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

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

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

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 A detailed description of FSCOCO and comparison with existing SketchyCOCO [5] and SketchyScene [22]

In Sec. 4.1 in the main document, we compare with existing datasets SketchyCOCO [5] and SketchyScene [22]. Here, we provide the detailed statistics on cat-

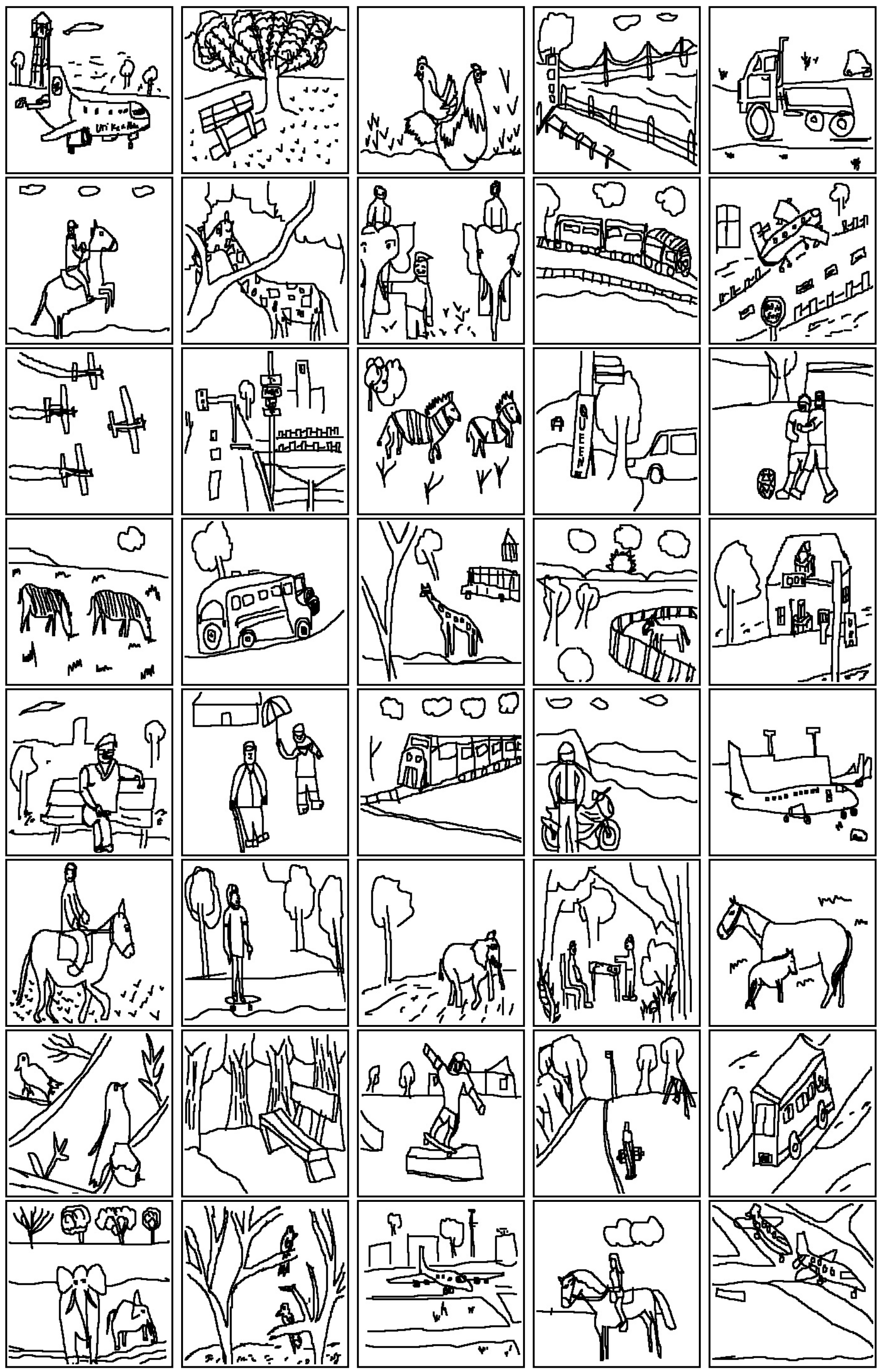


Fig. S1: Sample sketches from our FS-COCO dataset.

egories in SketchyCOCO [5] and SketchyScene [22] and our dataset in Tab. S1, Tab. S2 and Tab. S3, respectively.

