

# SceneSketcher: Fine-Grained Image Retrieval with Scene Sketches

Fang Liu<sup>1,2\*</sup>, Changqing Zou<sup>3\*</sup>, Xiaoming Deng<sup>1†</sup>, Ran Zuo<sup>1,2</sup>,  
Yu-Kun Lai<sup>4</sup>, Cuixia Ma<sup>1,2†</sup>, Yong-Jin Liu<sup>5,†</sup>, Hongan Wang<sup>1,2</sup>

<sup>1</sup>State Key Laboratory of Computer Science and Beijing Key Lab of  
Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences

<sup>2</sup> University of Chinese Academy of Sciences <sup>3</sup> HMI Lab, Huawei Technologies

<sup>4</sup> Cardiff University <sup>5</sup> Tsinghua University

**Abstract.** Sketch-based image retrieval (*SBIR*) has been a popular research topic in recent years. Existing works concentrate on mapping the visual information of sketches and images to a semantic space at the object level. In this paper, for the first time, we study the fine-grained *scene-level* SBIR problem which aims at retrieving *scene* images satisfying the user’s specific requirements via a freehand scene sketch. We propose a graph embedding based method to learn the similarity measurement between images and scene sketches, which models the multi-modal information, including the size and appearance of objects as well as their layout information, in an effective manner. To evaluate our approach, we collect a dataset based on SketchyCOCO and extend the dataset using Coco-stuff. Comprehensive experiments demonstrate the significant potential of the proposed approach on the application of fine-grained scene-level image retrieval.

**Keywords:** Sketch · Image retrieval · Graph convolutional network

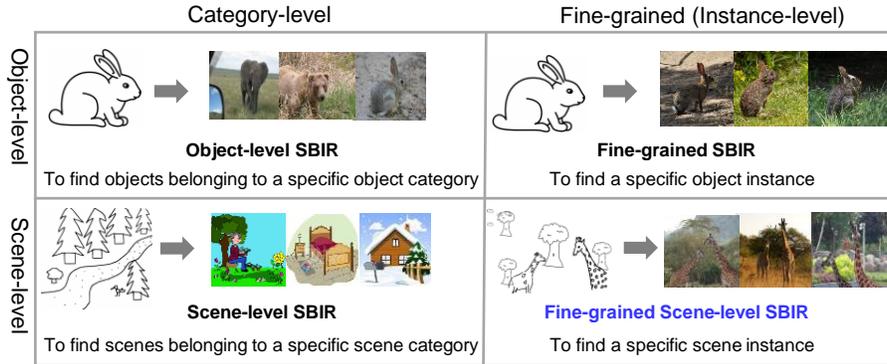
## 1 Introduction

Sketching is an effective way for humans to express target objects. Using sketches as a query to retrieve images [25] has drawn increasing interests in the last decade. Especially with the aid of touch devices, users can easily draw a sketch of the desired object, which facilitates the application of sketch-based image retrieval (SBIR). However, it is more desired to retrieve *scene-level* images using an input sketch in applications such as exploring a large number of landscape photos on a phone, or online interior style selection for bedroom design, etc.

Most current SBIR works are limited to retrieving images of the same category, while the shape, pose, and other fine-grained attributes of the retrieved images are often neglected [25]. Researchers presented various global and local descriptors to conduct the SBIR task, in which the key problem is to bridge the domain gap between sketches and images. Recently, the problem of fine-grained

---

\* indicates equal contributions. † indicates corresponding author.



**Fig. 1.** Illustration of the whole spectrum of SBIR problems. The proposed method, focusing on retrieving the scene-level images satisfying the user’s specific requirements via a freehand sketch, is in stark contrast to those of object-level SBIR methods [36, 25] and those focusing on retrieving scene-level images of the same scene class [7].

sketch-based image retrieval (FG-SBIR) was proposed in [36] (see the upper part of Fig. 1): it still performs the instance-level SBIR task, but allows users to not only query the target image with the same category, but also with the desired instance details. Although existing works conduct inspiring retrieval performance of images with a single object, to the best of our knowledge, sketch-based retrieval of fine-grained scene-level images consisting of multiple objects is still a new problem to explore.

In this paper, we address a new problem of fine-grained scene-level SBIR (see Fig. 1), which aims to conduct scene-level (i.e. with multiple objects and instances) sketch-based image retrieval, and enforces the layout of scenes and objects’ visual attributes such as relative sizes and poses. Compared to fine-grained scene-level SBIR, scene-level SBIR [32] overlooks the fine-grained details of scene layout and visual attributes, and only enforces the category of scenes, whereas (fine-grained) object-level SBIR [25, 36] only retrieves a single instance and overlooks the scene context of the object. Fine-grained scene-level SBIR can facilitate novel SBIR applications. For example, if a user wants to pick specific photos from the album on his mobile phone, he can first draw a scene sketch on the mobile phone to interpret the query intention, and then retrieve the desired photos. Although text can be an alternative to query scene-level images, it is hard to describe the layout and fine-grained details due to the inherent ambiguity of text. Moreover, users can continuously adjust the shape, size and position of the objects in the input scene sketch to obtain better retrieval results.

However, fine-grained scene-level SBIR is challenging. Firstly, the domain gap between sketches and images is large, and intra-variations between sketches in the same category or the same object can also be significant [33]. Secondly, it is unclear how to simultaneously represent the layout and visual details of instances

in the scene sketch and scene image, and design an embedding network to narrow down the domain difference between them.

