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

Pinaki Nath Chowdhury[1,2] Aneeshan Sain[1,2] Yulia Gryaditskaya[1,2,3] Ayan Kumar Bhunia[1]
Tao Xiang[1,2] Yi-Zhe Song[1,2]

[1] University of Surrey, CVSSP, United Kingdom
[2] iFlyTek-Surrey Joint Research Centre on Artificial Intelligence
[3] Surrey Institute for People-Centred AI, UK

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 well scene content 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 the fine-grained image retrieval from freehand scene sketches and sketch captions. We draw insights on: (i) Scene salience encoded in sketches with strokes temporal order; (ii) The retrieval performance accuracy from scene sketches against image captions; (iii) Complementarity of information in sketches and image captions, as well as the potential benefit of combining the two modalities. In addition, we propose new solutions enabled by our dataset: (i) We adopt meta-learning to show how the retrieval model can be fine-tuned to a new user style given just a small set of sketches, (ii) 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 "pretext" task. Our dataset enables for the first time research on freehand scene sketch understanding and its practical applications.*

1. Introduction

As research on sketching thrives [6, 14, 20, 42], the focus shifts from an analysis of quick single-object sketches to an analysis of scene sketches [16, 29, 63], and professional [18] or specialised [55] sketches. In the age of data-driven computing, conducting research on sketching requires representative datasets. For instance, the inception of object-level sketch datasets [14, 19, 20, 42, 46, 60] enabled and propelled research in diverse applications [5, 6, 10]. Recently, increasingly more attempts are conducted towards not only collecting the data but also understanding how humans sketch [6, 19, 21, 56, 59]. We extend these efforts to scene sketches by introducing FS-COCO, the first dataset of $10,000$ unique freehand scene sketches, drawn 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.

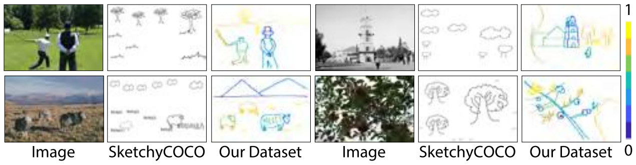


Figure 1. Existing datasets of scene sketches `SketchyScene` [63] and `SketchyCOCO` [16] are obtained by combining together clip-arts and sketches of individual objects, respectively. In this figure, we compare our sketches to scene sketches from `SketchyCOCO`. It can be observed that our freehand scene sketches exhibit object and scene level abstraction, and better capture the content of reference scenes. Moreover, the sketches in our dataset contain temporal strokes order information. We visualize the strokes order using the "Parula" color scheme: strokes in "blue" are drawn first, strokes in "yellow" are drawn last.

Thus far, research on scene sketches leveraged semi-synthetic [16, 29, 63] 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 [16] 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 95 categories from the COCO-stuff [7], while sketches in SketchyScene [63] and SketchyCOCO [16] 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. Therefore we collect *easy to recognize but quick to create* freehand scene sketches from recollection (similar to object sketches collected previously [14, 42]). 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 study how indicative is the performance observed through training and testing on semi-synthetic datasets [16, 63], that are easier to collect, of the performance on freehand sketches. 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 [16]. First, the diversity of user styles and large number of sketch instances for each of participants allows us to demonstrate the potential of personalization in sketch-related tasks. In particular, we adopt meta-learning [2, 15] to increase the accuracy of the retrieval for a particular subject given a few sketch examples, capturing abstraction specific to this subject. Second, we continue the recent trend on understanding and leveraging strokes order [6, 18, 19, 56] and observe the same trends of coarse-to-fine sketching in scene sketches. Notably, we study salience of early versus latter strokes for the retrieval task. Finally, we study the task of sketch-captioning as an example of a sketch understanding task.

Collecting human sketches is costly, and despite our dataset is relatively large-scale, it is hard to reach the scale of the existing datasets of photos [34, 44, 48]. To tackle this known problem of sketch data, recent work [5, 35] 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. They showed that this strategy is beneficial over pre-training on photos. Last but not least, in our work we build on [5] 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 the fine-grained retrieval and sketch-captioning tasks.

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) Contributing to sketch understanding, we study personalization with meta-learning, strokes salience and sketch-captioning tasks. (5) Finally, to address the challenges of scaling sketch datasets and complexity of scene sketches, we introduce a novel hierarchical sketch decoder. We leverage this decoder at the pre-training stage for the fine-grained retrieval and sketch captioning tasks.

2. Related Work

Single-Object Sketch Datasets Most freehand sketch datasets contain sketches of individual objects, annotated at the category level [14, 20] or part level [17], or have paired photos [42, 46, 60] or 3D shapes [39]. Category-level and part-level annotations enable tasks such as sketch recognition [43, 61] and sketch generation [6, 17]. *Paired* datasets allow to study practical tasks such as sketch-based image retrieval [60] and sketch-based image generation [54].

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 [23]. Hence, such paired datasets typically contain a few thousand sketches per category, *e.g*., QMUL-Chair-V2 [60] consists of 1432 sketch-photo pairs on a single 'chair' category, Sketchy [42] has an average of 600 sketches per category, albeit over 125 categories.

In contrast, our dataset contains *10,000 scene* sketches, each paired with a 'reference' photo and text description. It contains scene sketches that are more challenging to collect and excels the existing fine-grained datasets of single-object sketches in the amount of paired instances. It will foster research on scene-sketch-understanding, retrieval, and diverse generative tasks (*e.g*., sketch-based image generation).

Scene Sketch Datasets Probably the first dataset of 8,694 freehand scene sketches was collected within the multi-model dataset [3]. It contains sketches of 205 scenes, but the examples are not paired between modalities. Scene sketch datasets with the pairing between modalities [16, 63] have started to appear, however they are *'semi-synthetic'*. Thus, the SketchyScene [63] dataset contains $7,264$ sketch-image

Table 1. Comparison of scene sketch datasets: `SketchyScene` [63], `SketchyCOCO` [16] and FS-COCO. We use the semantic segmentation labels to compute the statistics on categories present in [16,63]. For our dataset, we look for the occurrence of one of COCO-Stuff [7] categories in sketch captions. Our dataset is well-balanced, 95 categories are uniformly distributed among sketches with the lowest standard deviation among the datasets.

Dataset	Abstraction		# photos	Vector Sketch	Captions	Freehand	#categories	# categories per sketch				# sketches per category			
	Object	Scene						Mean	Std	Min	Max	Mean	Std	Min	Max
`SketchyScene`	✓	✗	7,264	✗	✗	✗	45	7.88	1.96	4	20	1087.02	1416.49	31	5723
`SketchyCOCO`	✗	✓	14,081	✗	✗	✗	17	3.33	0.9	2	7	1932.41	3388.72	33	9761
FS-COCO (ours)	✓	✓	10,000	✓	✓	✓	95	1.37	0.57	1	5	89.72	159.9	1	853

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 [16] was generated automatically relying on the segmentation maps of photos from COCO-Stuff [7] and leveraging freehand sketches of single objects from [14,20,42]. Unlike existing semi-synthetic datasets, our dataset of freehand scene sketches captures abstraction on the intrinsic object level and holistic scene level (Tab. 1).

Sketch Understanding The lack of freehand scene sketch datasets severely limits the proliferation of research on scene sketch understanding. Nevertheless, leveraging the semi-synthetic datasets previous work studied scene sketch semantic segmentation [63], scene-level fine-grained sketch based image retrieval [29], and image generation [16]. Uniquely, our goal is to enable the analysis of *freehand* scene sketches. Following the footstep of understanding scenes in photos [31,53,58], we study sketch captioning for the first time using the paired sketch-text instances available in our dataset.

3. Dataset Collection

Targeting practical applications, we aimed to collect 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 [8, 9, 41]. Each photo is accompanied by image captions [28] and semantic segmentation [7]. 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 [7].

Task We built a custom web application[1] 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 [14,42]. 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. To ensure recognizable, but not overly detailed drawings, we also imposed time constraints 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. Repeated sketching trials were allowed, with a subject making 1.7 trials on average. Each trial repeats the entire process of viewing and sketching from a blank canvas. To reduce fatigue that can compromise data quality, we encourage participants to take frequent breaks and complete the task over multiple days. Hence, each participant spent $12-13$ hours to annotate 100 photos over an average period of 2 days.

Upon satisfaction with their sketch, we ask the same subject to describe their sketch in text, where instructions[2] to write a sketch caption are similar to that of Lin *et al.* [28].

Finally, to perform sketches quality check, we hired one dedicated person as a *human judge* (1) with experience in data collection and (2) non-expert in sketching. A human judge was tasked with the instruction: *"flag scene sketches that are too difficult to understand or recognize"*. The *flagged* photos were sent back to their assigned annotator. This process guarantees the resulting scene sketches are recognizable by a human, and hence in principle, should be understood by a machine.