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 [6], 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. 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 [6] or [11]. For our dataset, we compute two estimates of category distribution: (1) based on semantic segmentation labels of images FS-COCO (e_l), and (2) based on the occurrence of a word in a sketch caption FS-COCO (e_c). The detailed statistics is provided in Tab. S3.

Table S1: We present a detailed list of categories in SketchyCOCO (SketchyCOCO-All) [5] along with the number of sketches that contain each category (# sketches), and the percentage of sketches that include a particular category (# percentage). SketchyCOCO-FG denotes a subset of SketchyCOCO-All that is used for fine-grained scene-level sketch-based image retrieval.

SketchyCOCO-FG			SketchyCOCO-All		
Category	# sketches	# percentage	Category	# sketches	# percentage
clouds	824	67.27	clouds	9761	69.32
tree	784	64.00	tree	9051	64.28
grass	752	61.39	grass	8857	62.90
airplane	80	6.53	airplane	944	6.70
giraffe	60	4.90	giraffe	925	6.57
horse	53	4.33	zebra	595	4.23
zebra	48	3.92	horse	519	3.69
cow	43	3.51	cow	450	3.20
dog	43	3.51	dog	367	2.61
elephant	25	2.04	elephant	351	2.49
car	23	1.88	sheep	339	2.41
sheep	22	1.80	car	255	1.81
motorcycle	14	1.14	motorcycle	139	0.99
traffic light	10	0.82	fire hydrant	112	0.80
fire hydrant	9	0.73	traffic light	96	0.68
cat	5	0.41	bicycle	57	0.40
bicycle	5	0.41	cat	33	0.23

Table S2: A detailed list of categories is presented for SketchyScene (SketchyScene-All) [22] along with the number of sketches that contain each category (# sketches), and the percentage of sketches that include a particular category (# percentage). SketchyScene-FG denotes a subset of SketchyScene-All that is used for fine-grained scene-level sketch-based image retrieval.

SketchyScene-FG			SketchyScene-All		
Category	# sketches	# percentage	Category	# sketches	# percentage
tree	2154	79.07	tree	5723	40.64
grass	2084	76.51	grass	5412	38.43
cloud	1880	69.02	cloud	5170	36.72
road	1168	42.88	road	3067	21.78
sun	1020	37.44	sun	2917	20.72
house	936	34.36	house	2841	20.18
mountain	889	32.64	people	2417	17.16
people	802	29.44	mountain	2357	16.74
flower	786	28.85	flower	2077	14.75
fence	738	27.09	fence	1857	13.19
dog	507	18.61	dog	1485	10.55
bird	463	17.00	bird	1206	8.56
car	422	15.49	car	1084	7.70
bench	334	12.26	bench	971	6.90
cow	308	11.31	cow	781	5.55
sheep	307	11.27	sheep	763	5.42
rabbit	265	9.73	cat	726	5.16
cat	259	9.51	chicken	665	4.72
bus	259	9.51	rabbit	648	4.60
chicken	249	9.14	bus	636	4.52
butterfly	224	8.22	butterfly	603	4.28
duck	212	7.78	street	567	4.03
street	194	7.12	duck	507	3.60
picnic	142	5.21	picnic	437	3.10
basket	125	4.59	basket	384	2.73
apple	107	3.93	pig	333	2.36
bee	105	3.85	apple	330	2.34
pig	103	3.78	truck	293	2.08
truck	89	3.27	bee	243	1.73
horse	73	2.68	horse	235	1.67
moon	57	2.09	grape	214	1.52
grape	54	1.98	table	197	1.40
table	54	1.98	moon	193	1.37
banana	50	1.84	banana	162	1.15
bicycle	48	1.76	bicycle	155	1.10
bucket	45	1.65	chair	138	0.98

Continued on next page

Table S2 – continued from previous page

Category	# sketches	# percentage	Category	# sketches	# percentage
cup	37	1.36	bucket	125	0.89
chair	37	1.36	star	114	0.81
airplane	34	1.25	airplane	110	0.78
bottle	32	1.17	cup	109	0.77
star	28	1.03	bottle	106	0.75
balloon	27	0.99	balloon	90	0.64
dinnerware	23	0.84	umbrella	59	0.42
umbrella	20	0.73	dinnerware	51	0.36
sofa	3	0.11	sofa	31	0.22

Table S3: We list all categories present in FSCOCO. For our dataset, we compute two estimates of category distribution: (1) based on semantic segmentation labels of images (e_l), and (2) based on the occurrence of a word in a sketch caption (e_c). We present the number of sketches (# sketches) and percentage of sketches (# percentage) containing each category.

