The Visual Computer
https://doi.org/10.1007/s00371-022-02731-8

ORIGINAL ARTICLE

Stroke-based semantic segmentation for scene-level free-hand sketches

Zhengming Zhang[1,2] · Xiaoming Deng[1,2] · Jinyao Li[1,2] · Yukun Lai[3] · Cuixia Ma[1,2] · Yongjin Liu[4] · Hongan Wang[1,2]

Accepted: 14 November 2022

Abstract

Sketching is a simple and efficient way for humans to express their perceptions of the world. Sketch semantic segmentation plays a key role in sketch understanding and is widely used in sketch recognition, sketch-based image retrieval, or editing. Due to modality difference between images and sketches, existing image segmentation methods may not perform best, which overlook the sparse nature and stroke-based representation in sketches. The existing sketch semantic segmentation methods are mainly designed for single-instance sketches. In this paper, we present a new stroke-based sequential-spatial neural network (S^3NN) for scene-level free-hand sketch semantic segmentation, which leverages a bidirectional LSTM and graph convolutional network to capture the sequential and spatial features of sketches. In order to address the data lacking issue, we propose the first scene-level free-hand sketch dataset (SFSD). SFSD is composed of 12K sketch-photo pairs over 40 object categories, where the sketches were completely hand-drawn and each contains 7 objects on average. We conduct comparative and ablative experiments on SFSD to evaluate the effectiveness of our method. The experimental results demonstrate that our method outperforms state-of-the-art methods. The code, models, and dataset will be made public after acceptance.

Keywords Sketch dataset · Scene sketch · Free-hand sketch · Semantic segmentation

✉ Cuixia Ma
cuixia@iscas.ac.cn

Zhengming Zhang
zhangzhengming16@mails.ucas.ac.cn

Xiaoming Deng
xiaoming@iscas.ac.cn

Jinyao Li
ljyyolia@gmail.com

Yukun Lai
LaiY4@cardiff.ac.uk

Yongjin Liu
liuyongjin@tsinghua.edu.cn

Hongan Wang
hongan@iscas.ac.cn

[1] State Key Laboratory of Computer Science and Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, Beijing, China

[2] University of Chinese Academy of Sciences, Beijing, China

[3] Cardiff University, Cardiff, UK

[4] Tsinghua University, Beijing, China

1 Introduction

Sketching is one of the most important ways for humans to depict intents. Compared to images and text, sketches are more concise and can convey richer information. Thanks to the rapid development and popularity of stylus and touch screen devices, people can get access to free-hand sketch with more convenience. Sketch-based interactive applications have also emerged, such as daily tools (flowcharts and mind maps drawing) and software for more specialized work (industrial and mechanical design). These applications bring more fine-grained requirements on sketch operations.

Sketch semantic segmentation (SSS) is a fundamental problem in sketch understanding. SSS aims to assign strokes in sketch with certain semantic labels. According to the segmentation granularity and types of semantic labels, SSS can be divided into scene and object levels (Fig. 1). In scene-level segmentation, prior art methods [1] migrated the models in image domain to sketch domain for feature extraction. However, directly using image semantic segmentation for sketch ignores the strong temporal sequential context among strokes in hand-drawn sketch, because strokes belonging

Published online: 07 December 2022

to the same object are likely to be drawn in close proximity (see visualization of stroke IDs in Fig. 7). Besides, sketch has the characteristic of sparsity, and an ideal visual feature encoder is expected to leverage the sparsity characteristic. In order to address the above two issues, we utilize a stroke-based method for scene-level semantic segmentation. The input of our method is stroke sequences that are stored in a vector format. Although there are a few single-object sketch datasets annotated with drawing strokes, no scene-level sketch datasets are available so far.

The past decade has witnessed the construction of many sketch datasets. Early efforts [2,3] collected hand-drawn sketches of single objects. With tasks such as cross-modal retrieval and generation being proposed, subsequent work improved the construction of sketch datasets from two aspects. (1) Transition from unimodal to multi-modal. Other modalities such as real photos were adopted to establish inter-modal correspondences. (2) Lifting from single objects to multiple objects (scene-level). Scene-level sketches can describe rich scene details and this is consistent with the fact that realistic pictures usually contain multiple objects. Due to the time-consuming efforts of sketching multiple objects, existing work [1,4] mainly achieves the goal by combining existing single-object sketches. Compared to fully hand-drawn sketches, the sketches obtained by the above combination approach may lack certain scene context and variety. Moreover, the simple drag-and-drop operation disables the collection of stroke order. Therefore, in this paper, we construct the first scene-level free-hand sketch dataset (SFSD), which integrates multiple objects, free-hand sketches, sketch-photo pairs, and vector format storage in one sketch dataset.

Based on SFSD, we design a stroke-based sequential–spatial neural network (S^{3}NN) for scene-level SSS. Compared to images, sketches are highly sparse, and their appearance is dominated by outlines and edges. The key challenges of SSS lie in the sparseness and diversity of sketches. Thanks to the vector format of SFSD, we can easily extract each stroke and drawing order of a sketch. The stroke sequence representation of scene sketch reduces the sparsity issue of sketch. In order to extract the diverse feature of sketch, we integrate visual, sequential, and spatial information in S^{3}NN. Specifically, a pre-trained convolutional neural network (CNN) is utilized to extract the overall visual feature of each stroke. The sequential relationship of strokes and the spatial connection between neighboring strokes are then learned by a recurrent neural network (RNN) and a graph convolutional network (GCN).

Our main contributions can be summarized as follows:

- We built the first scene-level free-hand sketch dataset (i.e., SFSD) in vector format, which contains more than 12 thousand sketch-photo pairs. SFSD can facilitate the research and evaluation of stroke-based neural models.
- To the best of our knowledge, we are the first to conduct scene-level stroke-based sketch semantic segmentation. To tackle the challenges of sparseness and diversity in sketches, the proposed model incorporates visual, sequential, and spatial features of stroke sequences.
- Experiments on SFSD demonstrate that our segmentation model outperforms the state of the art (SOTA).

2 Related work

2.1 Sketch datasets

Several sketch datasets have been presented in the past decade to promote various sketch applications. Table 1 summarizes the representative datasets and our SFSD dataset. TU-Berlin [2] is the first large-scale sketch dataset, which consists of 20K sketches over 250 categories. QuickDraw [3] is a large dataset that includes 50M sketches across 345 categories. Both TU-Berlin and QuickDraw are single-modal free-hand sketch datasets, which are collected with vector storage formats and facilitate sketch editing. They are widely used in sketch recognition and text-sketch retrieval. Sketchy [5] and QMUL-Shoe-V2 [6] are two multi-modal single-object sketch datasets with sketch-photo pairs. SketchyScene [1] and SketchyCOCO [4] contribute scene-level sketch datasets with multiple foreground or background objects. However, these scene sketches are obtained by compositing single-instance sketches and are stored in image format. The category ‘Person’ is very common for many computer vision researches and applications. However, previous sketch datasets hardly included ‘Person’ as one of the categories due to the diversity of human, especially, varied poses, shapes, and actions of different subjects. SketchyScene [1] is the only dataset that also contains the category ‘Person’ of cartoon characters which are different to hand-drawn sketches in stroke and appearance style. In this work, we present the SFSD dataset featuring vector storage format, free-hand drawing, scene-level objects, sketch-photo pairs, and human categories, which can benefit sketch retrieval or editing researches.