In this paper, we propose a fine-grained scene-level SBIR method. To explicitly integrate the visual attributes and the layouts of target images, we present a sketch scene graph, which is composed of nodes that represent object entities in a sketch scene and edges that represent node distances and relationships. Then we use a Graph Convolutional Network (GCN) to embed the sketch scene graph into a feature space. To capture the size of instances, we design a category-wise IoU (Intersection over Union) score as the metric to evaluate the similarity between sketches and images. The proposed category-wise IoU can capture the size of object instances better than the IoU used in image segmentation, which is calculated by working out the IoU for each category and then taking the mean. Finally, we design a triplet network using the graph embedding of the layout and the size of object instances.

The main contributions of this work are as follows: 1) To the best of our knowledge, the problem of fine-grained scene-level sketch-based image retrieval is addressed for the first time, which can enable related SBIR research applications; 2) We propose to use a graph-based representation to explicitly model objects and layouts of sketch scenes, and design a category-wise IoU score to evaluate the similarity between sketches and images in terms of objects' relative sizes; 3) We integrate our sketch scene graph embedding and category-wise IoU score via a triplet training process. Experiments show that our method achieves state-of-the-art performance on our scene sketch database.

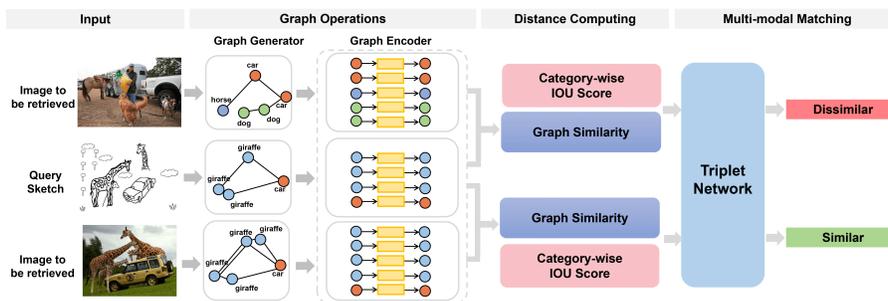
## 2 Related Work

**Sketch-based Image Retrieval (SBIR)** Sketch-based image retrieval has been extensively studied since 1990s [6], and has attracted more attention recently due to the booming of touch devices. The early SBIR works aim to retrieve images of the same category (category-level), usually using hand-crafted image descriptors (e.g. SIFT, HOG etc.), to conduct shape matching between sketches and edge maps of natural images [14, 19, 20, 3]. Eitz et al. [15] present a benchmark for evaluating the performance of large-scale SBIR systems, and utilize descriptors based on the bag-of-features approach for SBIR. Recently, several deep learning based SBIR methods [23, 29, 34, 4] have been proposed and refresh the performance of the major SBIR benchmarks. Sangkloy et al. [25] present the Sketchy database, the first large-scale dataset of sketch-photo pairs, and use it to train cross-domain neural networks which embed sketches and photos in a common feature space. Furthermore, a few methods of zero-shot sketch-based image retrieval (ZS-SBIR) have been proposed to handle the data deficiency of large-scale sketch-photo pairs. ZS-SBIR is an SBIR task that can conduct the retrieval task on unseen object classes, and it is often treated as a domain adaptation problem [38, 12]. Dey et al. [11] construct a dataset named QuickDraw-Extended to simulate ZS-SBIR of the real scenario, and exploit both visual and semantic information to conduct feature embedding.

**Fine-Grained Sketch-based Image Retrieval (FG-SBIR)** Compared to object-level SBIR, FG-SBIR requires that the retrieved images contain fine-grained details described in the input scene sketch. Yu et al. [36] introduce a database of sketch-photo pairs with fine-grained annotations, in which only one object exists in each sketch or image, and develop a deep triplet-ranking model for instance-level FG-SBIR. Song et al. [26] propose a fine-grained SBIR model that exploits semantic attributes and deep feature learning in a complementary way. Furthermore, a spatially aware model which combines coarse and fine semantic information is proposed by [27]. Pang et al. [24] identify cross-category generalization for FG-SBIR as a domain generalization problem and propose an unsupervised learning approach to modeling a universal manifold of prototypical visual sketch traits. Though FG-SBIR research works achieve inspiring progress, they focus on retrieving a single object, which may not fit well to SBIR applications in real scenarios. In this paper, we explore a new scene-level fine-grained SBIR, which utilizes local features such as object instances and their visual detail, and global context such as the scene layout.

**Scene Sketch** Existing scene sketch related work includes scene image synthesis via sketches [8], scene image retrieval (not fine-grained) [7], and semantic segmentation of scene sketches [40]. Chen et al. [8] composite a photo-realistic scene image with a hand-drawn sketch and text as input. The key idea is to retrieve initial candidates of object instances via the input text and sketch, and then blend the whole scene’s images. Compared to [8], our work aims to retrieve a specific image from an image gallery instead of compositing a synthesized image. Similarly, Dey et al. [10] present a multi-object image retrieval system using sketch and text as inputs at a coarse level. Castrejon et al. [7] propose a cross-modal scene representation for multi-modal data, and apply the class-agnostic representation in cross-modal retrieval. Xie et al. [32] introduce an ZS-SBIR framework based on this cross-modal scene dataset. The method mainly utilizes the overall visual features, while the layout and details in the scene are overlooked. Zou et al. [40] present a scalable scene sketch dataset with rich semantic and instance segmentation annotations, named SketchyScene, and conduct a preliminary study of scene-level SBIR using an object-level SBIR method [36].