Participants We recruited 100 participants, non-experts in sketching, from the age group $22-44$, with the mean age of 27.03, comprising 72 males and 28 females[3].

Dataset composition Our dataset consists of $10,000$ (a) unique freehand scene sketches, (b) textual descriptions of

[1] We will release the code for the annotation tool and backend-server script upon acceptance.

[2] We provide the detailed instructions in the Supplemental.

[3] See supplemental for details on how consent was obtained.

Table 2. Evaluation of a domain gap between 'semi-synthetic' sketches [16, 63] and freehand sketches FS-COCO. We show a quantitative comparison for fine-grained scene sketch-based image retrieval using some of the most popular and latest methods, see Sec. 4.1 for the details. Top-1/Top-10 accuracy (R@1/R@10) calculates the percentage of test sketches for which the ground-truth image is among the first 1/10 ranked retrieval results. For SketchyCOCO [16], following Liu *et al.* [29], we adopt the standard 1015/210 train/test split, where each scene sketch contains at least one *foreground* object. For SketchyScene [63], we use the standard train/test split of 2472/252 sketch-photo pairs. For our dataset FS-COCO 70% of each user sketches are used for training and the remaining 30% for testing. This results in 7000/3000 train/test split. In each experiment the image gallery size is equal to the test size. Note how training on 'semi-synthetic' datasets (`SketchyScene`, `SketchyCOCO`) generalizes poorly to freehand sketches FS-COCO, limiting its practical usage.

Methods	Trained On																	
	SketchyScene (S-Scene) [63]						SketchyCOCO (S-COCO) [16]						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 [60]	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 [47]	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 [41]	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

the sketches (sketch captions), (c) reference photos from the MS-COCO [28] dataset. Each photo in [28] contains 5 associated text descriptions (image captions) by different subjects [28]. Fig. S9 provides some examples, while Tab. 1 provides comparison with previous dataset and statistics on distribution of object categories in our sketches. The supplemental provides additional sketch examples and detailed statistics on object categories.

4. Towards scene sketch understanding

In this section, we aim at a deeper understanding of freehand scene sketches. In particular, (i) first, we analysis of the performance of fine-grained image retrieval from scene freehand and 'semi-synthetic' sketches from [16, 63]. (ii) Then, we analyze the complexity of our sketches and strokes usage over time. (iii) Next, we drive out insights on how well a sketch describe a reference image. (iv) We then evaluate a sketch against a text description and explore the role of sketches and text in fine-grained image retrieval: Given the same amount of training data, is scene sketch or text a better alternative as a query? (v) Moreover, we analyze the potential synergy between image captions and sketches. (vi) Importantly, for the first time, we study the problem of sketch captioning – an important task towards sketch-understanding. (vii) We conclude this section by analyzing the generalization of scene Sketch-Based Image Retrieval (SBIR) to the sketches by 'unseen' participants. We show how meta-learning can be used to adapt a generic model to a user-specific model – thus motivating future sketch research in personalized AI.

4.1. Fine-grained retrieval

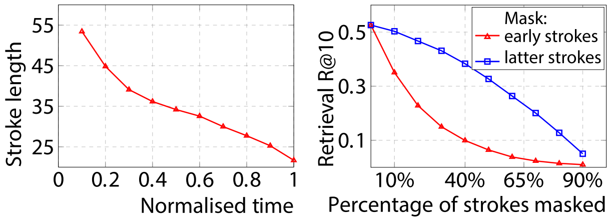
To study the domain gap between existing 'semi-synthetic' and our freehand scene sketches, we evaluate the SOTA methods FG-SBIR (Tab. 2). *Siam.-VGG16* adapts pioneering method [60] by replacing Sketch-a-Net [61] feature extractor with VGG16 [45] trained using the triplet loss [52, 57]. *HOLEF* [47] extends *Siam.-VGG16* by leveraging spatial attention to better capture fine-scale details and introducing a novel learnable distance function in the context of the triplet loss.

In addition, we explore CLIP [41], a recent SOTA method that has shown an impressive generalization ability across several photo datasets [28, 38]. *CLIP (zero-shot)* uses the pre-trained photo encoder, trained on 400 million text-photo pair. In our experiments, we use the publicly available ViT-B/32 version[4] of CLIP that use visual transformer backbone as feature extractor. Finally, *CLIP** adapts CLIP to our freehand scene sketch dataset. Since we found training CLIP to be very unstable, we only train the layer normalization [4] modules and add a fully connected layer to map sketch and photo representation to a shared 512 dimensional feature space. We train *CLIP** using triplet loss with batch size 256 and a low learning rate of 0.000001.

We train each model on the sketches from one of the three datasets: SketchyCOCO [16], SketchyScene [63] and FS-COCO (ours). Tab. 2 shows that training on 'semi-synthetic' sketch datasets like SketchyCOCO [16] and SketchyScene [63] with limited number of categories does not generalize to freehand scene sketches from our dataset with larger set of categories. This demonstrates the importance of our dataset. In the supplemental, we provide a comprehensive benchmarking by evaluating alternative sketch and image encoders.

As the image gallery when tested on our sketches is lager than for other datasets, the performance metrics on our sketches in Tab. 2 are lower even when trained on our sketches. For a more fair comparison, we create 10 additional test sets consisting of 210 sketch-image pairs (the image gallery size of the SketchyCOCO dataset) by randomly selecting them from the original set of 3000, we

[4] https://github.com/openai/CLIP



(a) Coarse-to-fine (b) Salient strokes first

Figure 2. Sketching strategies in our freehand scene sketches: Sec. 4.2. (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 at test time are masked. Top-10 accuracy calculates the percentage of test sketches for which the ground-truth image is among the first 10 ranked retrieval results.

achieve the *highest* retrieval accuracy training and testing on our data. 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.*

4.2. Strokes composition in freehand sketches

Sketch complexity Existing datasets of freehand sketches [14, 42] 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` [14] and `Sketchy` [42] datasets is 13 and 14, respectively.

4.3. What does 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 were previously observed in single object sketch datasets [14, 19, 42, 56]. We also verify the hypothesis that humans draw salient and recognizable regions early on [6, 14, 42]. We first train the classical SBIR method [60] on sketch-image pairs from our dataset: 70% of each user sketches are used for training and 30% for testing. During evaluation we follow two strategies: (i) We progressively mask a certain percentage of strokes drawn early on – indicated by the red line in Fig. 2 (b). (ii) Next, we progressively mask strokes drawn towards the end – indicated by the blue line in Fig. 2 (b). We observe that masking strokes towards the end has smaller impact on the retrieval accuracy than masking early strokes. Thus we quantitatively verify that humans draw longer (Fig. 2a) and more salient for retrieval (Fig. 2b) strokes early on.

4.3.1 Sketch captions vs. image captions

To gain insights into what information sketch captures we compare sketch captions and image captions. The vocabulary in our sketch captions overlaps with that of image captions by 81.50%. In particular, comparing sketch and image captions for each instance reveals that on average 66.5% words in sketch captions are common with image captions whereas 60.8% 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 cloud for sketch- and image captions. From Fig. 4 we observe that unlike image captions, sketch descriptions do not use "color" information. In addition, as shown in Fig. S9, we compute the percentage of nouns, verbs, and adjectives in sketch and image captions, showing 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").

4.3.2 Freehand sketches vs. image captions

We compare freehand scene sketch with textual description as queries for fine-grained image retrieval. We evaluate two baselines: (1) *CNN-RNN* the simple and classic approach that encodes text using LSTM and image using a CNN encoder (VGG-16 in our implementation) [24, 51], and (2) CLIP [41] which is one of state-of-the-art methods alongside [25] in text-based image retrieval. Both CLIP and Oscar [25] require training on *huge* datasets to be competitive. For purity of experiments we evaluate here CLIP, as its training data did not include MS-COCO dataset from which the reference images in our dataset are coming from.

CNN-RNN and *CLIP** are trained with a triplet loss. *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 [4] with batch size 256 and learning rate $1e-7$. We use the same split to train/test sets as in Tab. 2. For image-captions-based retrieval we use the same set of images and randomly select one of 5 available caption versions.

Tab. 3 shows that image captions result in better retrieval performance compared to sketch captions, containing additional information such as color. However, comparing Tab. 2 and Tab. 3 we observe that image captions are inferior to sketches for fine-grained image retrieval. While *CLIP**, using image captions as queries (Tab. 3), reaches close to R@10 accuracy of *Siam.-VGG16* (Tab. 2) using sketches as queries (Tab. 2), this is due to pre-training *CLIP** on 400 million text-photo pairs, whereas *Siam.-VGG16* was trained on a much smaller set of 7000 sketch-photo pairs.