2.2 Sketch semantic segmentation

Early efforts often use low-level geometric features [7,8] and traditional machine learning methods [9–12] to predict the categories that strokes in a sketch belong to. While some results could be achieved, these methods highly rely on specific input format and are time-consuming. Following the flourishing of deep learning, various neural network archi-

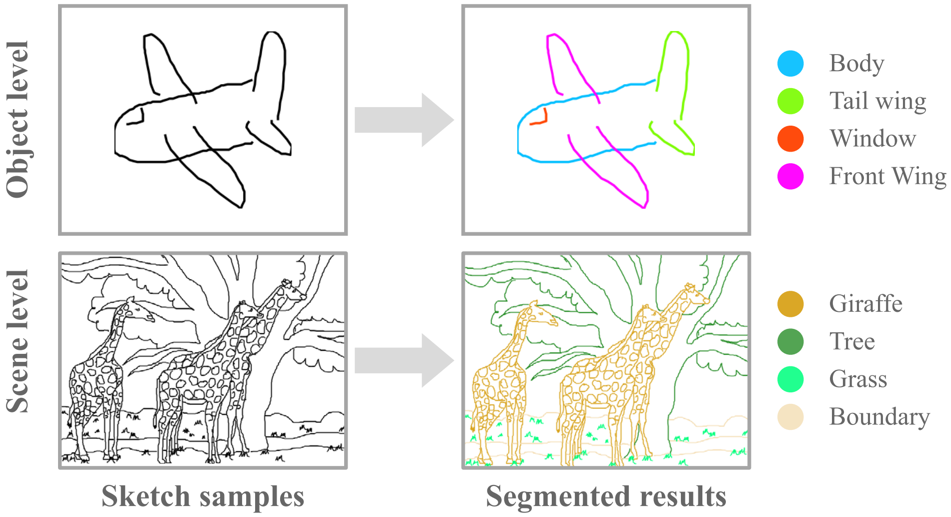

Fig. 1 Illustration of object-level and scene-level sketch semantic segmentation. Scene-level sketch segmentation aims to predict class label of each stroke in scene sketch, which outperforms object-level segmentation of a large margin in the aspect of semantic context

Table 1 Summary of representative sketch datasets and our SFSD dataset

Dataset	Sketch amount	Vector	Free-hand	Sketch-photo pair	Scene	Person
TU-Berlin [2]	20K	✓	✓			
QuickDraw [3]	50M+	✓	✓			
Sketchy [5]	75K	✓	✓	✓		
QMUL-Shoe-V2 [6]	700	✓	✓	✓		
SketchyScene [1]	7K+			✓	✓	✓
SketchyCOCO [4]	14K+			✓	✓	
SFSD (Ours)	12K+	✓	✓	✓	✓	✓

tectures are used for SSS, including CNN-based methods [13–16], and RNN-based methods [17–21].

CNN-based models treat SSS as an image segmentation task and pay more attention to the edge and outline features. Since a sketch is drawn by stroke sequences, sequence modeling of sketch strokes is a promising solution for SSS. RNN-based models extract the sequential features of stroke points. Besides the above visual and sequential features, the spatial relationship between strokes is also useful for SSS. Since graph-based networks can learn structural relationships effectively, some efforts use graph neural networks for single-object SSS [22,23]. In this paper, we adopt a hybrid architecture of CNN, RNN, and GCN to capture multi-scale sketch features, and conduct stroke-based multi-object SSS.

3 The SFSD dataset

SFSD has the characteristics of scene-level, completely freehand, multi-modal, and vector storage data format. It includes more than 12 thousand pairs of photo and sketch over 40 categories. The reference photos were selected from MS COCO [24]. Figure 2 shows 44 sketch-photo pairs from the proposed SFSD, where the annotation of sketches is instance level. All the 40 categories are included in the figure. Since MS COCO provides the textual description of each photo, we can even carry out cross-modal research upon SFSD. In addition to the semantic segmentation addressed in this paper, SFSD can also support retrieval, generation, and other sketch-related tasks as well. In this section, we introduce the process of dataset construction, which can be summarized into three phases, i.e., image preparation, sketch collection, and sketch annotation. Next, we report some statistics and analysis on SFSD.

3.1 Dataset construction

3.1.1 Image preparation

MS COCO dataset [24] includes 328K photos with 2.5M labeled instances. Considering the large volume of MS COCO, it is not realistic to sketch all the photos in the dataset. Besides, not all pictures are suitable for sketching. For example, a photo of a man feeding hundreds of pigeons has too many objects and it takes lots of effort to sketch the scene. To filter the photos, we first excluded those with more than 10 objects. Then, we manually selected the photos by the following criteria. (1) The scenes are restricted to wildlife, outdoor sports, and out-of-town streets. Other indoor and urban street scenes may contain too many trivial objects (some objects

