Im2text: Describing images using 1 million captioned photographs. In: NIPS (2011) [3](#)
34. Pang, K., Yang, Y., Hospedales, T.M., Xiang, T., Song, Y.Z.: Solving mixed-modal jigsaw puzzle for fine-grained sketch-based image retrieval. In: CVPR (2020) [3](#), [12](#)
35. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL (2002) [11](#)
36. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR (2016) [12](#)
37. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: ICCV (2015) [7](#)
38. Qi, A., Gryaditskaya, Y., Song, J., Yang, Y., Qi, Y., Hospedales, T.M., Xiang, T., Song, Y.Z.: Toward fine-grained sketch-based 3d shape retrieval. *IEEE-TIP* (2021) [3](#)
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020* (2021) [4](#), [7](#), [9](#), [10](#)
40. Sain, A., Bhunia, A.K., Potlapalli, V., Chowdhury, P.N., Xiang, T., Song, Y.Z.: Sketch3t: Test-time training for zero-shot sbir. In: CVPR (2022) [1](#)
41. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Trans. Graph.* (2016) [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [8](#)
42. Schneider, R.G., Tuytelaars, T.: Sketch classification and classification-driven analysis using fisher vectors. In: SIGGRAPH Asia (2014) [3](#)

43. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL (2018) [3](#)
44. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015) [7](#)
45. Song, J., Song, Y.Z., Xiang, T., Hospedales, T.M.: Fine-grained image retrieval: the text/sketch input dilemma. In: BMVC (2017) [1](#), [3](#), [10](#), [11](#)
46. Song, J., Yu, Q., Song, Y.Z., Xiang, T., Hospedales, T.M.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: ICCV (2017) [7](#)
47. Srinivasan, K., Raman, K., Chen, J., Bendersky, M., Najork, M.: Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. arXiv preprint arXiv:2103.01913 (2021) [3](#), [12](#)
48. Vedantam, R., Zitnick, C.L., Parikh, D.: Cider: Consensus-based image description evaluation. In: CVPR (2015) [11](#)
49. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: CVPR (2015) [9](#)
50. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: CVPR (2014) [7](#), [10](#), [11](#)
51. Wang, L., Schwing, A.G., Lazebnik, S.: Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In: NeurIPS (2017) [11](#)
52. Wang, S.Y., Bau, D., Zhu, J.Y.: Sketch your own gan. In: ICCV (2021) [3](#)
53. Wang, T.Y., Ceylan, D., Popovic, J., Mitra, N.J.: Learning a shared shape space for multimodal garment design. In: SIGGRAPH Asia (2018) [1](#)
54. Wang, Z., Qiu, S., Feng, N., Rushmeier, H., McMillan, L., Dorsey, J.: Tracing versus freehand for evaluating computer-generated drawings. ACM Trans. Graph. (2021) [1](#), [3](#), [8](#)
55. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: ECCV (2016) [7](#), [10](#)
56. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML (2015) [11](#)
57. Yan, C., Vanderhaeghe, D., Gingold, Y.: A benchmark for rough sketch cleanup. ACM Trans. Graph. (2020) [1](#)
58. Yu, Q., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: CVPR (2016) [1](#), [3](#), [7](#), [8](#), [10](#)
59. Yu, Q., Yang, Y., Song, Y.Z., Xiang, T., Hospedales, T.: Sketch-a-net that beats humans. In: BMVC (2015) [3](#), [7](#)
60. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016) [12](#)
61. Zou, C., Yu, Q., Du, R., Mo, H., Song, Y.Z., Xiang, T., Gao, C., Chen, B., Zhang, H.: Sketchyscene: Rickly-annotated scene sketches. In: ECCV (2018) [1](#), [2](#), [4](#), [6](#), [7](#), [8](#)