**Image Retrieval with Graph Convolutional Networks** Graph convolutional networks (GCNs) [22] are an effective neural network architecture on graphs, which extract features from graphs or nodes in graphs. GCNs have been successfully applied in a variety of applications such as image matching [31], action recognition [17], and text matching [39]. Tripathi et al. [30] adopt scene graphs to model the layouts of images, and apply them to conduct image synthesis. Khan et al. [21] use GCNs to solve the multi-label scene classification problem of very high resolution satellite remote sensing images. Chen et al. [9] develop a multi-label image recognition method, which integrates the dependencies of object labels via a directed graph consisting of the object labels into the extracted image features. Compared to these approaches, our method adopts multi-modal information to construct the node embedding and edge weights, including the size and appearance of objects as well as their layout.



**Fig. 2.** Overview of our fine-grained scene-level SBIR framework. Our network mainly consists of three phases: graph operations, distance computing and multi-modal matching. We first construct graphs for sketches and images, and then utilize GCNs to encode them. Finally, we integrate the graph information with our proposed category-wise IoU score to conduct retrieval via a triplet training process.

## 3 Methods

### 3.1 Overview

In this section, we learn a feature embedding of scene sketches and images to enforce the distance in the feature space to be closely related to the similarities of the layout, appearance and semantic information between scene sketches and images. For fine-grained scene-level SBIR, it is a key issue to model the correlation context between the object instances and capture the fine-grained details of each object instance. In this work, we propose to use graph convolutional networks (GCNs) and category-wise IoU to conduct the feature embedding of scene sketches and images. Fig. 2 shows an overview of our method. Our method consists of graph encoders, a graph similarity function, category-wise IoU measures, and a triplet similarity network.

### 3.2 Scene Graph Generation

We formulate a scene as a weighted, undirected scene graph, which models the visual appearance, size, pose and other fine-grained details of the object instances in a scene sketch or a scene image explicitly. Our scene graph can be represented as  $G = (N, E)$ , where  $N = \{n_i\}$  is the node set and  $E = \{e_{i,j}\}$  is the edge set, where  $e_{i,j} = (n_i, n_j)$  is the edge connecting nodes  $n_i$  and  $n_j$ . The category of the node set is denoted as  $C = \{c_i\}$ , where  $c_i$  is the category of node  $n_i$ . In this work, the nodes  $N$  are defined as the object instances, and  $E$  are the edges that link each pair of the nodes. The scene graph generation consists of two steps: node construction and edge construction.

**Node Construction** The node set  $N$  and the category set  $C$  of a scene graph can be obtained by either human annotations, or any pretrained object detectors (e.g. [40]). Each node  $n_i$  contains visual features  $v_i$ , object category label  $c_i$  and

its spatial position  $p_i$ . In order to get the visual feature of each node, we first adopt a sketch classification task to fine-tune the Inception-V3 [28] pretrained on the ImageNet using the object-level data from the collected dataset illustrated in Fig. 4, and then use this model to extract the 2048-d feature of each object from its bounding box. Category label  $c_i$  of each node is encoded to a 300-d vector  $\tilde{c}_i$  by Word2Vec [1]. We denote the spatial information of the node by a 4-d vector  $p_i$  indicating the top left and bottom right coordinates of the node bounding box. For each node, we get a fusion node feature  $x_i$  by concatenating  $v_i$ ,  $\tilde{c}_i$  and  $p_i$ . This fusion node feature captures the appearance feature, the semantic feature of each object, as well as the spatial information.

**Edge Construction** The object nodes are connected with undirected weighted edges, and the edge weight between a pair of object nodes shows their correlation. Given two object nodes  $n_i$  and  $n_j$  of the graph, we define the edge weight  $A_{i,j} \in (0, 1)$  between them using a normalized Euclidean distance as follows:

$$A_{i,j} = \frac{D_{i,j}}{\sum_{e_{p,q} \in E, p < q} D_{p,q}} \quad (1)$$

where  $D_{i,j} = \|x_j - x_i\|_2$  is the Euclidean distance of the fusion features of the nodes  $n_i$  and  $n_j$ .

### 3.3 Graph Encoder

After we generate the scene graphs for sketches and images, we adopt GCNs to learn node-level representations for our scene graph by updating the node features by propagating information between nodes. A GCN learns a function  $f(\cdot, \cdot)$  to extract features on a graph  $G = (N, E)$ , which takes a feature matrix  $H^{l-1}$  and the corresponding adjacency matrix  $A = \{A_{ij}\}$  as inputs. The  $l$ -th layer of the GCN can be formulated as

$$H^{(0)} = \{x_i\}_{i=1}^n \quad (2)$$

$$H^{(l)} = f(H^{(l-1)}, A), l > 1 \quad (3)$$

Then we adopt the propagation rule introduced in [22], and the function  $f(\cdot, \cdot)$  can be written by

$$f(H^{(l)}, A) = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \quad (4)$$

where  $\sigma(\cdot)$  is the *leaky\_relu* activation function,  $\hat{A} = A + I$ , and  $\hat{D}$  is the diagonal node degree matrix of  $\hat{A}$ , and  $W^{(l)}$  is a weight matrix to be learned.