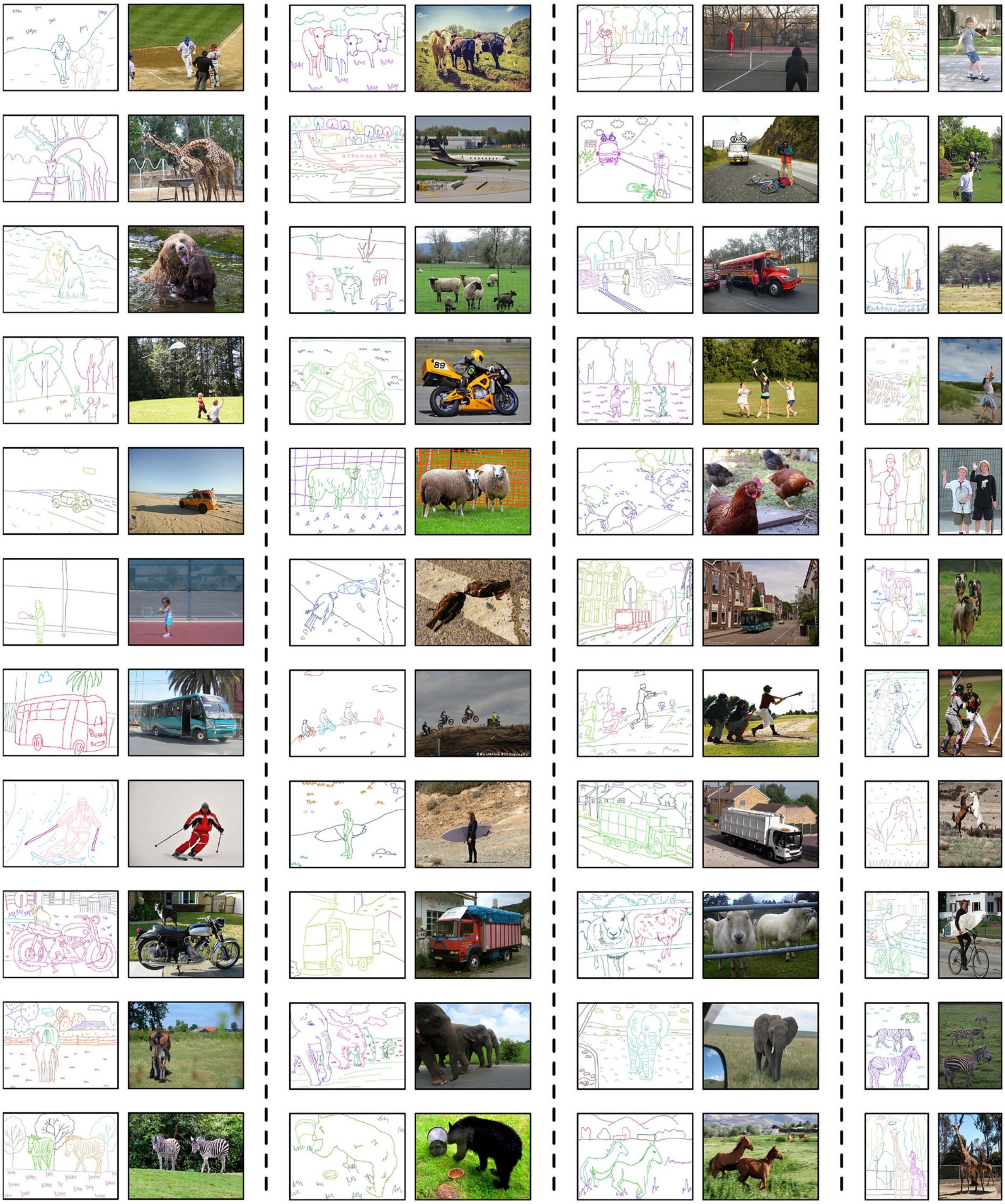

Fig. 2 Example sketch-photo pairs in SFSD which contain objects of all 40 categories. The sketches shown were annotated at the instance level. We can observe that the dataset is diverse in terms of object categories, sketch complexity, and drawing quality

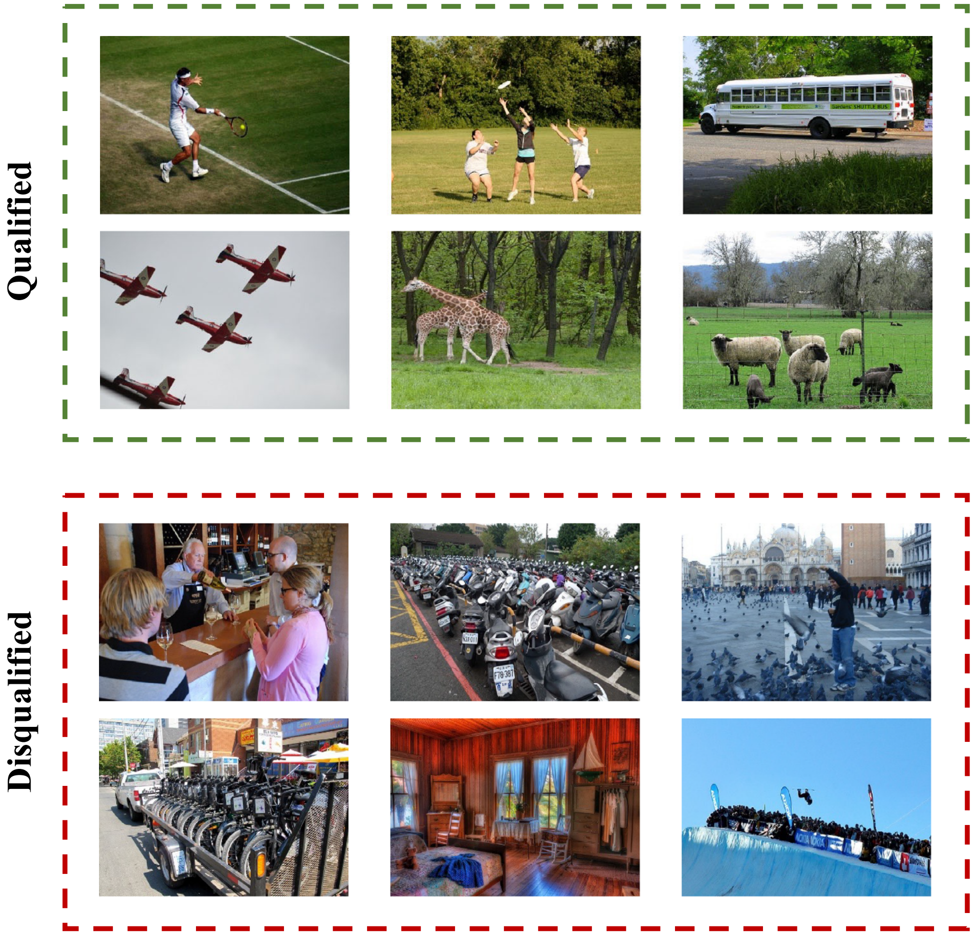

Fig. 3 Samples of qualified and disqualified photos during image selection process. These photos are taken from MS COCO

are difficult to identify even for humans after conversion to sketches) and the background may be hardly complete. (2) The photos have high integrity, moderate background complexity, and objects that are relatively easy to identify and draw. We recruited some participants to conduct a pre-experiment and then came to the above conclusion. In this way, we finally selected 12,115 pictures from MS COCO as reference photos for our SFSD. Figure 3 displays samples of selected qualified and disqualified images.

3.1.2 Sketch collection

We recruited 40 participants with different levels of painting skill. 1600 hours were spent in total to accomplish 12 thousand sketches. In order to standardize the process of drawing, we established an online sketching system to collect stroke sequences. We mainly collected the absolute coordinates of drawing track with a sampling rate of 120 Hz. Each stroke is represented by a sequence of two-dimensional coordinates, and each sketch is composed of a stroke sequence. Considering the multi-object characteristic of sketches in SFSD, we paid more attention to the overall layout and coordination between different parts of scene sketches. Instead of overlapping the panels of sketch and reference photo and allowing for direct tracing of the outlines as prior work [6], we placed the reference image on the left side of the drawing board and asked participants to give full play to their drawing ability. This setting enhanced the diversity of sketches for each individual object. In order to ensure the dataset to follow uniform standards, we adopt manual verification to discard sketches if the main objects cannot be identified by more than one person.

3.1.3 Sketch annotation

We deployed a sketch annotation system to annotate SFSD. Another group of participants were employed to finish the sketch annotation. Each stroke was assigned with certain background or foreground categories. Attributes like drawing completeness and similarity of all objects are also recorded for future work. The quality inspection of sketch includes two aspects, the drawing quality of sketches and the correctness of annotation. The quality metric of sketch includes overall legibility, sketch-photo matching degree, and object details. The annotation quality inspection aims to correct labeling errors of sketch strokes.

Table 2 Comparison and statistics of scene sketch datasets

Dataset	Categories	Sketches per category			Categories per sketch			Objects per sketch			Strokes per sketches		
		Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean
SketchyScene [1]	46	5723	31	1087.02	19	3	6.88	94	3	16.71	–	–	–
SketchyCOCO [4]	17	9051	33	1825.06	6	1	2.33	35	2	10.93	–	–	–
SFSD (Ours)	40	6429	141	1351.95	11	1	4.46	43	2	7.76	699	9	146.64

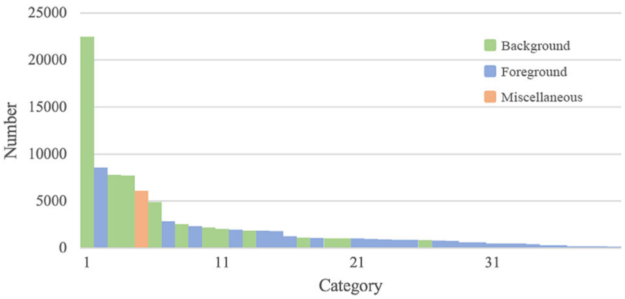

Fig. 4 Diagram of instance frequency distribution

3.2 Statistics and analysis

Table 2 shows comparison of different sketch components with existing scene sketch datasets, ranging from strokes, objects to categories. Our dataset contains 40 categories, more than twice the number of categories in SketchyCOCO, which also referenced real images. In our dataset, sketches contain an average of 146 strokes, which is much higher than previous single-object sketch dataset and can describe more details of the objects. Moreover, to the best of our knowledge, previous scene sketch datasets do not contain stroke order information.