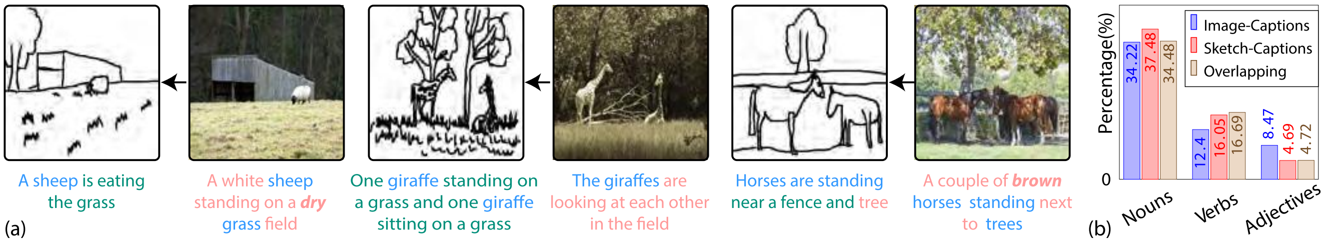
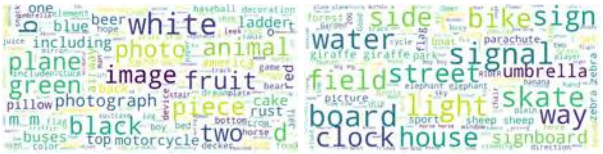


Figure 3. A qualitative and quantitative comparison of image- and sketch-captions. (a) The overlapping words are marked in Blue, the words present only in image-captions are marked in Red, while the words present only in sketch-captions are marked in Green. (b) Percentage of nouns, verbs and adjectives in image captions, sketch captions, and their overlapping words.



(a) Image Captions (b) Sketch Captions

Figure 4. The word clouds in (a) and (b) show frequently occurring words in image captions and sketch captions, respectively. The large is 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.

Table 3. Text-based image retrieval: Comparing fine-grained image retrieval when using image captions or sketch captions as input query. Image captions lead to superior result than sketch captions but inferior to SBIR, given same amount of training data.

	Retrieval accuracy			
	Image Captions		**Sketch Captions**	
Methods	R@1	R@10	R@1	R@10
CNN-RNN [46]	11.1	31.1	7.2	23.6
CLIP zero-shot [41]	21.0	50.9	11.5	35.3
CLIP*	22.1	52.3	14.8	36.6

This leads to two conclusion: (i) scene sketches can achieve higher fine-grained image retrieval accuracy than text descriptions, and (ii) scene sketches intrinsically encode fine-grained visual cues that is difficult to convey in text, hence sketch captions lead to inferior retrieval accuracy than image captions (which contains additional information such as "color").

4.3.3 Text and sketch synergy

While sketches have shown superior ability in expressing fine-grained visual cues, image captions convey additional information such as "color". We explore if a combination of both query modalities compliment each other to improve fine-grained image retrieval. Following [30], we use two simple approaches to combine sketch and text: (i) concatenate feature representation of sketch and text followed by a fully connection layer (-concat) (ii) additive approach that aggregates features from sketch and text (-add). Tab. 4 shows that combining image captions and scene sketches improves fine-grained image retrieval. This confirms that scene sketch compliments the information conveyed by text.

Table 4. Scene Sketch + Text-based fine-grained image retrieval: Combining both visual cues in scene sketch and semantic information in text leads to enhanced result across several state-of-the-arts.

Methods	R@1	R@10
CNN-RNN [46] -add	25.3	55.0
CNN-RNN [46] -concat	24.3	53.9
CLIP* -add	23.9	53.5
CLIP* -concat	23.3	52.6

4.4. Are scene sketches more informative than single-object ones?

To answer this question, we evaluate the generalization ability when trained either using object sketch or scene sketches. Training and testing *Siam.-VGG16* on object (Sketchy) and our scene (FS-COCO) sketch datasets gives 43.6 and 23.3 Top-1 retrieval accuracy (R@1), respectively. Next, we perform cross-dataset evaluation where a model trained on object sketches is evaluated on scene sketch dataset and vise-versa. Tab. 5 shows that training on object and testing on scene sketches significantly reduces R@1 from 23.3 to 4.3. However, training on scene and testing on object sketches leads to a smaller drop in R@1 from 43.6 to 29.8. This indicates that scene sketches are more informative than single-object ones for the retrieval task.

Table 5. We evaluate the generalization ability of scene sketches (ours) and object sketches [42] on the fine-grained sketch-based image retrieval task (Sec. 4.4). We show a top-1 retrieval accuracy R@1 in this table.

Trained on object sketches [42]		Trained on scene sketches	
Tested on sketches (R@1):		Tested on sketches (R@1):	
object [42]	scene (ours)	object [42]	scene (ours)
43.6	4.3	29.8	23.3

4.5. 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

Table 6. Sketch captioning (Sec. 4.5): our dataset, for the first time, 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) [36], METEOR (M) [11], ROUGE (R) [26], CIDEr (C) [50], SPICE (S) [1].

Methods	B4	M	R	C	S
Xu *et al.* [58]	13.7	17.1	44.9	69.4	14.5
AG-CVAE [53]	16.0	18.9	49.1	80.5	15.8
LNFMM [31]	16.7	21.0	52.9	90.1	16.0
LNFMM (H-Decoder)	17.3	21.1	53.2	95.3	17.2

for scene understanding. The lack of paired scene-sketch and text datasets posed the biggest bottleneck. Our dataset allows to study this problem for the first time. We evaluate several popular and SOTA methods in Tab. 6: Xu *et al.* [58] is one of the early popular work that leveraged attention mechanism with LSTM for image captioning. AG-CVAE [52] is a SOTA image captioning model that use variational auto-encoder along with an additive gaussian prior. Finally, LNFMM [31] is a recent SOTA approach using normalizing flows [13] to capture the complex joint distribution of photos and text. We show the qualitative results in Fig. 5, using the LNFMM model with the pre-training strategy we introduce in Sec. 5.

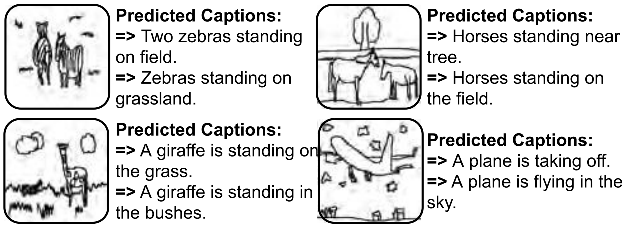
While SOTA methods like LNFMM [31] achieve a high CIDEr score of 98.4 (which goes up to 170.5 when 100 generated captions are evaluated against the ground-truth instead of 1) for image captioning on MS-COCO dataset, performance for sketch captioning (see Tab. 6) drops to 90.1. This indicates the potential for future research in developing more effective approaches for sketch captioning.

4.6. User-style adaptation

In this section, we partition a dataset differently from previous sections: we train the models discussed in Sec. 4.1 using sketches from 70 users, and test on the sketches of remaining 30 "unseen" users. Tab. 7 'Before Adapt.' column shows that the performance on sketches of "unseen" users is much worse than the one shown in Tab. 2. Hence, it is important to study methods that can provide a personalization to a new user in a few-shot scenario. Here, we adopt meta-learning [2, 15] to increase the accuracy of the fine-grained retrieval for a particular subject given just 5 subject-specific sketch examples. We repeat each experiment 5 times with 5 randomly selected sketches each time, and indicate the average performance and the standard deviation among the experiments. Tab. 7 'After Adapt.' column shows that using just 5 subject-specific sketch examples greatly improve scene-level FG-SBIR performance for *Siam.-VGG16* and *HOLEF* models. Tab. 7 shows that such large models as CLIP are less beneficial in the context of personalization.

Table 7. User-style adaptation (Sec. 4.6). We evaluate generalization of sketch-based fine-grained image retrieval models to "unseen" user styles (Before Adapt.), and the proposed personalization to a user style via meta-learning with just 5 user-scene-sketches (After Adapt.).

Methods	Before Adapt.		After Adapt.	
	R@1	R@10	R@1	R@10
Siam.-VGG16	10.6	32.5	15.5±1.4	37.6±1.9
HOLEF [47]	10.9	33.1	15.5±1.3	38.1±1.5
CLIP* [41]	4.2	22.3	4.2±0.1	22.4±0.1


Predicted Captions:
=> Two zebras standing on field.
=> Zebras standing on grassland.

Predicted Captions:
=> Horses standing near tree.
=> Horses standing on the field.

Predicted Captions:
=> A giraffe is standing on the grass.
=> A giraffe is standing in the bushes.