FS-COCO (e_c)			FS-COCO (e_l)		
Category	# sketches	# percentage	Category	# sketches	# percentage
grass	866	8.66	tree	6789	67.89
road	643	6.43	grass	6486	64.86
tree	638	6.38	sky-other	5530	55.3
giraffe	637	6.37	person	3813	38.13
kite	543	5.43	building-other	2235	22.35
zebra	422	4.22	clouds	2161	21.61
horse	407	4.07	bush	1616	16.16
clock	394	3.94	metal	1404	14.04
dog	338	3.38	road	1382	13.82
cow	308	3.08	pavement	1269	12.69
sheep	305	3.05	dirt	1235	12.35
train	305	3.05	fence	1206	12.06
person	292	2.92	car	1162	11.62
bird	267	2.67	airplane	1065	10.65
elephant	232	2.32	clothes	1001	10.01
bench	206	2.06	house	935	9.35
frisbee	200	2	plant-other	916	9.16
airplane	162	1.62	frisbee	777	7.77
light	156	1.56	giraffe	770	7.7
house	156	1.56	kite	743	7.43
car	146	1.46	bird	617	6.17
bear	129	1.29	mountain	617	6.17
mountain	114	1.14	truck	608	6.08

Continued on next page

Table S3 – continued from previous page

Category	# sketches	# percentage	Category	# sketches	# percentage
bus	103	10.3	cow	577	5.77
skateboard	90	0.9	zebra	562	5.62
river	88	0.88	bench	544	5.44
umbrella	88	0.88	wall-concrete	529	5.29
branch	87	0.87	horse	528	5.28
fence	84	0.84	sheep	521	5.21
truck	76	0.76	clock	517	5.17
hill	71	0.71	traffic light	496	4.96
bridge	63	0.63	roof	485	4.85
boat	60	0.60	ground-other	484	4.84
wood	38	0.38	wood	452	4.52
bush	30	0.3	dog	438	4.38
rock	28	0.28	hill	434	4.34
fruit	26	0.26	branch	418	4.18
cat	25	0.25	rock	367	3.67
chair	22	0.22	stop sign	356	3.56
bicycle	22	0.22	river	333	3.33
table	20	0.2	train	333	3.33
flower	19	0.19	light	308	3.08
snow	16	0.16	gravel	301	3.01
banana	16	0.16	skateboard	294	2.94
mirror	13	0.13	backpack	293	2.93
apple	13	0.13	elephant	279	2.79
window	11	0.11	water-other	266	2.66
plate	11	0.11	textile-other	259	2.59
motorcycle	10	0.1	leaves	251	2.51
tent	10	0.1	railroad	250	2.5
stone	9	0.09	structural-other	242	2.42
sea	9	0.09	window-other	238	2.38
shoe	8	0.08	handbag	238	2.38
platform	8	0.08	stone	236	2.36
vase	7	0.07	sports ball	229	2.29
orange	7	0.07	plastic	221	2.21
leaves	5	0.05	bus	212	2.12
hat	4	0.04	wall-other	212	2.12
mat	4	0.04	umbrella	196	1.96
banner	4	0.04	wall-brick	178	1.78
metal	4	0.04	flower	178	1.78
donout	4	0.04	cage	173	1.73
railing	4	0.04	straw	172	1.72
net	3	0.03	banner	162	1.62
roof	3	0.03	bicycle	162	1.62