The number of annotated instances in each background and foreground category can be found in Table 4. There are 12 background classes, 27 foreground classes, and 1 miscellaneous class (other). The total number of objects is 94,037. In other words, we contributed a large number of single-object sketches since the annotation is instance level. Due to the frequent occlusion problems in real photos, the dataset contains a large number of incomplete sketches, which can be used for tasks like sketch completion. During the image selection process, we did not prefer any specific category. Naturally, an obvious long-tail distribution can be observed on the instance frequency (Fig. 4). As the focus of segmentation, foreground categories are mainly concentrated in the long-tail section, which increases the difficulty of SSS but is more in line with practical applications.

4 Methodology

The overview of proposed S^{3}NN is illustrated in Fig. 5. Given an input scene sketch, we first compute statistical parameters (i.e., length, drawing duration, and bounding box) for each stroke as its global features. Then, we feed the image patch corresponding to the bounding box of each stroke into a pre-trained CNN to extract the primary visual features of the stroke. The above two stroke features are concatenated and fed to subsequent modules, sequential encoder (SeqE) and spatial encoder (SpaE). SeqE utilizes bidirectional LSTM (BiLSTM) to extract temporal features, and SpaE leverages the spatial context modeling ability of graph convolutional network (GCN) to extract spatial features. Finally, we feed the extracted temporal/spatial features into a fully connected layer with softmax to predict the class label of each stroke.

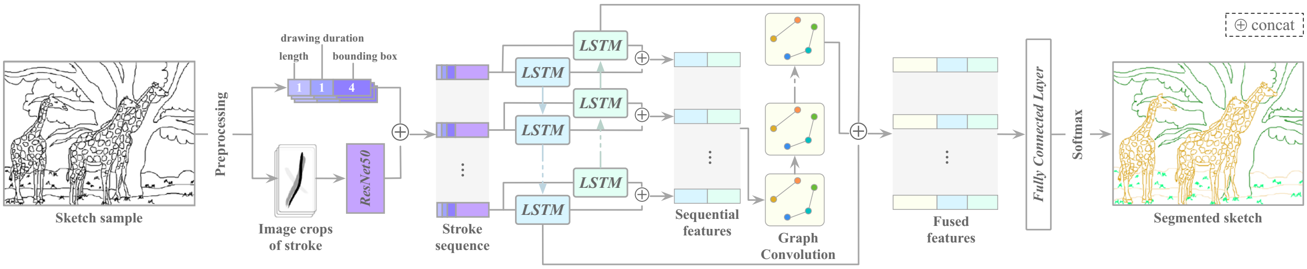

Fig. 5 The framework of S^{3}NN. For a sketch sample, the preprocessing includes computing statistic features and capturing visual features of each stroke via ResNet50. The concatenated sequence feature is cascadingly fed into the sequential encoder (SeqE) for temporal relationship extraction and spatial encoder (SpaE) for spatial connection learning. Finally, the fusion of spatial and global sequential features is mapped to 40 categories. Classification is conducted by the softmax probabilities

4.1 Input representation

A scene-level sketch contains a certain number of strokes. Each stroke S can be represented by a point sequence $[(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)]$, where (x_k, y_k) are the coordinates of the kth point and n is the number of points in a stroke. The feature of the ith stroke $\mathbf{f}_i = \text{concat}(f_i^{\text{len}}, f_i^{\text{dur}}, \mathbf{f}_i^{\text{box}}, \mathbf{f}_i^{\text{cnn}})$ can be obtained by concatenating four types of features. (1) A scalar of stroke length f^{len}, i.e., the sum of Euclidean distances between each pair of adjacent points. (2) A scalar of drawing duration f^{dur}, which indicates the time spent to draw a particular stroke. (3) 4D vector of stroke bounding box $\mathbf{f}^{\text{box}}$. (4) 256D visual feature $\mathbf{f}^{\text{cnn}}$ obtained by feeding image crop of stroke into a pre-trained CNN for feature extraction. We obtained the image region of each stroke by converting a sketch from vector format into image format and cropping the bounding box area of the corresponding stroke in the image. Finally, the sketch features $\mathcal{F}$ can be obtained by $\mathcal{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m]$, where m is the number of strokes in the scene sketch.

4.2 Sequential encoder

In free-hand sketching, the sequence of strokes can convey clues of human sketching mechanism, and plays a crucial role in the understanding of sketches. Strokes belonging to the same object are found likely to be drawn in close proximity, so it is a key problem to effectively incorporate this sequential context into feature learning of strokes. BiLSTM [25] built upon LSTM can effectively model temporal sequential context of the past or future in sketching by learning long-term memory and short-term memory. In this paper, we utilize BiLSTM for the sequential encoder of strokes. Although other RNNs can be alternatives, experiments demonstrate that BiLSTM is more effective. The forward and backward modules of BiLSTM can be formulated as follows

$$\mathcal{L}_{\text{f}}([\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m]) = [\overrightarrow{\mathbf{h}_1}, \overrightarrow{\mathbf{h}_2}, \ldots, \overrightarrow{\mathbf{h}_m}] \in \mathbb{R}^{d_{\text{h}} \times m} \tag{1}$$

$$\mathcal{L}_{\text{b}}([\mathbf{f}_m, \mathbf{f}_{m-1}, \ldots, \mathbf{f}_1]) = [\overleftarrow{\mathbf{h}_1}, \overleftarrow{\mathbf{h}_2}, \ldots, \overleftarrow{\mathbf{h}_m}] \in \mathbb{R}^{d_{\text{h}} \times m} \tag{2}$$

where $\mathcal{L}_{\text{f}}$ and $\mathcal{L}_{\text{b}}$ denote the forward and backward LSTM operations, and d_{h} is the hidden unit dimension. The output of BiLSTM is $\mathbf{H}_t = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_m]$, where $\mathbf{h}_i = \text{concat}(\overrightarrow{\mathbf{h}_i}, \overleftarrow{\mathbf{h}_{m-i+1}})$. The hidden states will be used as the feature vector of nodes in the subsequent modules for spatial encoder and temporal features for stroke segmentation.