Predicted Captions:
=> A plane is taking off.
=> A plane is flying in the sky.

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

5. Efficient "pretext" task

Our dataset is sufficiently large (10,000 scene sketches!), especially 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 [48] might be intractable. 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 [33,37,62]. Such tasks exploit self-supervised learning, allowing to pretrain the encoder for the 'source' domain leveraging unpaired/unlabeled data. In the context of sketching, solving jigsaw puzzles [35] and converting raster to vector sketch [5] "pretext" tasks were considered. We extend the state-of-the-art sketch-vectorization [5] "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 [5] leverages a single layer Recurrent Neural Network (RNN) for sketch decoding but those designs can only reliably model up to around 200 stroke points [20], whereas our scene sketches can contain more than 3000 stroke points, making scene sketch modeling intractable. However, 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 [22]. Hence, we propose a novel 2-layered hierarchical LSTM decoder.

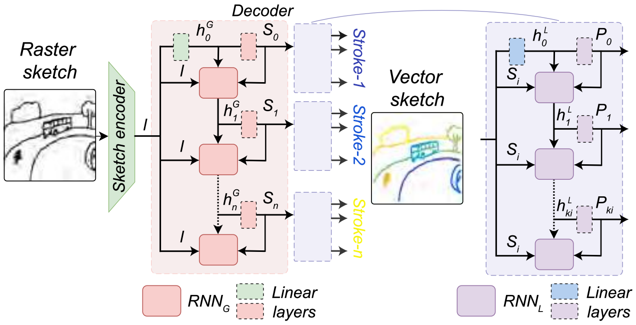
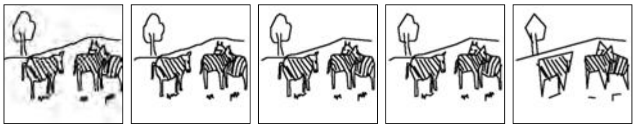


Figure 6. The proposed hierarchical decoder used for pre-training a sketch encoder.



Original Sketch 3385 points | 1136 points | 685 points | 350 points | 122 points

Figure 7. Simplifying scene sketch using RDP algorithm looses salient information. RNNs can reliably model around 200 points. The training of the single-layer RNN exploits the simplification matching the most right image.

5.1. 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, $l_{\mathrm{R}} \in \mathbb{R}^{512}$.

Naively decoding l_{R} using a single layer RNN is intractable [20]. We propose a two-level decoder consisting of two LSTMs, referred to as global and local. The global LSTM ($\mathrm{RNN}_{\mathrm{G}}$) predicts a sequence of feature vectors, each representing a stroke. The second local LSTM ($\mathrm{RNN}_{\mathrm{L}}$) predicts a sequence of points for any stroke, given its predicted feature vector.

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

Given each stroke representation S_i, the initial hidden state of local $\mathrm{RNN}_{\mathrm{L}}$ is obtained as: $h_0^{\mathrm{L}} = W_h^{\mathrm{L}} S_i + b_h^{\mathrm{L}}$. Next, h_j^{L} is updated as: $h_j^{\mathrm{L}} = \mathrm{RNN}_{\mathrm{L}}(h_{j-1}^{\mathrm{L}}; [S_i, P_{t-1}])$, where P_{t-1} is the last predicted point of the i-th stroke. A linear layer is used to predict a point: $P_t = W_y^{\mathrm{L}} h_j^{\mathrm{L}} + b_j^{\mathrm{L}}$, where 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 denotes pen's state position (q_t^1, q_t^2, q_t^3) [20].

We supervise the absolute coordinate and pen state prediction, using mean-squared error and categorical cross-entropy losses, respectively, as was proposed in [5].

Table 8. The role of pre-training with H-Decode in retrieval. The Siam.-VGG16 exploits the pre-training on ImageNet via image-classification task, while CLIP* baseline uses the model weights in ViT-B/32.

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

5.2. Evaluation & Discussion

We show the efficiency of our proposed H-Decoder in pre-training the raster sketch encoder for fine-grained image retrieval (Tab. 8) and sketch captioning (Tab. 6).

We begin the pre-training of VGG-16-based encoders of *Siam.VGG16* (Tab. 8) and *LNFMM* (Tab. 6) on the large freehand object sketch dataset QuickDraw [20], following Bhunia *et al.* [5], 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. Following the pre-training, the training of the downstream tasks starts from the weights learned with the pre-training.

Tabs. 6 and 8 show the benefit of the pre-training with the proposed decoder. In addition, we compare the performance of *Siam.VGG16*, when the pre-training is performed with the proposed H-Decoder, against a more naive approach. If we simplify scene sketches with the Ramer-Douglas Peucker (RDP) algorithm (Fig. 7), and pre-train with a single layer RNN, as proposed in [5], *Siam.VGG16* achieves R@10 of 52.1. On average, the simplified sketches contain 165 stroke points, while the original sketches contain 2437 stroke points. It performs worse than using VGG16 encoder pre-trained on ImageNet (Tab. 8). This further demonstrates the advantage of the proposed hierarchical decoder.

6. Future work and Conclusion

We introduce to the sketch community a much-needed freehand scene sketch dataset with fine-grained paired text information. With the dataset, we made the first stab towards freehand scene sketch understanding, studying tasks such as fine-grained image retrieval from scene sketches and captioning of scene sketches. We demonstrated the potential of the solutions targeting personalization to a new user style and pre-training leveraging unlabeled data. We hope that our dataset will promote research on freehand scene sketch to photo generation, better sketch captioning, and novel sketch encoding approaches that are well suited for the complexity of freehand scene sketches. We will release the dataset upon acceptance.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: semantic propositional image caption evaluation. In *ECCV*, 2016. 7

[2] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *ICLR*, 2019. 2, 7

[3] Yusuf Aytar, Lluis Castrejon, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Cross-modal scene networks. *IEEE-TPAMI*, 2018. 2

[4] Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. In *NIPS Deep Learning Symposium*, 2016. 4, 5

[5] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, and Yi-Zhe Song. Vectorization and rasterization: Self-supervised learning for sketch and handwriting. In *CVPR*, 2021. 1, 2, 7, 8

[6] Ayan Kumar Bhunia, Ayan Das, Umar Riaz Muhammad, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Pixelor: A competitive sketching ai agent. so you think you can beat me? In *SIGGRAPH Asia*, 2020. 1, 2, 5

[7] H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018. 2, 3

[8] Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. *arXiv preprint arXiv:2102.10407*, 2021. 3

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 3

[10] Ayan Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. Béziersketch: A generative model for scalable vector sketches. In *ECCV*, 2020. 1, 15

[11] Michael J. Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *WMT@ACL*, 2014. 7

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 14

[13] L Dinh, D Krueger, and Y Bengio. Nice: non-linear independent components estimation. In *ICLR, Workshop Track Proc*, 2015. 7

[14] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph.*, 2012. 1, 2, 3, 5

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2, 7

[16] Chengying Gao, Qi Liu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchycoco: Image generation from freehand scene sketches. In *CVPR*, 2020. 1, 2, 3, 4, 13, 15

[17] Songwei Ge, Vedanuj Goswami, C. Lawrence Zitnick, and Devi Parikh. Creative sketch generation. In *ICLR*, 2021. 2

[18] Yulia Gryaditskaya, Felix Hähnlein, Chenxi Liu, Alla Sheffer, and Bousseau. Lifting freehand concept sketches into 3d. In *SIGGRAPH Asia*, 2020. 1, 2

[19] Yulia Gryaditskaya, Mark Sypesteyn, Jan Willem Hoftijzer, Sylvia Pont, Frédo Durand, and Adrien Bousseau. Opensketch: a richly-annotated dataset of product design sketches. *ACM Trans. Graph.*, 2019. 1, 2, 5, 13

[20] David Ha and Douglas Eck. A neural representation of sketch drawings. In *ICLR*, 2018. 1, 2, 3, 7, 8, 15

[21] Aaron Hertzmann. Why do line drawings work? *Perception*, 2020. 1

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997. 7

[23] Josh Holinaty, Alec Jacobson, and Fanny Chevalier. Supporting reference imagery for digital drawing. In *ICCV Workshop*, 2021. 2

[24] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE-TPAMI*, 2017. 5

[25] Xiujun Li, Xi Yin, Chunyan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020. 5

[26] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004. 7

[27] Hangyu Lin, Yanwei Fu, Yu-Gang Jiang, and Xiangyang Xue. Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. In *CVPR*, 2020. 15

[28] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: common objects in context. In *ECCV*, 2014. 2, 3, 4, 11, 14

[29] Fang Liu, Changqing Zhou, Xiaoming Deng, Ran Zuo, Yu-Kun Lai, Cuixia Ma, Yong-Jin Liu, and Hongan Wang. Scenesketcher: Fine-grained image retrieval with scene sketches. In *ECCV*, 2020. 1, 3, 4, 13