Continued on next page

Table S3 – continued from previous page

Category	# sketches	# percentage	Category	# sketches	# percentage
surfboard	3	0.03	motorcycle	160	1.6
bowl	3	0.03	fire hydrant	158	1.58
carrot	3	0.03	chair	155	1.55
tie	3	0.03	fog	153	1.53
bottle	3	0.03	tent	149	1.49
laptop	3	0.03	bridge	146	1.46
snowboard	3	0.03	boat	143	1.43
sand	3	0.03	bear	141	1.41
book	3	0.03	baseball bat	135	1.35
suitcase	3	0.03	wall-stone	126	1.26
cloth	3	0.03	stairs	118	1.18
cage	2	0.02	railing	115	1.15
paper	2	0.02	baseball glove	108	1.08
cup	2	0.02	wall-wood	86	0.86
pavement	2	0.02	playingfield	83	0.83
pizza	2	0.02	mud	81	0.81
door	2	0.02	furniture-other	80	0.8
bed	2	0.02	door-stuff	78	0.78
cake	2	0.02	solid-other	71	0.71
mud	2	0.02	bottle	70	0.7
toilet	1	0.01	platform	69	0.69
clothes	1	0.01	floor-other	68	0.68
toothbrush	1	0.01	ceiling-other	59	0.59
blender	1	0.01	cloth	59	0.59
railroad	1	0.01	tennis racket	56	0.56
scissors	1	0.01	potted plant	56	0.56
skyscraper	1	0.01	dining table	54	0.54
			table	47	0.47
			cell phone	46	0.46
			tie	45	0.45
			net	45	0.45
			apple	45	0.45
			snowboard	42	0.42
			suitcase	41	0.41
			wall-panel	41	0.41
			teddy bear	40	0.4
			floor-stone	40	0.4
			paper	39	0.39
			cat	37	0.37
			surfboard	35	0.35
			moss	26	0.26
			cup	25	0.25

Continued on next page

Table S3 – continued from previous page

Category	# sketches	# percentage	Category	# sketches	# percentage
			skis	25	0.25
			bowl	22	0.22
			banana	22	0.22
			vase	21	0.21
			fruit	20	0.2
			orange	19	0.19
			floor-wood	17	0.17
			mirror-stuff	16	0.16
			book	15	0.15
			parking meter	14	0.14
			blanket	12	0.12
			carboard	11	0.11
			laptop	11	0.11
			floor-tile	10	0.1
			food-other	9	0.09
			towel	9	0.09
			hot dog	8	0.08
			sandwich	7	0.07
			window-blind	6	0.06
			carrot	6	0.06
			waterdrops	6	0.06
			cake	6	0.06
			ceiling-tile	4	0.04
			toilet	4	0.04
			wall-tile	4	0.04
			fork	4	0.04
			toothbrush	4	0.04
			rug	3	0.03
			oven	3	0.03
			knife	3	0.03
			vegetable	3	0.03
			pizza	3	0.03
			remote	3	0.03
			couch	2	0.02
			donout	2	0.02
			spoon	2	0.02
			wine glass	2	0.02
			scissors	2	0.02
			mat	1	0.01
			counter	1	0.01
			hair dryer	1	0.01
			napkin	1	0.01

Continued on next page

Table S3 – continued from previous page

Category	# sketches	# percentage	Category	# sketches	# percentage
			keyboard	1	0.01

S2.1 Indoor categories in FSCOCO

List of Indoor categories for FSCOCO (l): toothbrush, banner, orange, donut, pizza, metal, table, book, apple, laptop, cup, fruit, chair, mat, plate, bowl, window, door, carrot, clothes, blender, banana, light, mirror, cloth, scissors, toilet, bed, cake, paper, clock, vase, bottle
List of Indoor categories for FSCOCO (u): toothbrush, fork, banner, keyboard, donut, orange, knife, pizza, hot dog, metal, window-blind, table, dining table, book, apple, couch, napkin, wall-stone, laptop, floor-tile, floor-wood, rug, cup, fruit, sandwich, chair, potted plant, floor-stone, towel, blanket, ceiling-tile, mat, mirror-stuff, stairs, cell phone, bottle, counter, bowl, wall-other, door-stuff, ceiling-other, spoon, carrot, clothes, floor-other, banana, wall-brick, wall-panel, furniture-other, light, wall-concrete, window-other, cloth, scissors, hair drier, toilet, remote, textile-other, plastic, teddy bear, wine glass, paper, cardboard, cake, wall-wood, wall-tile, clock, vase, vegetable, oven, food-other