We denote the outputs of the last layer of graph convolution networks for sketches and images to be two encoded feature graphs  $\mathcal{G}_S$  and  $\mathcal{G}_I$ , respectively, with the encoded node features denoted as  $\{\hat{x}_S^i\}$  and  $\{\hat{x}_I^j\}$ .

### 3.4 Graph Similarity Function

After we get the encoded feature graphs  $\mathcal{G}_S$  and  $\mathcal{G}_I$  for a sketch and an image (refer to Sec. 3.3), we utilize a graph matching function to measure the similarity of the two graphs.

Denote  $N_S$  and  $N_I$  to be the node numbers in  $\mathcal{G}_S$  and  $\mathcal{G}_I$ , respectively. Firstly, we compute a score matrix  $\hat{S}$  of the size  $N_S \times N_I$  by computing the similarity between all node pairs in  $\mathcal{G}_S$  and  $\mathcal{G}_I$ , where cosine distance is used to calculate the similarity between the features of two nodes. Secondly, we select the maximum score of each row, i.e. for each encoded feature  $\hat{x}_S^i$  in  $\mathcal{G}_S$ , get the most similar node feature  $\hat{x}_I^j$  in  $\mathcal{G}_I$ . Finally, we compute the overall similarity of  $\mathcal{G}_S$  and  $\mathcal{G}_I$  by averaging the maximum scores of all rows as:

$$\phi_{GM}(\mathcal{G}_S, \mathcal{G}_I) = \frac{1}{N_S} \sum_{p=1}^{N_S} \max_{q \in [1, N_I]} \text{Cosine}(\hat{x}_S^p, \hat{x}_I^q). \quad (5)$$

### 3.5 Category-wise IoU

To evaluate the similarity of the layout and sizes of object instances between a sketch and an image, we design a category-wise IoU. Denote  $M_S^i$  and  $M_I^i$  to be the union sets of the object masks for an object category label  $c_i$  in a pair of sketch  $S$  and image  $I$ , respectively. Then we compute the intersection and the union of  $M_S^i$  and  $M_I^i$  by  $M_S^i \cap M_I^i$  and  $M_S^i \cup M_I^i$ .

Finally, we define the category-wise IoU score  $\phi_{IoU}$  between sketch  $S$  and image  $I$  as the division of the sum of the intersection masks of all object categories and the sum of the union masks of all object categories:

$$\phi_{IoU}(S, I) = \frac{\sum_{i=1}^{|C|} M_S^i \cap M_I^i}{\sum_{i=1}^{|C|} M_S^i \cup M_I^i} \quad (6)$$

where  $|C|$  the number of object categories.

The proposed category-wise IoU can capture the size of object instances better than the IoU used in image segmentation, which is calculated by working out the IoU for each category and then taking the mean (See Sec. 4.4).

### 3.6 Loss Function

**Triplet Loss** Inspired by [25, 36], we adopt the ranking triple loss that can express the fine-grain relationship better than Siamese loss [25]. The triple loss aims to enforce that the embedding features of two examples with the same label are close and the embedding features of two examples with different labels are far away.

The input of a triplet network is a triplet  $(S, I^+, I^-)$ , where  $S$  is a scene sketch,  $I^+$  is the corresponding image of  $S$ , and  $I^-$  is an image of a different scene. The triple loss  $L_{tri}$  of  $(S, I^+, I^-)$  can be computed by

$$L_{tri} = \max(d(S, I^+) - d(S, I^-) + m, 0) \quad (7)$$

where  $d(\cdot, \cdot)$  is the distance function in the embedding space, and  $m$  is a margin which is set to 0.4.

**Table 1.** Comparison of the existing sketch databases and our database.

Dataset	# of Sketches	Any Images?	Is Paired?	Annotation Type	Multi-objects?
TU-Berlin [13]	20,000	No	-	Class label	No
QuickDraw [18]	over 50 million	No	-	Class label	No
The Sketchy Database [25]	75,471	Yes	Yes	Class label	No
Shoes [36]	419	Yes	Yes	Class label	No
Chairs [36]	297	Yes	Yes	Class label	No
SketchyScene [40]	7265	Yes	Yes	Segmentation	Yes
CMPlaces [7]	8694	Yes	No	Class label	Yes
Our scene sketch database	<b>1225</b>	<b>Yes</b>	<b>Yes</b>	<b>Segmentation</b>	<b>Yes</b>

With three scene graphs  $G_S$ ,  $G_{I^+}$  and  $G_{I^-}$  of the triplet  $(S, I^+, I^-)$ , we define  $d(S, I^+)$  and  $d(S, I^-)$  of Eq. (7) by integrating the graph similarity score  $\phi_{GM}$  in Eq.(5) and the category-wise IoU  $\phi_{IoU}$  in Eq.(6) by

$$d(S, I) = 1 - \lambda_1 \phi_{GM}(S, I) - \lambda_2 \phi_{IoU}(S, I) \quad (8)$$

where  $I$  is an image (which can be  $I^+$  or  $I^-$ ),  $\lambda_1$  and  $\lambda_2$  are the weights of  $\phi_{GM}$  and  $\phi_{IoU}$ . In our experiments, we set  $\lambda_1 = 1$  and  $\lambda_2 = 0.8$ .