[30] Kuan Liu, Yanen Li, Ning Xu, and Prem Nataranjan. Learn to combine modalities in multimodal deep learning. *arXiv preprint arXiv:1805.11730*, 2018. 6

[31] Shweta Mahajan, Iryna Gurevych, and Stefan Roth. Latent normalizing flows for many-to-many cross-domain mappings. In *ICLR*, 2020. 3, 7, 15

[32] Gioacchino Noris, Daniel Sýkora, Ariel Shamir, Stelian Coros, Brian Whited, Maryann Simmons, Alexander Hornung, Marcus Gross, and Robert Sumner. Smart scribbles for sketch segmentation. *Comp. Graph. Forum*, 31(8), 2012. 13

[33] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 7

[34] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 2

[35] Kaiyue Pang, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Solving mixed-modal jigsaw puzzle for fine-grained sketch-based image retrieval. In *CVPR*, 2020. 2, 7

[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 7

[37] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 7

[38] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 4

[39] Anran Qi, Yulia Gryaditskaya, Jifei Song, Yongxin Yang, Yonggang Qi, Timothy M. Hospedales, Tao Xiang, and Yi-Zhe Song. Toward fine-grained sketch-based 3d shape retrieval. *IEEE-TIP*, 2021. 2

[40] Yonggang Qi, Guoyao Su, Pinaki Nath Chowdhury, Mingkang Li, and Yi-Zhe Song. Sketchlattice: Latticed representation for sketch manipulation. In *ICCV*, 2021. 14, 15

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 3, 4, 5, 6, 7, 14, 15

[42] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Trans. Graph.*, 2016. 1, 2, 3, 5, 6

[43] R. G. Schneider and T. Tuytelaars. Sketch classification and classfication-driven analysis using fisher vectors. In *SIGGRAPH Asia*, 2014. 2

[44] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 2

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4, 14

[46] Jifei Song, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Fine-grained image retrieval: the text/sketch input dilemma. In *BMVC*, 2017. 1, 2, 6

[47] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *ICCV*, 2017. 4, 7, 14, 15

[48] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. *arXiv preprint arXiv:2103.01913*, 2021. 2, 7

[49] Christian Szegedy, Vincet Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 14

[50] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 7

[51] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 5

[52] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014. 4, 7

[53] Liwei Wang, Alexander G. Schwing, and Svetlana Lazebnik. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *NeurIPS*, 2017. 3, 7, 15

[54] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *ICCV*, 2021. 2

[55] Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popovic, and Niloy J. Mitra. Learning a shared shape space for multimodal garment design. In *SIGGRAPH Asia*, 2018. 1

[56] Zeyu Wang, Sherry Qiu, Nicole Feng, Holly Rushmeier, Leonard McMillan, and Julie Dorsey. Tracing versus freehand for evaluating computer-generated drawings. *ACM Trans. Graph.*, 2021. 1, 2, 5

[57] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016. 4

[58] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 3, 7, 15

[59] Chuan Yan, David Vanderhaeghe, and Yotam Gingold. A benchmark for rough sketch cleanup. *ACM Trans. Graph.*, 2020. 1

[60] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M. Hospedales, and Chen Change Loy. Sketch me that shoe. In *CVPR*, 2016. 1, 2, 4, 5, 14

[61] Qian Yu, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy Hospedales. Sketch-a-net that beats humans. In *BMVC*, 2015. 2, 4, 14

[62] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 7

[63] Changqing Zou, Qian Yu, Ruofei Du, Haoran Mo, Yi-Zhe Song, Tao Xiang, Chengying Gao, Baoquan Chen, and Hao Zhang. Sketchyscene: Rickly-annotated scene sketches. In *ECCV*, 2018. 1, 2, 3, 4, 13, 15

Supplementary Material

S1. Ethical considerations in data collection

Our dataset contains scene sketches of photos with paired textual description of the sketches. It does not include any personally identifiable information. Each sketch and caption are associated only with an ID.

Prior to agreeing to participate in the data collection, each participant was informed of the purpose of the dataset: namely that the dataset would be publicly available and released as part of a research paper with potential for commercial use. The participants were asked to accept the Contributor License Agreement that explains legal terms and conditions, and in particular it specifies that the *data collector* has the rights to distribute the data under any chosen license: The participants granted to the *data collectors* and recipients of the data distributed by the data collectors a perpetual, worldwide, non-exclusive, nocharge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sub-license, and distribute participants contributions and such derivative works. We further requested a written confirmation from annotators that they give the *data collector* permission to conduct research on the collected data and release the dataset.

Each participant who approved these terms, was assigned a random user ID. Each participant was given the option of deleting any or all their annotations/collected data at any point during the data collection process.

We also included an anonymous public discussion forum in our annotation web portal which could be used by any participant to raise concerns and collectively inform others. Annotators were also given the option of directly contacting us to raise concerns privately.

S2. Data collection: Additional detail

S2.1. Instruction for collection of Sketch Captioning

The instruction for collection of sketch captioning is similar to that of MS-COCO [28]. In particular, the subjects received the following instructions:

- Describe all the important parts of the scene.
- Do not start the sentence with “There is”.
- Do not describe unimportant details.
- Do not describe things that might have happened in the future or past.
- Do not describe what a person might say.
- Do not give proper names.
- The sentence should contain at least 5 words.

S2.2. UI of our data collection tool

Fig. S8 shows the user interface of our data collection tool. We will release the frontend and backend scripts upon acceptance. The frontend and backend scripts communicate using REST API. We use MongoDB to store the dataset in our database server. An auxiliary script was also deployed to take backups from the database server at regular time intervals to prevent data loss due to server crashes.

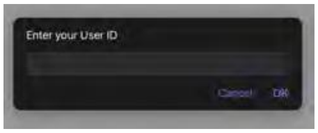
S2.3. Sample data from our dataset

Fig. S9 shows sample scene sketches from FS-COCO dataset. We will release the entire dataset upon acceptance.

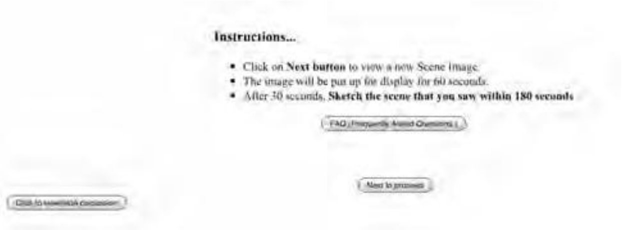
S2.4. Pilot study on optimal sketching and viewing duration

As we mention in the main document in Sections 1 and 3: “To ensure recognizable but not too detailed sketches we impose a 3-minutes sketching time constraint, where the optimal time duration was determined through a series of pilot studies. A scene reference photo is shown to a subject for 60 seconds before being asked to sketch from memory. We determined the optimal time limits through a series of pilot studies with 10 participants.” Here we provide the details of the pilot study.

We find the optimal duration for viewing a reference scene photo and drawing a scene sketch by conducting a series of pilot study on 10 individuals: (i) We started with a low duration of 30 seconds to view a reference photo and 60 seconds to draw a scene sketch. This resulted in freehand sketches that were flagged as unrecognizable by our human judge. (ii) Next, we increased the drawing time to 120 seconds while keeping the viewing time to 30 seconds. Based on interviews



(a) Login Page to Data Annotation Tool.



Instructions...

- Click on **Next button** to view a new Scene Image.
- The image will be put up for display for 60 seconds.
- After 30 seconds, **Sketch the scene that you saw within 180 seconds**

FAQ (Frequently Asked Questions)
Next to proceed
Click to view/add discussion

(b) Welcome page with instruction.

(c) Public discussion forum to raise concerns or doubts.

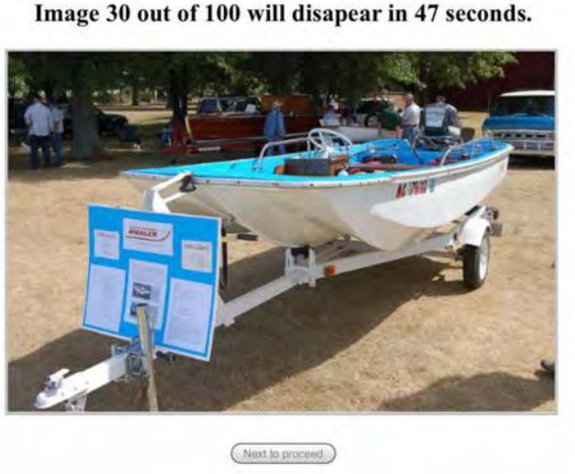
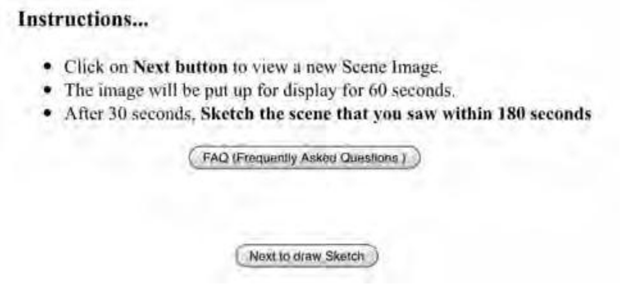


Image 30 out of 100 will disapear in 47 seconds.