Table 3 Sketch semantic segmentation accuracy (%) on SFSD

Model	C-metric	P-metric	MIoU
FPN	75.84	74.06	40.01
DeepLabv3+	76.04	74.89	40.61
LDP	78.34	76.40	42.79
Sketch-RNN	68.56	66.70	28.62
SketchGNN	57.04	56.56	21.37
Ours w/o SeqE	78.74	75.73	41.77
Ours w/o SpaE (BiLSTM)	76.62	73.26	40.04
	71.08$^{\triangle}$	64.86$^{\triangle}$	30.99$^{\triangle}$
Ours w/o SpaE (LSTM)	74.37	70.40	38.95
Ours w/o fusion	80.14	77.35	44.39
Ours w/o w_c	80.61	77.37	44.61
Ours	**80.72**	**77.65**	**45.34**
	78.38$^{\triangle}$	73.85$^{\triangle}$	39.70$^{\triangle}$

The results marked with $\triangle$ are evaluated based on the test set with shuffled strokes

Bold values indicate a method achieves the best performance compared with other methods in a certain evaluation metric

4.3 Spatial encoder

A complete sketch can be seen as the integration of multiple strokes. The combination of stroke position and shape conveys semantic information. There is uncertainty in the reliability of sequential features, e.g., two temporally adjacent strokes may belong to the end of one object and the start of another object, respectively. In order to compensate for the probably of wrong classification caused by SeqE, we further consider spatial information in this module. Taking each stroke as a node, SpaE mainly learns the correlations between different strokes at spatial level by GCN. Given a scene sketch, we construct a scene sketch graph $G = (V, E)$ to extract spatial features of strokes, where $V = \{v_i\}$ and $E = \{e_{ij}\}$ are vertices and edges of graph G, respectively. Vertex v_i denotes stroke S_i, and an edge e_{ij} links each pair of vertices and denotes the spatial correlation between strokes S_i and S_j.

Given two vertices v_i and v_j of the graph, we define an edge $e_{ij} \in \{0, 1\}$ according to their spatial proximity, i.e., $e_{ij} = 1$ if the bounding box $B(S_i)$ of stroke S_i contains part of stroke S_j or vice versa

$$e_{ij} = \begin{cases} 1 & B(S_i) \cap b(S_j) \neq \varnothing \text{ or } B(S_j) \cap b(S_i) \neq \varnothing \\ 0 & \quad\text{otherwise} \end{cases} \tag{3}$$

where $B(\cdot)$ is the bounding box of a stroke, and $b(\cdot)$ is the set of points in a stroke. $\mathbf{E}$ is the matrix that represents edges.

For each vertex, we get a fused feature $\mathbf{h}_i$ by concatenating forward and backward sequential features of stroke S_i. To extract spatial features among strokes, we adopt four graph convolution layers similar to [26] to learn spatial features $\mathbf{P}^{(l+1)}$ by propagating features between adjacent vertices, where we input the feature $\mathbf{P}^{(l)}$ of the previous layer and the adjacency matrix. Formally,

$$\mathbf{P}^{(0)} = \{\mathbf{h}_i\}_{i=1}^{m} \tag{4}$$

$$\mathbf{P}^{(l+1)} = \text{ReLU}(\tilde{\mathbf{A}}\mathbf{P}^{(l)}\mathbf{W}^{(l)}) \tag{5}$$

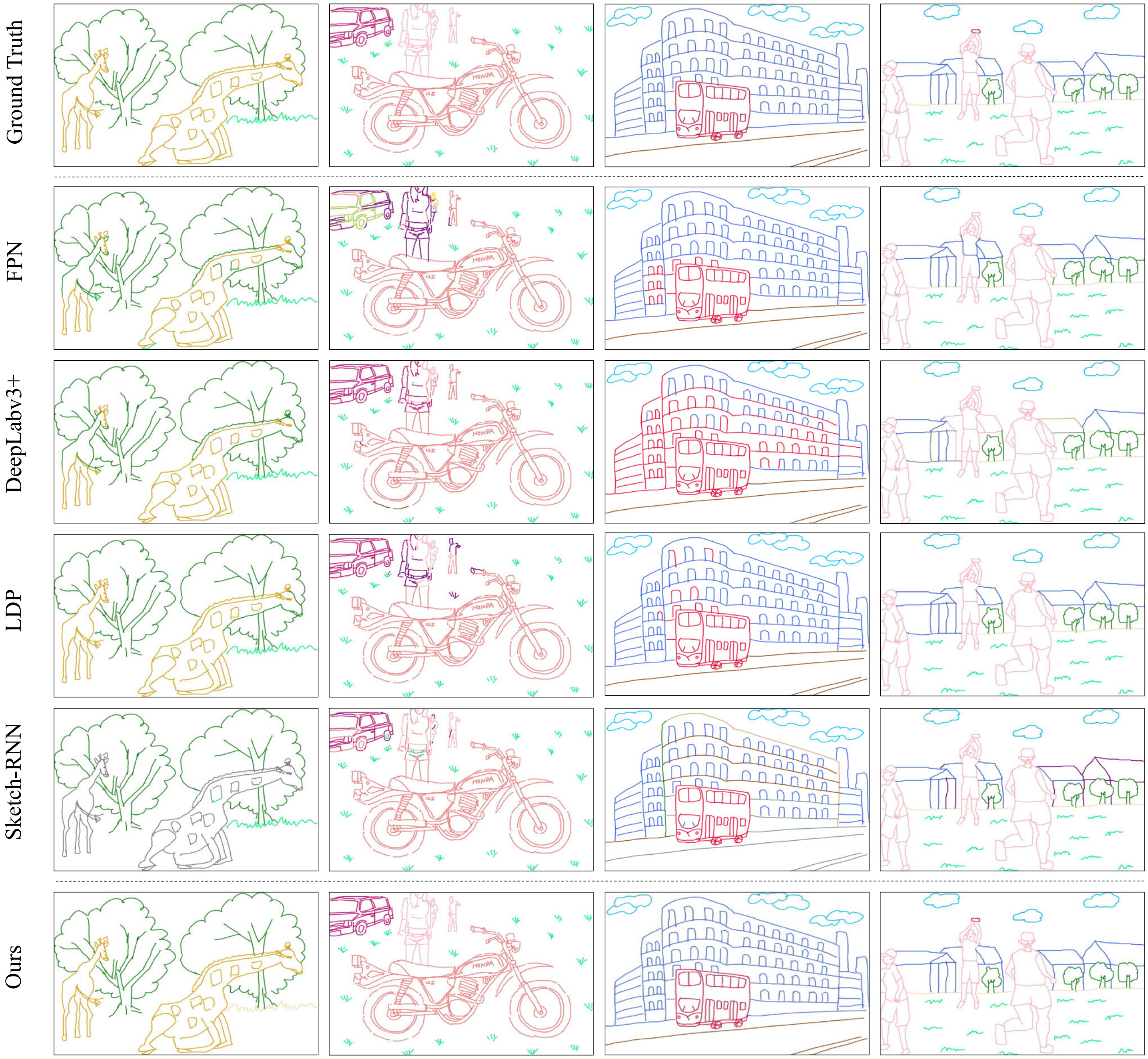

Fig. 6 Visualization of representative segmentation results by the SOTA methods and our model

where $\tilde{\mathbf{A}} = \mathbf{E} + \mathbf{I}$ is the adjacency matrix, $\mathbf{I}$ is an identity matrix, and $\mathbf{W}^{(l)}$ is a learnable weight matrix.