## 4 Experiments

### 4.1 Datasets

Although several sketch datasets [13, 18, 25, 36, 40, 7] are publicly available (shown in Table 1 and Fig. 3), none of them are suitable to evaluate our method. TU-Berlin [13], QuickDraw [18] and The Sketchy Database [25] are all datasets of sketches of single object instances. Sketch me that shoe [36] is the first dataset of fine-grained sketch-photo pairs, and facilitates the fine-grained sketch-related applications. However, each sketch-photo pair in this dataset also contains only one object instance, where all the sketches and images have clean backgrounds. Moreover, there are only a few hundred images in the database, which is insufficient for large-scale SBIR. SketchyScene [40] and CMPlaces [7] are the two available scene-level sketch datasets. SketchyScene cannot be used to train and evaluate our fine-grained scene-level SBIR network, which requires the visual features of object instances, because it does not contain the bounding box or object instance segmentation annotations. The images of SketchyScene are all cartoon clips, while we intend to retrieve natural photos. CMPlaces, in which only scene category labels are available, cannot be used for our problem either. On the one hand, it does not contain paired image and sketch data. On the other hand, it does not contain object instance segmentation annotations such as SketchyScene.

**Our Sketch Database.** Existing benchmarks for SBIR do not fit our problem, they either just contain a single object in one photo, or no fine-grain annotations of objects are available. Thus, we collect a scene sketch-image database (referred to as our sketch database) based on SketchyCOCO [16], and utilize the Scene

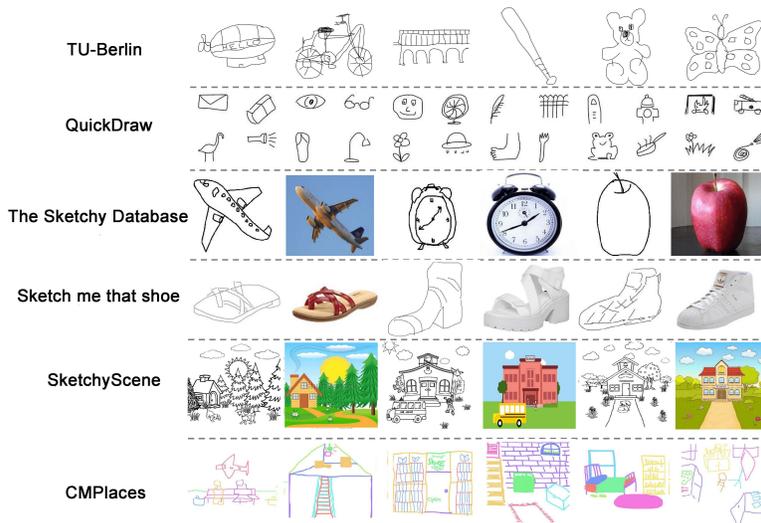


Fig. 3. Examples of the existing sketch databases.

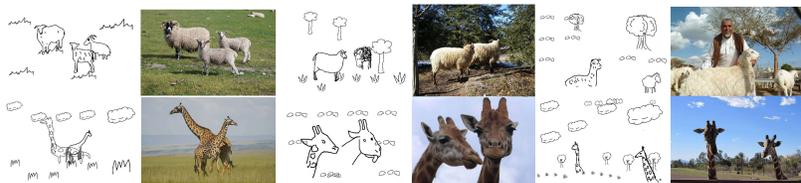


Fig. 4. Examples of our fine-grained scene-level sketch dataset.

SBIR database for our fine-grained scene-level SBIR task. SketchyCOCO contains over 14,000 scene-level sketch-photo pairwise examples, but most of them only contain one foreground instance. We pick up 1,225 scene sketch-photo pairs containing more than one object instance from SketchyCOCO, covering 14 object categories (bicycle, car, motorcycle, airplane, traffic light, fire hydrant, cat, dog, horse, sheep, cow, elephant, zebra, giraffe). Fig. 4 shows several examples of our database. In each row, we display three samples with the same object categories, and fine-grained SBIR models are needed to differentiate a specific scene.

## 4.2 Evaluation Metrics

We split our scene sketch dataset into training and testing sets, containing 1015 and 210 sketch-image pairs, respectively. We adopt a standard evaluation metric for retrieval as [36], recall at rank  $K$  (Recall@K), which is computed with the percentage of test queries where the target image is within the top  $K$  retrieved images.



**Fig. 5.** Top-10 fine-grained scene-level SBIR results with our method. The true matches are highlighted with red rectangles.