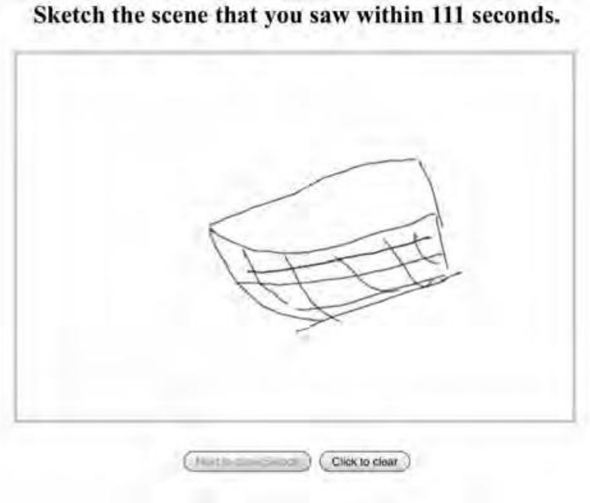
(d) View the photo for 60 seconds.



Instructions...

- Click on **Next button** to view a new Scene Image.
- The image will be put up for display for 60 seconds.
- After 30 seconds, **Sketch the scene that you saw within 180 seconds**

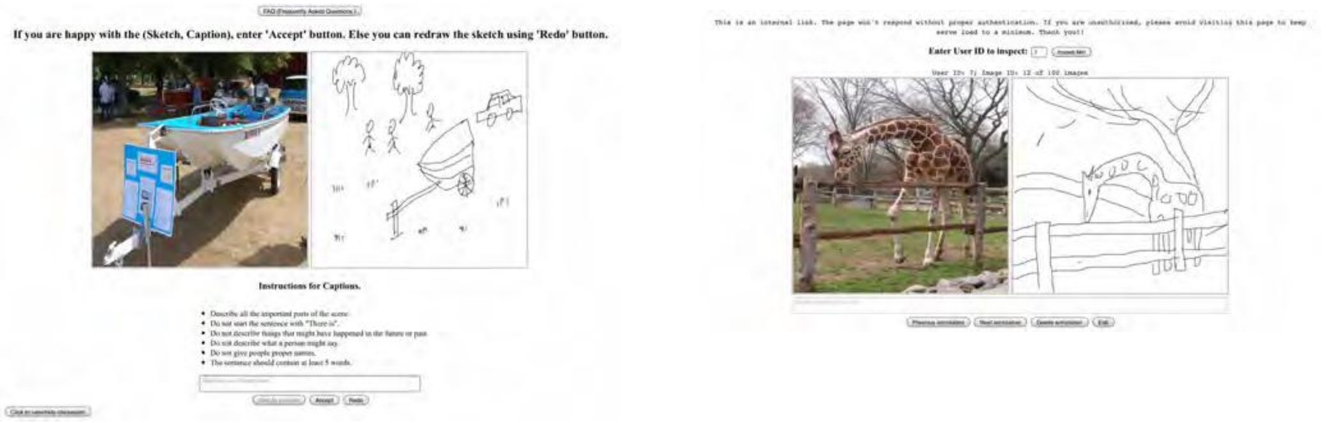
(e) Instruction after viewing photo and before starting to draw a scene sketch.



Sketch the scene that you saw within 111 seconds.

(f) Empty canvas to draw a scene sketch from memory. The sketch shown in this illustration is drawn by the authors and not by our annotators.

with our human judge and annotators we conclude that while the increase in sketching time results in barely recognizable scene sketches, annotators still missed important scene information due to the short viewing duration of 30 seconds. (iii) In



If you are happy with the (Sketch, Caption), enter 'Accept' button. Else you can redraw the sketch using 'Redo' button.

Instructions for Captions.

- Describe all the important parts of the scene.
- Do not start the sentence with "There is".
- Do not describe things that might have happened in the future or past.
- Do not describe what a person might say.
- Do not give people proper names.
- The sentence should contain at least 5 words.

This is an internal link. The page won't respond without proper authentication. If you are unauthorised, please avoid visiting this page to keep serve load to a minimum. Thank you!!

Enter User ID to inspect: 7

User ID: 7; Image ID: 12 of 100 images

(g) Verify if the drawn scene sketch is acceptable or understandable by the annotator before submitting to the server. Annotators can REDO and start from beginning of viewing the photo and drawing a scene sketch from scratch in an empty canvas.

(h) During the annotation process, a human judge evaluates if a scene sketch can be recognizable or understandable using our inspection tool. Badly drawn scene sketches will be deleted and sent back to their respective annotator for a REDO. The annotators also have the option of deleting their sketch from the portal.

Figure S8. User Interface of our data collection tool. We shall release our data collection tool upon acceptance.

the final phase of our pilot study, we increased the viewing duration to 60 seconds and sketching time to 180 seconds. This helped non-expert annotators to create scene sketches in an average of 1.7 attempts that could be understood or recognized by a human judge.

In our experiments, increasing the viewing or sketching time beyond 60 and 180 seconds resulted in overly detailed sketches. Guided by practical applications, we limit the viewing and sketching time to a duration that allows for recognizable, but not overly detailed sketches.

S2.5. Limitations and future work

Our FS-COCO includes freehand scene sketches of photos along with the textual description of the sketch. However, we did not collect stroke- or object-level annotations.

One option would have been to let sketchers to assign labels by selecting a label for each stroke while sketching. Following the arguments from the previous work on data collection [19], we refrained from this option, as that could have disturbed the natural sketching process, resulting in non-representative sketches. Indeed, we observe that objects in sketches in our dataset can share certain strokes and that participants can progress on multiple objects iteratively, not sketching one object at a time. Therefore, annotating while sketching would have turned sketching to a very tedious process and could have disturbed a natural sketching flow.

Having done a huge step towards enabling scene sketch understanding, we leave the stroke- and object-level annotations for future work. Such annotations can be done using the tools from [19] or [32].

S3. Quantitative comparison with existing datasets.

Table S9. Comparing existing Scene Sketch datasets: Unlike SketchyScene [63] and SketchyCOCO [16], our FS-COCO provides human drawn freehand scene sketches that enable analysis towards insights into how humans sketch, not possible with earlier datasets [16, 63].

Scene Sketch Dataset	Abstraction		# photos	Vectored Sketch	Paired Text	Real Human Drawn
	Object-level	Scene-level				
`SketchyScene` [63]	✓	✗	7,264	✗	✗	✗
`SketchyCOCO` [16]	✗	✓	14,081	✗	✗	✗
`SceneSketcher` [29]	✗	✓	1,225	✗	✗	✗
Our Dataset	✓	✓	10,000	✓	✓	✓

S4. Subjective quality of sketches: FS-COCO vs SketchyCOCO

Subjective quality of sketches We conduct a perceptual study to judge the subjective quality of our freehand scene sketches. In our perceptual study, each of our 5 participants were shown 100 randomly selected triplets consisting of: (i) a photo, (ii) our freehand scene sketch, (iii) and a sketch from the SketchyCOCO dataset. The participants were prompted: *"Which of the two sketches best represent the image (content relevance) and is likely to be drawn by a human (visual quality)?"*. Our freehand sketches were preferred 72.6% of the time.