4.4 Stroke segmentation

After we conduct the above two encoders, we fuse the learned sequential and spatial features of strokes, which can be used to predict the class label of each stroke. Specifically, we first get the fused feature $\mathbf{R}_i$ by concatenating the output feature of the GCN's last layer and two global features of BiLSTM since the transformation of GCN may lead to loss of sequential information. Then, $\mathbf{R}_i$ is further fed into a fully connected layer and softmax for stroke classification. Formally,

$$\hat{Y}_i = \text{softmax}(fc(\mathbf{R}_i)) \tag{6}$$

$$\mathbf{R}_i = \text{concat}(\mathbf{P}, \overrightarrow{\mathbf{h}_m}, \overleftarrow{\mathbf{h}_m}) \tag{7}$$

where $fc(\cdot)$ is the fully connected layer.

4.5 Loss function

We adopt a cross-entropy loss function for sketch stroke segmentation as follows

$$\text{Loss} = -\frac{1}{m}\sum_{i=1}^{m} w_c \cdot Y_i \cdot \log(\hat{Y}_i) \tag{8}$$

where Y_i is the ground truth label, and $\hat{Y}_i$ denotes the probability of the stroke segmentation prediction. In order to address the long-tailed distribution of each class, we adopted a weight w_c for each class c, computed as the ratio of the median of class frequencies and class frequency of c. Therefore, less frequent categories have higher weight.

5 Experiments

5.1 Baselines and implementation details

We use five SOTA baselines for comparison, including FPN [27], DeepLabv3+ [28], LDP [29], Sketch-RNN [3], SketchGNN [22]. DeepLabv3+ and FPN are commonly used image semantic segmentation baselines. DeepLabv3+ is the extension of DeepLabv3 [30]. FPN is a feature pyramid network for semantic segmentation, which was the winning entry of COCO stuff 2017 competition. LDP is a scene sketch segmentation method by enhancing local detail perception. Sketch-RNN was originally designed for sketch generation. We utilized its encoder to perform SSS. SketchGNN uses a well-designed GCN for object-level sketch semantic segmentation.

We evaluated the baselines and our models on the proposed SFSD. Experiments were not done on other datasets since SFSD is the first scene-level sketch dataset in vector format and our model is stroke-based. We split 12,115 sketches into 9115 for training and the remaining 3000 for testing. We converted the sketches into images and generated masks according to the semantic annotations as input for FPN, DeepLabv3+, and LDP. ResNext50 and ResNet50 are used as the backbone networks of FPN and DeepLabv3+, respectively. For Sketch-RNN, we followed the input format proposed by [3] and transformed each stroke point into a 5D vector, i.e., $[\Delta x_i, \Delta y_i, p_1, p_2, p_3]$. For SketchGNN, we resampled the points to 2048 for each sketch as input. For sketches with less than 2048 points, we randomly interpolated the stroke points to 2048 points. For sketches with more than 2048 points, we searched for the strokes with the highest number of points at a time, and then deleted the point whose curvature is closest to 180 degrees to the adjacent points. In our method, SpaE's vertex feature of all layers is 256D. We apply the Adam optimizer for optimization and set the learning rate to 0.001. All models are trained on a single GeForce RTX 3090 for 150 epochs.

5.2 Evaluation metrics

We evaluate the performance of different methods using three standard metrics as [11,17,29].

Pixel-based accuracy (P-metric) indicates the percentage of correctly classified pixels to pixels of all sketches.

Table 4 Number of annotated instances and segmentation accuracy by the proposed S^{3}NN method for forty object categories in SFSD dataset

Category	No.	C-m	P-m	Category	No.	C-m	P-m	Category	No.	C-m	P-m	Category	No.	C-m	P-m
Tree*	22494	0.89	0.91	Fence*	2037	0.54	0.56	Tennis racket	1042	0.71	0.70	Snowboard	494	0.17	0.16
Person	8572	0.97	0.97	Cow	1959	0.62	0.62	Horse	990	0.55	0.60	Truck	488	0.43	0.45
Cloud*	7784	0.94	0.96	Stone*	1857	0.58	0.51	Bus	902	0.81	0.82	Motorcycle	481	0.74	0.76
Grass*	7706	0.96	0.96	Sheep	1856	0.76	0.72	Bird	874	0.57	0.57	Frisbee	412	0.38	0.51
Others#	6077	0.31	0.31	Elephant	1812	0.88	0.86	Skis	869	0.41	0.40	Dog	319	0.17	0.21
Boundary*	4886	0.49	0.51	Airplane	1277	0.88	0.89	River*	826	0.79	0.81	Bear	319	0.23	0.23
Zebra	2847	0.98	0.98	Playground*	1136	0.31	0.37	Skateboard	814	0.63	0.58	Backpack	199	0.06	0.07
Road*	2571	0.41	0.49	Car	1103	0.40	0.37	Sports ball	746	0.53	0.57	Surfboard	194	0.22	0.35
Giraffe	2335	0.98	0.97	Mountain*	1078	0.38	0.50	Baseball bat	615	0.39	0.43	Kite	180	0.17	0.14
house*	2179	0.60	0.57	Snowfield*	1062	0.80	0.76	Baseball glove	607	0.29	0.27	Bicycle	176	0.61	0.58

C-m and P-m represent the criteria of C-metric and P-metric separately. The categories marked with * indicate background objects. The marker # is miscellaneous category. Those without special symbols are foreground objects

Component-based accuracy (C-metric) evaluates the percentage of correctly classified strokes to total strokes. A stroke label is determined by its most frequent pixel label.

Mean Intersection over Union (MIoU) evaluates the average of the ratios between the intersection and the union of ground truth and predicted labels over all classes.

5.3 Comparison to state-of-the-art methods

As shown in Table 3, our model outperforms the compared baselines. Our full model achieves performance gain by 2.38% on C-metric, 1.25% on P-metric, and 2.55% on MIoU than LDP, which is the best performing model in all baselines. Even our model without the SpaE or SeqE module achieves higher accuracy than DeepLabv3+. FPN and DeepLabv3+ perform closely with accuracy of around 75%, which indicates that they are saturated using only visual features. Our network also performs much better than Sketch-RNN. Sketch-RNN was originally designed for single-object sketches. When it is applied to a scene-level sketch with multiple objects, the patterns of input stroke sequences may be too complex for Sketch-RNN to learn. Similarly, SketchGNN was originally designed for single-object sketch segmentation, which is much simpler than scene-level sketch segmentation. However, the scene-level sketch contains more complex semantic and structural information, which makes the single-object approach SketchGNN hard to perform well.

Figure 6 shows the qualitative comparison of segmentation results of sketch examples. We can observe that our model performs better, especially in the cases of occlusive, overlapping regions. In the third sketch, the bus and the building are overlapped. FPN, DeepLabv3+ and LDP label part of the building as bus. In the forth sketch, the person in the middle has a small frisbee attached to his hands, which is easily classified into the person category. Only our model identifies the frisbee. By checking the stroke sequence, we found that although these objects (the building and the bus, or the frisbee and the person) are spatially close, they are far away in temporal sequential orders. Conceptually, the performance gain of our method could be due to stroke representation of sketch and the temporal context of stroke sequences.

Table 4 shows the detailed segmentation performance of our method on all the 40 categories. Our method achieves competitive segmentation performance for object categories with large numbers of instances, and provides a baseline model for scene-level stroke-based SSS. Although promising results are achieved, we observe two types of categories with poor segmentation performance for future improvement: (1) objects with few occurrences, such as dogs and kites; (2) small objects attached to large objects (i.e., human), such as backpacks and baseball gloves. However, these are also common issues for image semantic segmentation.