S2.2 Outdoor categories in FSCOCO

List of Outdoor categories for FSCOCO (l): person, house, kite, branch, fence, mud, leaves, mountain, bush, cat, hill, skyscraper, river, umbrella, railing, boat, bridge, horse, sea, pavement, surfboard, airplane, bear, skateboard, frisbee, bird, stone, tie, train, suitcase, flower, tent, snowboard, railroad, rock, grass, motorcycle, dog, net, cow, platform, sheep, giraffe, road, sand, roof, wood, hat, truck, snow, car, shoe, bicycle, bus, tree, bench, elephant, cage, zebra.
List of Outdoor categories for FSCOCO (u): person, house, kite, branch, water-other, fence, mud, leaves, mountain, bush, structural-other, cat, hill, moss, fire hydrant, stop sign, dirt, straw, ground-other, river, skis, umbrella, baseball glove, railing, boat, bridge, horse, pavement, surfboard, airplane, bear, traffic light, waterdrops, building-other, bird, stone, tennis racket, train, tie, suitcase, tent, fog, railroad, flower, handbag, plant-other, snowboard, rock, grass, motorcycle, frisbee, dog, net, cow, platform, sports ball, sheep, giraffe, baseball bat, road, clouds, roof, wood, truck, car, skateboard, sky-other, playingfield, backpack, bicycle, bus, tree, gravel, bench, elephant, cage, parking meter, solid-other, zebra.

S2.3 Categories common between FSCOCO and SketchyCOCO [5]

List of categories common between FSCOCO (l) and SketchyCOCO: car, grass, motorcycle, dog, horse, cow, giraffe, cat, bicycle, airplane, tree, sheep, elephant, zebra.
List of categories common between FSCOCO (u) and SketchyCOCO: car, grass, motorcycle, dog, horse, cow, cat, bicycle, fire hydrant, airplane, tree, traffic light, sheep, elephant, giraffe, clouds, zebra.

S2.4 Categories common between FSCOCO and SketchyScene [22]

List of categories common between FSCOCO (l) and SketchyScene: house, fence, table, mountain, cat, apple, umbrella, horse, cup, chair, airplane, bird, flower, grass, dog, cow, banana, sheep, road, truck, car, bus, bicycle, tree, bench, bottle.
List of categories common between FSCOCO (u) and SketchyScene: house, fence, table, mountain, cat, apple, umbrella, horse, cup, chair, airplane, bird, flower, grass, dog, cow, banana, sheep, road, truck, car, bus, bicycle, tree, bench, bottle.

S3 Data collection: Additional detail

S3.1 Instructions for sketch captioning

The instructions for sketch captioning are similar to that of MS-COCO [9]. Namely, 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.

S3.2 UI of our data collection tool

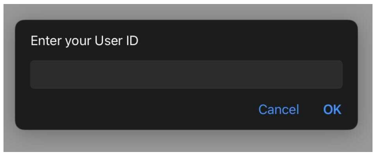
Figs. S2 to S4 shows the user interface of our data collection tool. We release the frontend and backend scripts at `https://github.com/pinakinathc/SketchX-SST`. The frontend and backend scripts communicate using REST API.

S3.3 Sample data from our dataset

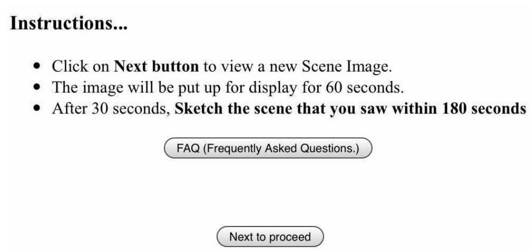
Fig. 3 shows sample scene sketches from FS-COCO. We released the dataset under CC BY-NC 4.0 license at `https://github.com/pinakinathc/fscoco`.

S3.4 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.

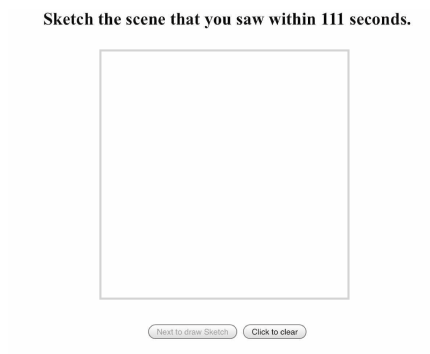


(a) Login page to a data annotation tool.



(b) Welcome page with instruction.

(c) View the photo for 60 seconds.



(d) Sketching area.

Fig. S2: User sketching interface of our data collection tool. We will release our data collection tool upon acceptance.