S5. Additional experiments for Sec. 4.1 in the main document: Fine-grained scene sketch-based image retrieval

We provide additional experiments for Sec. 4.1 in Tab. S10. *Siam.-SN* [60] employs triplet ranking loss with Sketch-a-Net [61] as its baseline feature extractor. *HOLEF-SN* [47] extends over *Siam.-SN* employing spatial attention along with higher-order ranking loss. Our experiments suggest inferior results using Sketch-a-Net [61] backbone feature extractor. Hence, we replace the backbone feature extractor of *Siam.-SN* with VGG16 [45], we refer to this setting as *Siam.-VGG16*. Similarly, we replace Sketch-a-Net [61] backbone in *HOLEF-SN* with VGG16: *HOLEF-VGG16*. In contrast to *Siam.-VGG16* that use a common shared encoder for both sketch and photo, we use different encoders for sketches and photos in *Heter.-VGG16*. However, we note that using separate encoders leads to an inferior result. A similar drop in performance on using a heterogenous sketch/photo encoder was previously observed by Yu *et al*. [60] for object sketch datasets. Instead of using a CNN-based sketch encoder, *SketchLattice* adapts the graph-based sketch encoder proposed by Qi *et al*. [40]. We use a 32×32 evenly spaced grid or lattice for sketch representation of a rasterized scene sketch. To encode photos, we use VGG16 [45]. While such a latticed sketch representation is beneficial for sketch manipulation of object sketches, an off-the-shelf adaptation for fine-grained scene sketch-based image retrieval results in inferior to VGG16 performance. In addition, we replace our sketch encoder with a BERT-like model [12] where VGG16 is used to encode photo in *SkBert-VGG16*. Since the sketch encoding module requires vector data, we only show result on our FS-COCO. *SketchyScene* is an extension of *Siam.-SN* by replacing the backbone feature extractor from Sketch-a-Net to InceptionV3 [49]. CLIP [41] is a recent state-of-the-art method that has shown an impressive generalization ability across several photo datasets. In *CLIP (zero-shot)* we use the pre-trained photo encoder from the publicly available ViT-B/32 weights [5] as a common backbone feature extractor for scene sketch and photo. In *CLIP-variant*, we fine-tune the layer normalization layers in CLIP using our train/test split with triplet loss, batch size 256, and a very low learning rate of 0.000001.

S6. Additional discussion for Sections 4.3.1 and 4.3.2 in the main document: Fine-grained text-based image retrieval

In section 4.3.1 and 4.3.2 in the main document, our objective is to judge, given the same amount of training data, if scene sketch or image-caption, or sketch-caption is a better query modality for fine-grained image retrieval. Our FS-COCO dataset consisting of 10,000 scene sketch, photo, image-caption, and sketch-cation is a subset of the larger MS-COCO dataset. While Oscar gives a high R@1 score of 57.5 for text based image retrieval, it was trained on the entire training set of MS-COCO [28]. This results in an unfair comparison. Hence for a fair evaluation, we use CLIP [41] which in spite of training of a much larger dataset of 400 million text-image pairs, did not include MS-COCO.

S6.1. Additional experiments for Sec. 4.5 in the main document: Sketch Captioning

Tab. S11 includes additional experiments for Sec. 4.5 for sketch captioning using existing state-of-the-art methods.

S7. H-Decoder: Additional experiments and discussions

S7.1. H-Decoder implementation details

We use the data format that represents a sketch as a set of pen stroke actions. A sketch is a list of points, and each point is a 5 dimensional vector: $(x, y, q1, q2, q3)$. The first two logits (x, y) represent the absolute coordinate in the x and y directions of the pen. The later three $(q1, q2, q3)$ represent a binary one-hot vector of 3 possible states: (i) *pen down state:* The first pen state $q1$ denotes that the pen is touching the paper. This indicates that a line will be drawn connecting the next point with the current point. (ii) *pen up state:* The second pen state $q2$ indicates the pen will be lifted from the paper after the current point

[5] https://github.com/openai/CLIP

Table S10. Fine-grained freehand-scene-sketch-based image retrieval: Additional experiments for Sections 4.3.1 and 4.3.2 in the main document.

Methods	Trained On																	
	SketchyScene (S-Scene) [63]						SketchyCOCO (S-COCO) [16]						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.-SN	2.7	17.3	<0.1	1.1	0.1	3.2	<0.1	<0.1	6.2	32.9	<0.1	<0.1	1.2	9.1	<0.1	3.9	4.7	21.0
Siam.-VGG16	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
Heter.-VGG16	15.9	38.4	0.2	3.7	0.8	5.8	0.1	1.6	34.9	76.1	<0.1	0.3	4.2	20.1	1.9	10.7	19.2	47.6
HOLEF-SN [47]	2.9	17.7	<0.1	1.3	0.2	3.2	<0.1	<0.1	6.2	40.7	<0.1	<0.1	1.2	9.3	<0.1	4.1	4.9	21.7
HOLEF-VGG16 [47]	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
SketchLattice [40]	15.9	37.2	0.1	3.3	0.8	5.6	0.1	1.5	33.7	74.3	<0.1	0.3	3.7	19.4	0.7	9.5	18.9	46.5
Lin *et al*. [27] (SkBert-VGG16)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	11.3	37.2
SketchyScene [63]	20.6	41.7	0.9	3.9	1.8	6.1	0.2	1.7	36.5	78.6	<0.1	0.4	5.1	24.1	2.4	11.5	23.0	52.3
CLIP (zero-shot) [41]	1.26	9.70	–	–	–	–	–		1.85	9.41	–	–	–	–	–	–	1.17	6.07
CLIP-variant	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

Table S11. Sketch Captioning: Our novel dataset, for the first time, enables captioning of scene sketches. We provide the results of some popular captioning methods originally developed for photos. Empirical results suggests there is significant gap in performance in comparison to image captioning literature. We hope our dataset and quantitative results will inspire future methods to caption scene sketches.

Methods	Belu-1	Belu-2	Belu-3	Belu-4	Meteor	Rouge	CIDEr	Spice
Xu *et al*. [58]	46.2	29.1	17.8	13.7	17.1	44.9	69.4	14.5
GMM-CVAE [53]	49.6	33.9	18.2	15.5	18.3	48.7	77.6	15.5
AG-CVAE [53]	50.9	34.1	19.2	16.0	18.9	49.1	80.5	15.8
LNFMM [31]	52.2	35.7	20.0	16.7	21.0	52.9	90.1	16.0
LNFMM (H-Decoder)	**54.7**	**37.3**	**22.5**	**17.3**	**21.1**	**53.2**	**95.3**	**17.2**

to mark the end of a stroke. (iii) *pen end state:* The final pen state $q3$ represent that the drawing of scene sketch has ended, and subsequent points will not be rendered.

Our hierarchical decoder consists of two LSTMs: (i) The global LSTM (RNN_{G}) that predicts a sequence of feature vectors, each representing a stroke. (ii) A second local LSTM (RNN_{L}) predicting a sequence of points for any stroke, given its predicted feature vector. The stroke points P_t are predicted across i^{th} and j^{th} steps in RNN_{G} and RNN_{L} respectively. In more details, let's assume the local RNN_{L} predicts P_t with pen up state $(0, 1, 0)$ at the j^{th} unroll step, given input stroke feature S_i. It will then trigger a single step unroll of the global RNN_{G} to predict the next stroke representation S_{i+1}. This will re-initialise RNN_{L} to predict stroke points starting with P_{t+1} for S_{i+1} where P_t is the last predicted point. The unrolling of both RNN_{L} and RNN_{G} comes to a halt upon predicting P_t with pen end state $(0, 0, 1)$. We define P_0 as $(0, 0, 1, 0, 0)$.

S7.2. Learning to synthesize human-like sketches

A byproduct of our hierarchical sketch decoder is a naive photo to vector sketch synthesis pipeline. Fig. S9 shows preliminary samples of scene sketches synthesized using our proposed sketch decoder. To improve these results, future work can exploit VAE-based solutions, sequentially generating sketches [20], or paramaterized strokes representation [10] to tackle the challenges posed by scene sketches.