5.4 Ablation study

Effect of SeqE As shown in Table 3, after removing SeqE, the performance drops by 1.98% on C-metric, 1.92% on P-metric, and 3.57% on MIoU. SeqE introduces the pattern of stroke drawing orders and enables S^{3}NN to cope with some otherwise intractable cases, e.g., occlusion, overlap. To further validate the effectiveness of BiLSTM in SeqE, we replaced BiLSTM with LSTM and observed a decrease of 2.25% on C-metric and 2.86% on P-metric. As shown in the second row of Fig. 7, the strokes of skateboard are spatially separated but temporally close due to continuous stroke ID of skateboard. Our model without SeqE wrongly labels the right part of the skateboard as a frisbee. After incorporating SeqE, the temporal correlation of these two parts of skateboard is utilized, and the skateboard can be segmented correctly. We can also observe, due to the similarity of stripe patterns of the boy's shoes and zebra, our model without SeqE is confused to recognize the boy's shoes as zebra. However, by leveraging sequential correlation of strokes with SeqE, our full model can achieve correct segmentation results. Therefore, SeqE is effective for stroke-based scene-level SSS.

Effect of SpaE As shown in Table 3, without SpaE, the accuracy drops 4.10% on C-metric, 4.39% on P-metric, and 5.30% on MIoU, which indicates the importance of this module. During the prediction, SpaE tends to group spatially close strokes and can correct part of the segmentation error due to stroke temporal order. As shown in Fig. 7, we can observe that there are temporal gaps in drawing order between the strokes of elephants' body and leg, and the strokes of the each elephant. SeqE tends to label the temporal separated strokes as another object. However, SpaE exploits the spatial correlation of stroke and can enhance the segmentation results.

Effect of feature fusion To validate the effects of global temporal feature in Eq. 7, we built a degraded model by feeding the output feature of GCN's last layer into the fully connected layer for prediction. As shown in Table 3, our full model achieves 0.58% higher on C-metric, 0.30% higher on P-metric, and 0.95% higher on MIoU. Therefore, the feature fusion has positive impacts on the stroke-based semantic segmentation task.

Effect of class-aware loss weight w_c The long-tail distribution of SFSD's instance frequency results in the difficulty of making a balanced learning between different categories. In order to tackle the above issue, we introduce a different weight w for each category in Eq. 8. The effect of them was tested by removing w from the loss function. From Table 5, we can see that the overall effect is limited, but the improvement on some low-frequency categories is promising.

Robustness to stroke orders We shuffle the strokes of sketches in the testset for 10 times, perform the semantic segmentation, and compute the average evaluation metrics

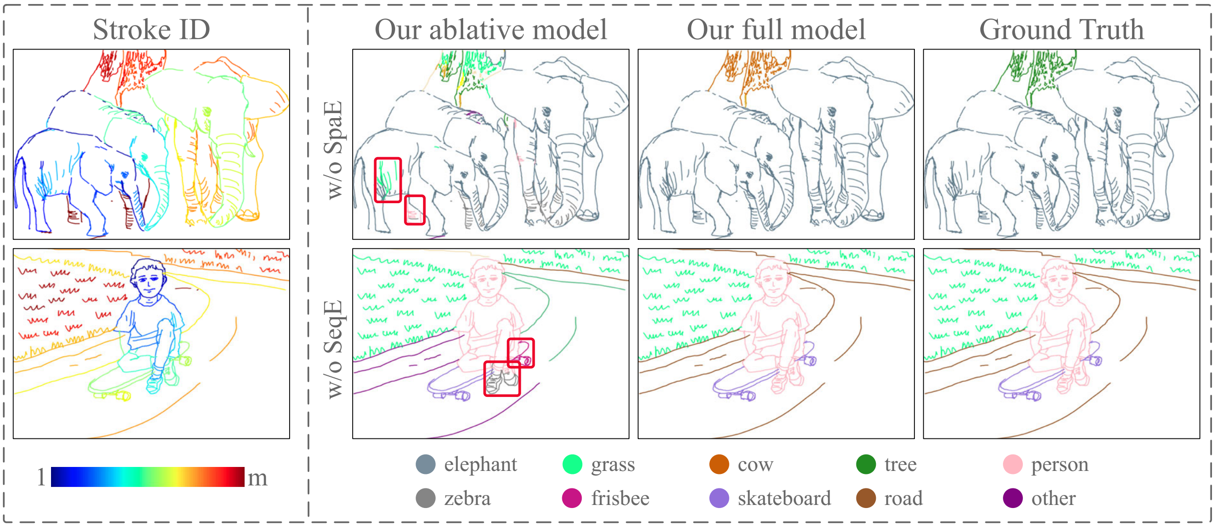

Fig. 7 Visualization of drawing orders and segmentation results for ablation study. The legends represent color encoding for stroke ID and object categories, and *m* is the stroke amount in a sketch. The red boxes highlight the wrongly labeled segmentation results with the degraded models and fixed by our full model

Table 5 Segmentation accuracy of each categories for ablation study of weight w_c in loss function

Category	C-m	P-m	Category	C-m	P-m	Category	C-m	P-m	Category	C-m	P-m
Tree*	0.92	0.94	Fence*	0.47	0.49	Tennis racket	0.70	0.69	Snowboard	0.08	0.07
Person	0.97	0.97	Cow	0.64	0.63	Horse	0.52	0.52	Truck	0.37	0.36
Cloud*	0.93	0.94	Stone*	0.57	0.50	Bus	0.81	0.84	Motorcycle	0.92	0.92
Grass*	0.94	0.95	Sheep	0.76	0.74	Bird	0.50	0.48	Frisbee	0.26	0.36
Others#	0.35	0.38	Elephant	0.85	0.83	Skis	0.40	0.42	Dog	0.09	0.11
Boundary*	0.41	0.43	Airplane	0.85	0.86	River*	0.77	0.80	Bear	0.25	0.26
Zebra	0.98	0.98	Playground*	0.27	0.31	Skateboard	0.67	0.62	Backpack	0.02	0.02
Road*	0.46	0.54	Car	0.37	0.37	Sports ball	0.45	0.45	Surfboard	0.14	0.17
Giraffe	0.97	0.96	Mountain*	0.33	0.45	Baseball bat	0.34	0.38	Kite	0.19	0.16
House*	0.58	0.55	Snowfield*	0.82	0.80	Baseball glove	0.19	0.18	Bicycle	0.49	0.50

C-m and P-m are obtained by the degraded model without the class-aware weight w_c

of semantic segmentation. As shown in Table 3, compared to evaluation with original strokes, the average accuracy of our S^{3}NN using shuffled strokes drops 2.34%, 3.80%, and 5.64% on the three metrics, and the model without SpaE drops 5.54%, 8.40%, and 9.05%. These results demonstrate that the stroke order affects the performance of SeqE, but SpaE can compensate for the performance drop. Therefore, S^{3}NN is robust to stroke orders.

6 Conclusion and future work

In this paper, we present SFSD, the first large-scale dataset of free-hand scene sketches. SFSD provides a large repository of scene and object sketches. Benefiting from SFSD, we propose an effective stroke-based model for scene-level SSS, which models multi-modal features, i.e., visual feature, sequential information, and spatial features. We conduct comparative experiments and ablative study on SFSD to evaluate the proposed model. Experiments demonstrate that our model outperforms the SOTA methods, and it can also handle challenging cases such as occlusion and overlap well.