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 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 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 applica-

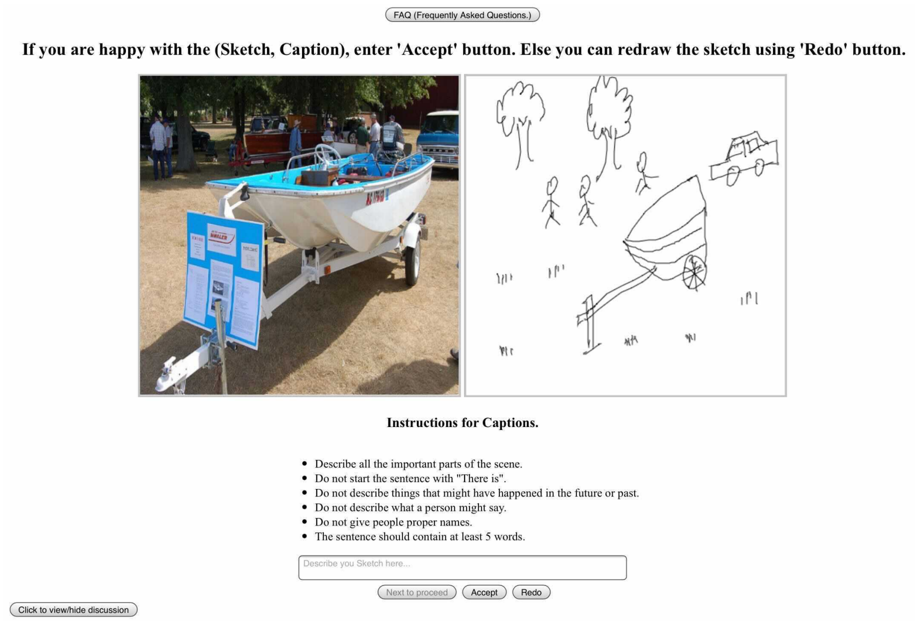


Fig. S3: Review by an annotator before submitting a sketch and a caption. If annotators are not satisfied with the sketch, they can redo the sketch by first observing the photo and then drawing the scene sketch from scratch on a blank canvas.

tions, we limit the viewing and sketching time to a duration that allows for recognizable, but not overly detailed sketches.

S4 Additional experiments for Sec. 5.1 in the main document: Fine-grained scene sketch-based image retrieval

We provide additional experiments for Sec. 5.1 in Tab. S5. *Siam.-SN* [20] employs triplet ranking loss with Sketch-a-Net [21] as its baseline feature extractor. *HOLEF-SN* [16] extends over *Siam.-SN* employing spatial attention along with higher-order ranking loss. Our experiments suggest inferior results using Sketch-a-Net [21] backbone feature extractor. Hence, we replace the backbone feature extractor of *Siam.-SN* with VGG16 [15], we refer to this setting as *Siam.-VGG16*. Similarly, we replace Sketch-a-Net [21] 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.* [20] for object sketch

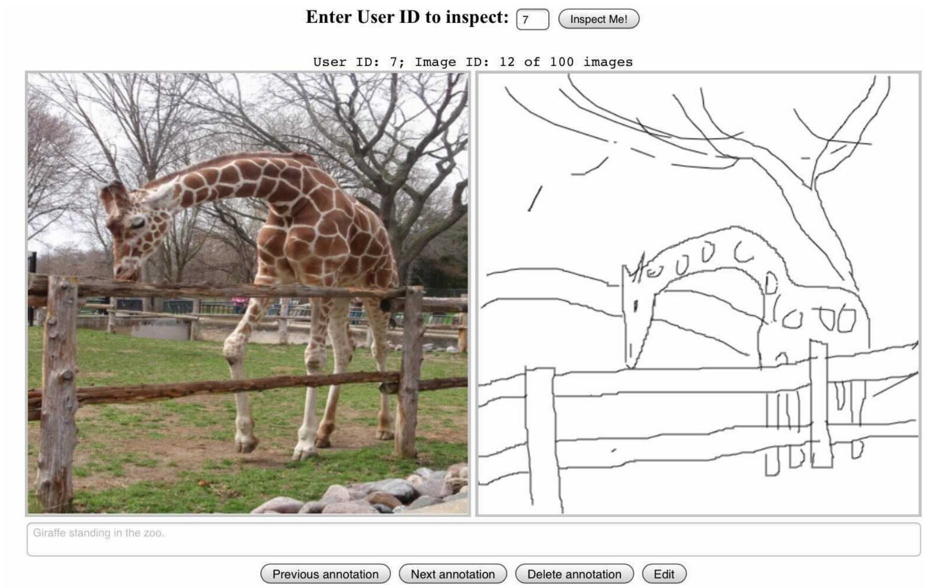


Fig. S4: One dedicated human judge evaluates if a scene sketch is recognizable or understandable. Poorly drawn scene sketches are removed and sent back to the appropriate annotator for rework.

datasets. Instead of using a CNN-based sketch encoder, *SketchLattice* adapts the graph-based sketch encoder proposed by Qi *et al*. [12]. We use a 32×32 evenly spaced grid or lattice for sketch representation of a rasterized scene sketch. To encode photos, we use VGG16 [15]. 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 [3] 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 [17]. CLIP [13] 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 [1] 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.