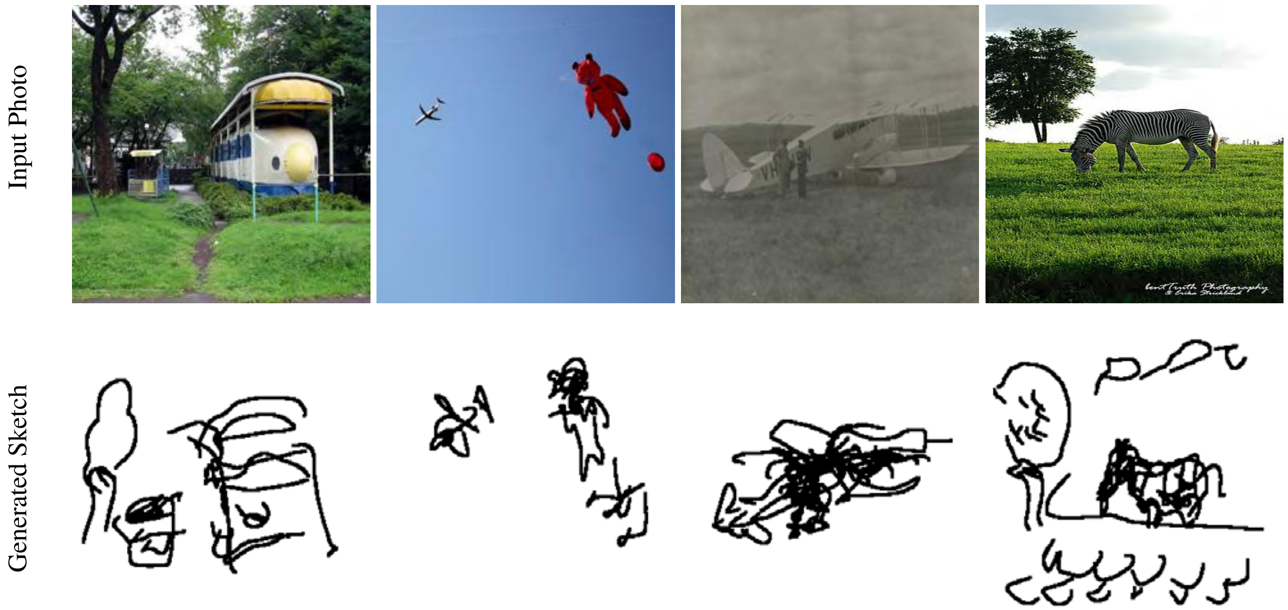
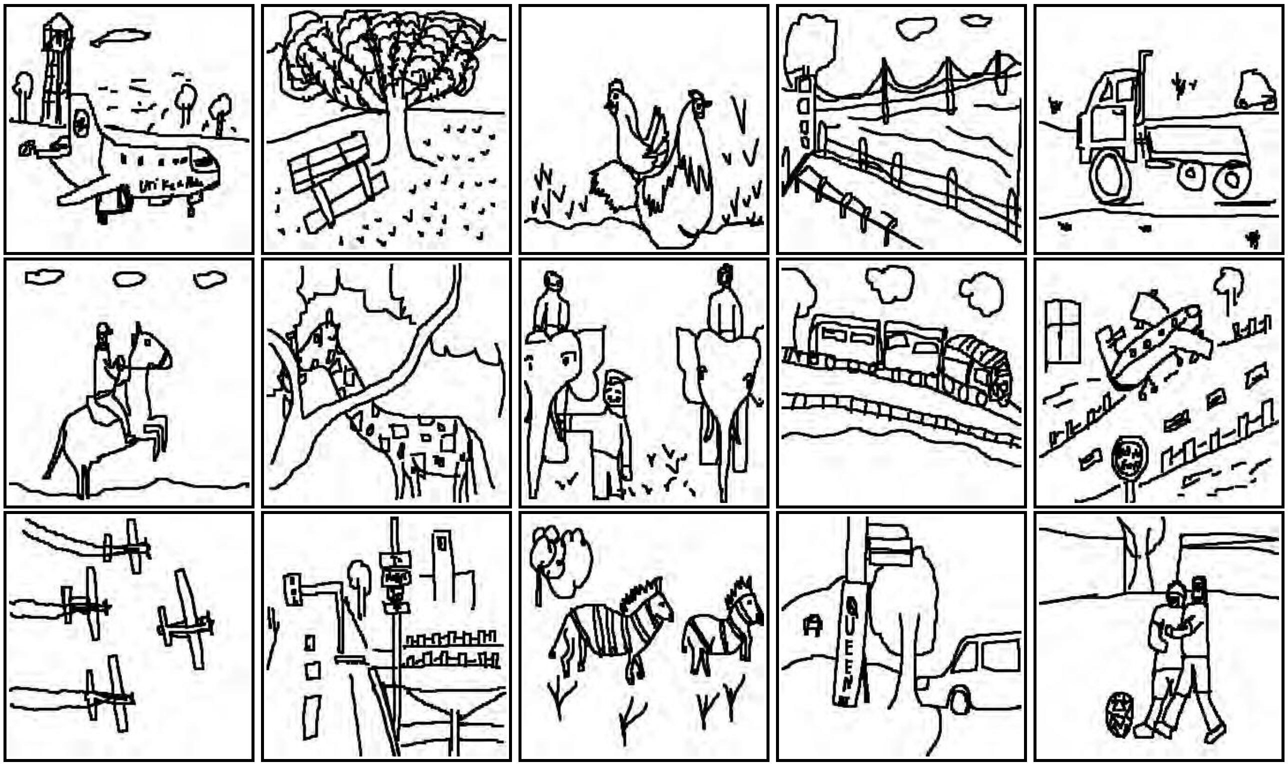


Figure S9. Photo to vectored sketch synthesis: Our novel dataset allows interesting downstream applications such photo to vectored scene sketch synthesis as a byproduct of our hierarchical decoder used during pre-training. For brevity, we only show qualitative results using VGG-16 encoder followed by the hierarchical decoder.



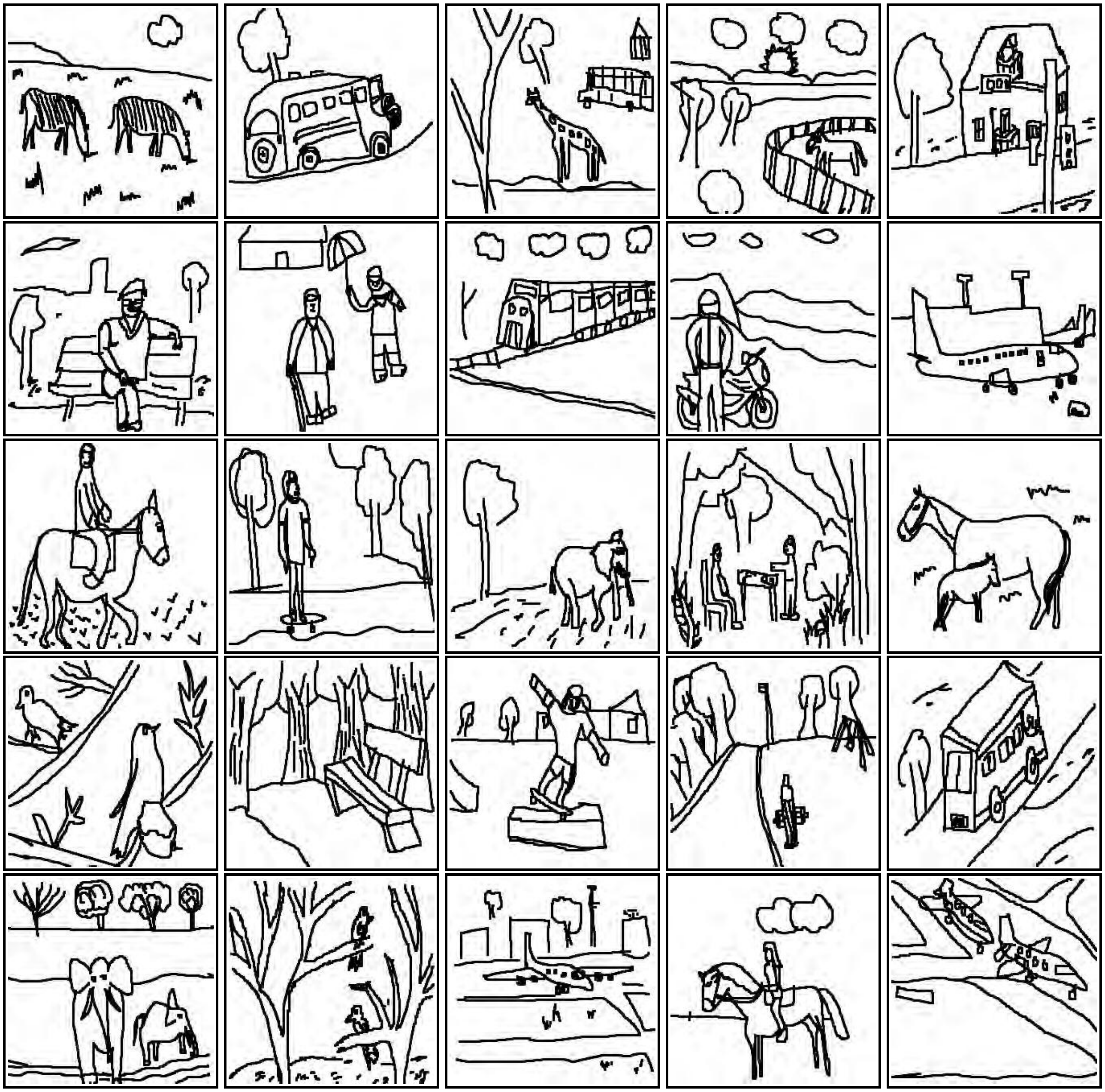


Figure S9. Sample sketches from our FS-COCO dataset, the first dataset of 10,000 unique freehand scene sketches, drawn by 100 non-expert participants. We envision this dataset to permit a multitude of novel tasks and to contribute to the fundamental understanding of visual abstraction and expressivity in scene sketching. With our work, we make the first stab in this direction by (i) studying the role of freehand sketches in fine-grained image retrieval and (ii) sketch understanding on the example of sketch-captioning.