Although our method can achieve promising results, it can be improved in the future work: (1) The stroke-based segmentation model can be further improved to handle corner cases. (2) SFSD is a multi-modal dataset, so it can enable more scene sketch researches such as sketch-based image retrieval and generation, and scene sketch generation.

Acknowledgements This work was supported by the Natural Science Foundation of China under Grant 61872346, Beijing Natural Science Foundation under Grant L222008, and 2019 China Prize of Newton Prize Project under Grant NP2PB/100047.

References

1. Zou, C. et al.: Sketchyscene: Richly-annotated scene sketches. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 421–436 (2018)
2. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Trans. Graph. **31**(4), 1–10 (2012)
3. Ha, D., Eck, D.A.: Neural representation of sketch drawings. In: International Conference on Learning Representations (ICLR) (2018)
4. Gao, C., et al.: Sketchycoco: image generation from freehand scene sketches. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5174–5183 (2020)
5. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. ACM Trans. Graph. **35**(4), 1–12 (2016)
6. Yu, Q., et al. Sketch me that shoe. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 799–807 (2016)
7. Delaye, A., Lee, K.: A flexible framework for online document segmentation by pairwise stroke distance learning. Pattern Recogn. **48**(4), 1197–1210 (2015)
8. Gennari, L., Kara, L.B., Stahovich, T.F., Shimada, K.: Combining geometry and domain knowledge to interpret hand-drawn diagrams. Comput. Graph. **29**(4), 547–562 (2005)
9. Sun, Z., Wang, C., Zhang, L., Zhang, L.: Free hand-drawn sketch segmentation. In: European Conference on Computer Vision (ECCV), pp. 626–639. Springer (2012)
10. Schneider, R.G., Tuytelaars, T.: Example-based sketch segmentation and labeling using CRFS. ACM Trans. Graph. **35**(5), 1–9 (2016)
11. Huang, Z., Fu, H., Lau, R.W.: Data-driven segmentation and labeling of freehand sketches. ACM Trans. Graph. **33**(6), 1–10 (2014)
12. Qi, Y., et al.: Making better use of edges via perceptual grouping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1856–1865 (2015)
13. Li, L., Fu, H., Tai, C.-L.: Fast sketch segmentation and labeling with deep learning. IEEE Comput. Graph. Appl. **39**(2), 38–51 (2018)
14. Wang, F., et al.: Multi-column point-CNN for sketch segmentation. Neurocomputing **392**, 50–59 (2020)
15. Zhu, X., Xiao, Y., Zheng, Y.: 2d freehand sketch labeling using CNN and CRF. Multimedia Tools Appl. **79**(1), 1585–1602 (2020)
16. Sarvadevabhatla, R.K., Dwivedi, I., Biswas, A., Manocha, S.: Sketchparse: towards rich descriptions for poorly drawn sketches using multi-task hierarchical deep networks. In: Proceedings of the 25th ACM International Conference on Multimedia, pp. 10–18 (2017)
17. Wu, X., Qi, Y., Liu, J., Yang, J.: Sketchsegnet: a RNN model for labeling sketch strokes. In: 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6. IEEE (2018)
18. Qi, Y., Tan, Z.-H.: Sketchsegnet+: an end-to-end learning of RNN for multi-class sketch semantic segmentation. IEEE Access **7**, 102717–102726 (2019)
19. Li, K., et al.: Toward deep universal sketch perceptual grouper. IEEE Trans. Image Process. **28**(7), 3219–3231 (2019)
20. Kaiyrbekov, K., Sezgin, M.: Deep stroke-based sketched symbol reconstruction and segmentation. IEEE Comput. Graph. Appl. **40**(1), 112–126 (2019)
21. Li, K. et al.: Universal sketch perceptual grouping. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 582–597 (2018)
22. Yang, L., et al.: Sketchgnn: semantic sketch segmentation with graph neural networks. ACM Trans. Graph. **40**(3), 1–13 (2021)
23. Hähnlein, F., Gryaditskaya, Y., Bousseau, A. Bitmap or vector? A study on sketch representations for deep stroke segmentation. Journées Francaises d'Informatique Graphique et de Réalité virtuelle (2019)
24. Lin, T.-Y. et al.: Microsoft coco: common objects in context. In: European Conference on Computer Vision (ECCV), pp. 740–755. Springer (2014)
25. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**(5–6), 602–610 (2005)
26. Welling, M., Kipf, T.N.: Semi-supervised classification with graph convolutional networks. In: Journal of International Conference on Learning Representations (ICLR) (2016)
27. Kirillov, A., He, K., Girshick, R., Dollár, P.: A unified architecture for instance and semantic segmentation (2017). http://presentations.cocodataset.org/COCO17-Stuff-FAIR.pdf (2017)
28. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
29. Ge, C., Sun, H., Song, Y.-Z., Ma, Z., Liao, J.: Exploring local detail perception for scene sketch semantic segmentation. IEEE Trans. Image Process. **31**, 1447–1461 (2022)
30. Long, J., Shelhamer, E. & Darrell, T.: Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440 (2015)

Zhengming Zhang received his B.S. degree from China University of Petroleum, Beijing in 2016. He is currently pursuing the Ph.D. degree with University of Chinese Academy of Sciences, Beijing, China. His current research interests include human–computer interaction and computer vision.

Xiaoming Deng received the bachelor's and master's degrees from Wuhan University, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences (CAS). He is currently a professor with the Institute of Software, CAS. He has been a research fellow with the National University of Singapore, and a postdoctoral fellow with the Institute of Computing Technology, CAS, respectively. His main research topics are in computer vision, and specifically related to 3D reconstruction, human motion tracking and synthesis, and natural user interfaces.

Jinyao Li received her M.S. degree in the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences in 2022. She received her Bachelor degree in the College of Information Science and Technology, Beijing Normal University in 2019. Her research interests include affective computing and human–computer interaction.

Yu-Kun Lai received the bachelor's and Ph.D. degrees in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a Professor of visual computing with the School of Computer Science and Informatics, Cardiff University. His research interests include computer graphics, geometry processing, image processing, and computer vision. He is on the editorial boards of Computer Graphics Forum and The Visual Computer.

Cuixia Ma received the B.S. and M.S. degrees from Shandong University, China, in 1997 and 2000, respectively, and the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 2003. She was a Research Associate with the Department of Computer Science, Naval Postgraduate School, Monterey, CA, USA, from 2005 to 2006. She is currently a Professor with the Institute of Software, Chinese Academy of Sciences. Her research interests include sketch interaction, multi-modal interaction, and cognitive computation.

Yong-Jin Liu received the B.Eng. degree from Tianjin University, Tianjin, China, in 1998, and the M.Phil. and Ph.D. degrees from The Hong Kong University of Science and Technology, Hong Kong, China, in 2000 and 2004, respectively. He is currently a Professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include computational geometry, computer vision, cognitive computation, and pattern analysis.

Hongan Wang received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 1999. He is currently a Professor with the Institute of Software, Chinese Academy of Sciences. He is the Director of the Intelligence Engineering Laboratory. His research interests include human–computer interaction, real-time intelligence, and real-time active database.