### 4.3 Comparison with Baselines

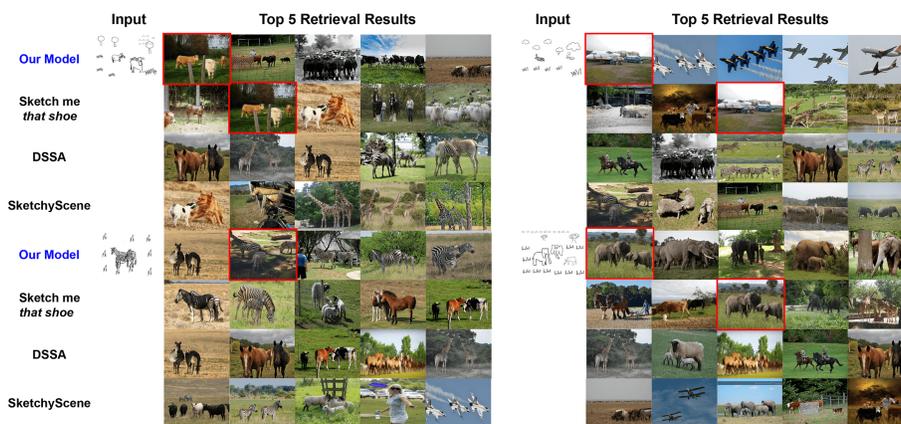
Fig. 5 shows several fine-grained SBIR examples with our method. For each query sketch, there are typically a handful of visually very similar photos; the lower-rank accuracy, especially at top-1, thus is a better indication on how well the model is capable of distinguishing fine-grained subtle differences between candidate photos. When collecting the dataset, some sketches do not match the photos exactly, thus there are cases that no images in the database can fully match the input sketch. Moreover, we investigate the performance of our fine-grained scene-level SBIR by making images extremely similar in overall layout of sketches, category of objects, and their position and shape, which makes the task more challenging (see supplementary material).

We compare our model with several state-of-the-art (SOTA) object-level SBIR and fine-grained SBIR approaches.

**HOG-BoW+RankSVM** [20] and **Dense HOG+ RankSVM** [36] are two methods using hand-crafted features. HOG-BoW descriptor is a popular visual feature in SBIR [19, 20]. We extract HOG features for each image, and feed them to the BoW (Bag-of-Words) framework for feature encoding. Then, we train a RankSVM model to rank the results as [35]. In the comparison, the used triplet annotations are the same as those we use in our experiment. We also compare with the model [36] by extracting Dense HOG features via concatenating HOG features over a dense grid. We follow the same setting in [36] to extract Dense HOG features.

**Table 2.** Comparison of scene-level SBIR performance with existing SBIR methods on our database (210 testing images) and our extended database (5210 testing images).

	Our Sketch Database			Extended Database		
	Recall@1	Recall@5	Recall@10	Recall@10	Recall@50	Recall@100
HOG+BoW+RankSVM [20]	0.48	1.43	4.76	0.48	0.48	0.48
Dense HOG+RankSVM [36]	0.48	3.81	5.71	0	0.95	1.91
Sketch-a-Net+RankSVM [37]	0.48	3.33	4.76	0	0.95	2.86
Sketch me that shoe [36]	6.19	17.15	32.86	1.90	6.19	8.57
DSSA [27]	0.48	3.81	7.62	0	0.95	1.90
SketchyScene [40]	1.43	4.76	8.57	0.48	0.95	2.86
<b>Our model</b>	<b>31.91</b>	<b>66.67</b>	<b>86.19</b>	<b>38.10</b>	<b>68.10</b>	<b>82.86</b>



**Fig. 6.** Comparison of scene-level SBIR results with our method and three state-of-the-art SBIR methods: Sketch me that shoe [36], DSSA [27], SketchyScene [40]. The ground truth matches are highlighted with red rectangles.

**Sketch-a-Net+RankSVM** [37], **Sketch me that shoe** [36], **DSSA** [27], and **SketchyScene** [40] adopt deep features for SBIR. In Sketch-a-Net+RankSVM [37], the deep features are ranked with RankSVM. And Sketch me that shoe [36] presents a deep triplet ranking model for fine-grained SBIR, where free-hand sketches are used as queries for instance-level retrieval of images. We have achieved a better result with our data by loading the pre-trained model and fine-tuning it. DSSA [27] utilizes a deep spatial-semantic attention mechanism for fine-grained SBIR, and it models the fine-grained details and their spatial context instead of only adopting a coarse holistic matching strategy. We also compare our method with SketchyScene, which conducts a scene-level SBIR based on the triplet ranking network similar to the network in [36], using the overall deep features of the scene sketch as input.

Table 2 shows the comparison of the retrieval recalls with our model and the compared methods. Fig. 6 compares the qualitative results with our method and most related SOTAs: Sketch me that shoe [36], DSSA [27], and SketchyScene [40].

The results indicate that our model achieves significantly higher recall than the other baselines. Conventional SBIR methods with hand-crafted features designed for SBIR with a single object get poor performance on our scene sketch dataset. Sketch me that shoe [36] is a more related SOTA SBIR model, which is also the first work on fine-grained SBIR task. However, the Recall@1, Recall@5 and Recall@10 with our method are about 25%, 50% and 54% higher than those with Sketch me that shoe. Therefore, our method achieves the best retrieval performance on our database, which demonstrates that our method is effective.

#### 4.4 Ablation Study

Our fine-grained scene-level SBIR method adopts scene sketch graphs to explicitly model the layout and local details of each object in sketches and images, and uses a category-wise IoU score between a pair of sketch and image to enforce the size and position of object instances in sketch and image. In order to demonstrate the contribution of each component, we compare our full model with the following eight stripped-down models:

1. **Visual features as graph only.** When constructing the nodes in the sketch scene graph, we only use visual features extracted by Inception-V3 as the node feature. We use the same category-wise IoU loss as our full model.
2. **Category labels as graph only.** We only use the category label of each object as the node feature, and visual features and spatial positions are not included. The same category-wise IoU loss as our full model is used.
3. **Visual features and category labels as graph.** To show the effect of spatial information, we use the visual features and category label of each object as the node feature, while no spatial positions of object bounding boxes are included. The category-wise IoU loss as our full model is used.
4. **Graph triplet loss only.** We use the same graph generator and graph encoder as our full model, but category-wise IoU loss is not used.
5. **Category-wise IoU loss only.** We use our category-wise IoU to rank the pairs of sketch and image. The scene sketch graph is not used.
6.  **$IoU_{category}$  only.** We replace the category-wise IoU loss in model (5) by  $IoU_{category}$  [2], which is the major evaluation metric in semantic segmentation.  $IoU_{category}$  is computed as  $\phi_{IoU_{category}}(S, I) = \sum_{i=1}^{|C|} \frac{M_S^i \cap M_I^i}{M_S^i \cup M_I^i}$ . The scene sketch graph is not used.
7. **Global IoU only.** We replace the category-wise IoU loss in model (5) by global IoU [2]. Denote  $M_S$  and  $M_I$  to be the union sets of all the object masks in sketch and image, respectively. Unlike our category-wise IoU, global IoU ignores category information and is computed as  $\phi_{Global\_IoU}(S, I) = \frac{M_S \cap M_I}{M_S \cup M_I}$ . The scene sketch graph is not used in this model.
8.  **$IoU_{category} +$  Graph feature.** We replace our category-wise IoU with  $IoU_{category}$  [2], and combine it with the scene sketch graph via triplet training. The graph feature of our full model is used in this setting.

The performances of our full models and the above eight models on the fine-grained scene-level SBIR are shown in Table 3. 1) Compared to the recalls with scene sketch graphs using different node features (1, 2, 3, 9 in Table 3), we observe that the Recall@1, Recall@5 and Recall@10 using the node features with visual information, category labels and spatial information (our full model) are about 8%, 15% and 9% higher than those using only visual features, about 2%, 4% and 4% higher than those using only category labels, and about 1.5%, 2% and 2% higher than those using visual features and category labels. Thus, visual features, category labels and position information all contribute to enhancing the retrieval performance. 2) By comparing the ranking results using our category-wise IoU only,  $IoU_{category}$  only and global IoU, we observe that the recall with our category-wise IoU is better (5, 6, 7 in Table 3). When integrated with graph features (8, 9 in Table 3), category-wise IoU is still superior to  $IoU_{category}$ . 3) Compared to the recalls with graph features only, our full model gains about 18%, 36% and 39% on Recall@1, Recall@5 and Recall@10 (4, 9 in Table 3). Thus, category-wise IoU is important for our network.

**Table 3.** Effect of each components on the fine-grained scene-level SBIR on our scene sketch database (210 testing images) and our extended database (5210 testing images).

Model settings	Our Sketch Database			Extended Database		
	Recall@1	Recall@5	Recall@10	Recall@1	Recall@5	Recall@10
1. Visual feature as graph only	24.29	51.90	77.14	8.09	18.09	25.23
2. Category label as graph only	29.52	62.86	82.38	8.57	20.95	30.00
3. Visual feature and Category label as graph	30.48	64.76	83.81	11.43	23.33	30.95
4. Graph Triplet loss only	13.33	30.00	47.62	2.38	6.67	10.00
5. Category-wise IoU only	28.10	61.90	80.0	6.67	19.05	24.29
6. $IoU_{category}$ only	23.82	59.05	76.19	4.76	16.19	23.81
7. Global IoU only	5.24	19.05	28.10	0	0.48	2.38
8. $IoU_{category}$ +Graph feature	24.76	59.05	78.57	4.76	16.67	23.81
9. Our model	<b>31.91</b>	<b>66.67</b>	<b>86.19</b>	<b>12.38</b>	<b>26.67</b>	<b>38.10</b>

#### 4.5 Results on Our Extended Scene Sketch Database

In order to investigate the performance of our method using a larger image gallery, we extend our scene sketch database with natural images from Cocosuff [5], named our extended scene sketch database. We select 21,379 natural images, the objects of which are within the 14 categories in our scene sketch database. These natural images do not have corresponding sketches in our scene sketch database. Then, we split these natural images into a test dataset with 5,000 images and a training dataset with 16,379 images, and combine them with the images of the training and test dataset in our scene sketch database.

We compare the scene-level SBIR performance with the existing SOTA methods on our extended database (shown on the right three columns in Table 2). Since Recall@5 of the compared methods is close to zero, we show the Recall@10, Recall@50 and Recall@100 instead. Our method achieves significantly better

performance than the compared methods. Existing object-level SBIR methods perform worse due to the fact that these methods directly compare the visual features of sketches and images, but neglect the key scene contexts such as object layout. Fig. 7 shows the qualitative results of scene-level SBIR on our extended database. Our model can capture details well and retrieve fine-grained images.