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

S4.1 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. S4 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 S4: We evaluate the generalization ability of scene sketches (ours) and object sketches [14] on the fine-grained sketch-based image retrieval task (Sec. S4.1). We show a top-1 retrieval accuracy R@1 in this table.

Trained on object sketches [14]		Trained on scene sketches	
Tested on sketches (R@1):		Tested on sketches (R@1):	
object [14]	scene (ours)	object [14]	scene (ours)
43.6	4.3	29.8	23.3

S4.2 Additional discussion on the need for computing two estimates of the category distribution in FSCOCO.

As mentioned in Sec. 4.1 of the main document, to compute the statistics on the categories present in FSCOCO, we use two estimates: (1) e_l, based on the semantic segmentation labels in images and (2) e_c, based on the occurrence of a word in a sketch caption. The reason for using two estimates is elaborated in Fig. S5 where counting occurrence of categories in FS-COCO based on the occurrence of a word in a sketch-caption (FS-COCO (e_c)) would lead to a lower estimate. This is because participants in FS-COCO no not exhaustively describe in sketch-caption all the objects present in sketches. Simultaneously, counting occurrence of categories in FS-COCO based on the semantic segmentation labels in images (FS-COCO (e_l)) would lead to a higher estimate since not all regions in a photo are drawn by a participant.

S5 Additional discussion for Sec. 5.2 in the main document: Fine-grained text-based image retrieval

In Sec. 5.2 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

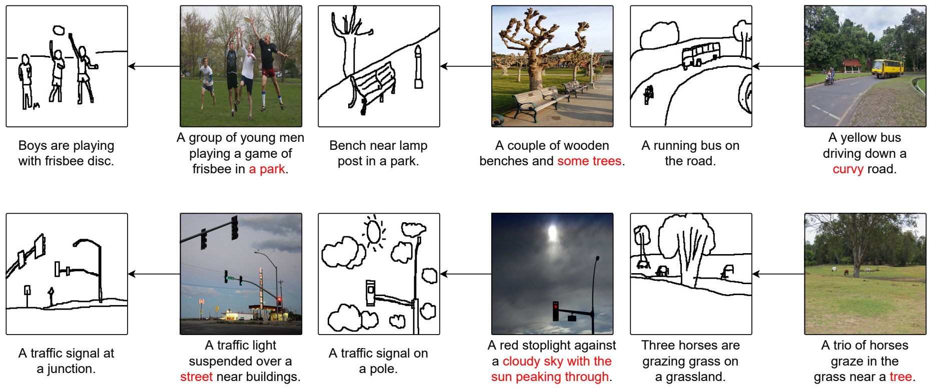


Fig. S5: The Participants in FS-COCO do not exhaustively describe in sketch-captions all the objects present in sketches. The categories that are drawn in sketch but not described in sketch-captions are marked in red.

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 [9]. This results in an unfair comparison. Hence for a fair evaluation, we use CLIP [13] which in spite of training on a much larger dataset of 400 million text-image pairs, did not include MS-COCO.

Table S5: Fine-grained freehand-scene-sketch-based image retrieval: Additional experiments for Sec. 5.2 in the main document.

Methods	Trained On																	
	SketchyScene (S-Scene) [22]						SketchyCOCO (S-COCO) [5]						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 [16]	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 [16]	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 [12]	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.* [8] (SkBert-VGG16)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	11.3	37.2
SketchyScene [22]	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) [13]	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

S5.1 Additional experiments for Sec. 5.3 in the main document: Sketch Captioning

Tab. S6 includes additional experiments for Sec. 5.3 for sketch captioning using existing state-of-the-art methods.

Table S6: 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.* [19]	46.2	29.1	17.8	13.7	17.1	44.9	69.4	14.5
GMM-CVAE [18]	49.6	33.9	18.2	15.5	18.3	48.7	77.6	15.5
AG-CVAE [18]	50.9	34.1	19.2	16.0	18.9	49.1	80.5	15.8
LNFMM [10]	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**

S6 User-style adaptation

In this section, we split the dataset differently than in the main paper: we train the models discussed in Sec. 5.1 using sketches from 70 users, and test on the sketches of remaining 30 "unseen" users. Tab. S7 'Before Adapt.' column shows that the performance on sketches of "unseen" users is worse than the one shown in Tab. 3. Hence, it is important to explore techniques that can provide personalization to a new user in a few-shot scenario. Here, we use meta-learning [4,1] 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. S7 '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. S7 shows that such large models as CLIP are less beneficial in the context of personalization.

Table S7: User-style adaptation (Sec. S6). 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 [16]	10.9	33.1	15.5±1.3	38.1±1.5
CLIP* [13]	4.2	22.3	4.2±0.1	22.4±0.1

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 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. S6 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 [7], or paramaterized strokes representation [2] to tackle the challenges posed by scene sketches.

References

1. Antoniou, A., Edwards, H., Storkey, A.: How to train your maml. In: ICLR (2019) 16
2. Das, A., Yang, Y., Hospedales, T., Xiang, T., Song, Y.Z.: Béziersketch: A generative model for scalable vector sketches. In: ECCV (2020) 17
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019) 13
4. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017) 16
5. Gao, C., Liu, Q., Wang, L., Liu, J., Zou, C.: Sketchycoco: Image generation from freehand scene sketches. In: CVPR (2020) 1, 3, 9, 15
6. 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) 3
7. Ha, D., Eck, D.: A neural representation of sketch drawings. In: ICLR (2018) 17

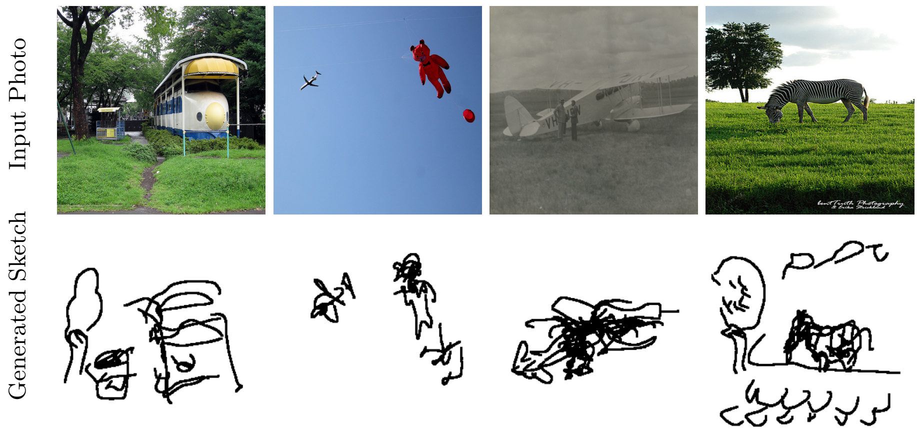


Fig. S6: Photo to vectored sketch synthesis: Our novel dataset allows interesting downstream applications such photo to scene vector sketch synthesis as a byproduct of our hierarchical decoder. Here, we show qualitative results using VGG-16 encoder followed by the hierarchical decoder.

8. Lin, H., Fu, Y., Jiang, Y.G., Xue, X.: Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. In: CVPR (2020) 15
9. 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) 10, 15
10. Mahajan, S., Gurevych, I., Roth, S.: Latent normalizing flows for many-to-many cross-domain mappings. In: ICLR (2020) 16
11. Noris, G., Sýkora, D., Shamir, A., Coros, S., Whited, B., Simmons, M., Hornung, A., Gross, M., Sumner, R.: Smart scribbles for sketch segmentation. Comp. Graph. Forum **31**(8) (2012) 3
12. Qi, Y., Su, G., Chowdhury, P.N., Li, M., Song, Y.Z.: Sketchlattice: Latticed representation for sketch manipulation. In: ICCV (2021) 13, 15
13. 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) 13, 15, 16
14. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: Learning to retrieve badly drawn bunnies. ACM Trans. Graph. (2016) 14
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015) 12, 13
16. 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) 12, 15, 16
17. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016) 13
18. 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) 16

19. 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) 16
20. Yu, Q., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: CVPR (2016) 12
21. Yu, Q., Yang, Y., Song, Y.Z., Xiang, T., Hospedales, T.: Sketch-a-net that beats humans. In: BMVC (2015) 12
22. 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, 3, 4, 10, 15