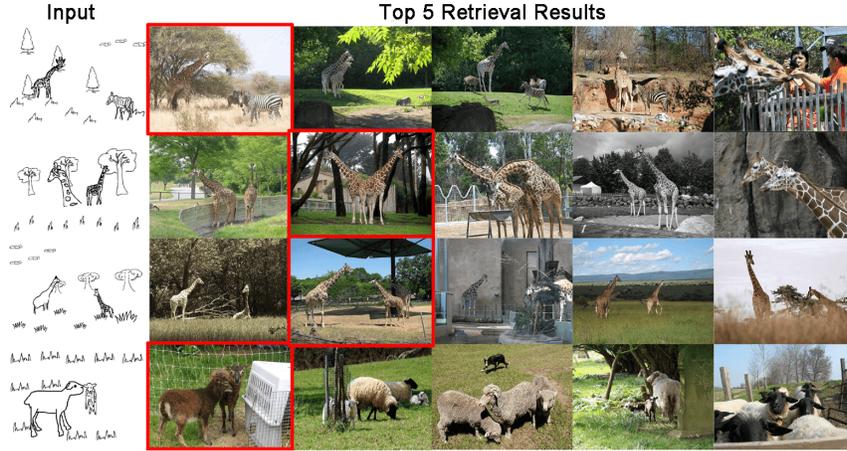


Fig. 7. Examples of scene sketch SBIR results on our extended database.

We also conduct an ablation study of each component of the proposed method on our extended database (shown in Table 3). The results again demonstrate that the scene graph with nodes using visual features, category label and spatial information, and category-wise IoU are both effective.

## 5 Conclusion

In this work, for the first time, we have addressed and explored the new problem of scene-level fine-grained sketch-based image retrieval. A graph-based framework has been proposed to explicitly model the layout and fine-grained details of sketch scenes at the same time. A category-wise IoU was designed to enhance the SBIR performance in a simple and effective manner. Experiments show that our method is superior to the existing sketch-based image retrieval methods. In the future, we would fuse semantic analysis and scene understanding to promote the method to work on larger datasets.

**Acknowledgements** This work was supported by the National Key Research and Development Plan (2016YFB1001200), Natural Science Foundation of China (61872346, 61725204, 61473276), Natural Science Foundation of Beijing (L182052), and Royal Society-Newton Advanced Fellowship (NA150431).

## References

1. <https://code.google.com/archive/p/word2vec/>
2. <https://www.cityscapes-dataset.com/benchmarks/>
3. Belongie, S., Malik, J., Puzicha, J.: Shape context: A new descriptor for shape matching and object recognition. In: *Advances in neural information processing systems*. pp. 831–837 (2001)
4. Bui, T., Ribeiro, L., Ponti, M., Collomosse, J.: Sketching out the details: Sketch-based image retrieval using convolutional neural networks with multi-stage regression. *Computers & Graphics* **71**, 77–87 (2018)
5. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1209–1218 (2018)
6. Cao, Y., Wang, C., Zhang, L., Zhang, L.: Edgel index for large-scale sketch-based image search. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 761–768 (2011)
7. Castrejon, L., Aytar, Y., Vondrick, C., Pirsiavash, H., Torralba, A.: Learning aligned cross-modal representations from weakly aligned data. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2940–2949 (2016)
8. Chen, T., Cheng, M.M., Tan, P., Shamir, A., Hu, S.M.: Sketch2photo: Internet image montage. In: *ACM transactions on graphics (TOG)*. vol. 28, p. 124 (2009)
9. Chen, Z.M., Wei, X.S., Wang, P., Guo, Y.: Multi-label image recognition with graph convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 5177–5186 (2019)
10. Dey, S., Dutta, A., Ghosh, S.K., Valveny, E., Lladós, J., Pal, U.: Learning cross-modal deep embeddings for multi-object image retrieval using text and sketch. In: *24th International Conference on Pattern Recognition*. pp. 916–921 (2018)
11. Dey, S., Riba, P., Dutta, A., Lladós, J., Song, Y.Z.: Doodle to search: Practical zero-shot sketch-based image retrieval. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2179–2188 (2019)
12. Dutta, A., Akata, Z.: Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 5089–5098 (2019)
13. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? *ACM Transactions on graphics (TOG)* **31**(4), 1–10 (2012)
14. Eitz, M., Hildebrand, K., Boubekur, T., Alexa, M.: An evaluation of descriptors for large-scale image retrieval from sketched feature lines. *Computers & Graphics* **34**(5), 482–498 (2010)
15. Eitz, M., Hildebrand, K., Boubekur, T., Alexa, M.: Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE transactions on visualization and computer graphics* **17**(11), 1624–1636 (2010)
16. Gao, C., Liu, Q., Xu, Q., Wang, L., Liu, J., Zou, C.: SketchyCOCO: Image generation from freehand scene sketches. In: *Proceedings of the European Conference on Computer Vision*. pp. 5174–5183 (2020)
17. Guo, M., Chou, E., Huang, D.A., Song, S., Yeung, S., Fei-Fei, L.: Neural graph matching networks for fewshot 3D action recognition. In: *Proceedings of the European Conference on Computer Vision*. pp. 653–669 (2018)
18. Ha, D., Eck, D.: A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477* (2017)

19. Hu, R., Barnard, M., Collomosse, J.: Gradient field descriptor for sketch based retrieval and localization. In: IEEE International Conference on Image Processing. pp. 1025–1028 (2010)
20. Hu, R., Collomosse, J.: A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *Computer Vision and Image Understanding* **117**(7), 790–806 (2013)
21. Khan, N., Chaudhuri, U., Banerjee, B., Chaudhuri, S.: Graph convolutional network for multi-label vhr remote sensing scene recognition. *Neurocomputing* **357**, 36–46 (2019)
22. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
23. Liu, L., Shen, F., Shen, Y., Liu, X., Shao, L.: Deep sketch hashing: Fast free-hand sketch-based image retrieval. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2862–2871 (2017)
24. Pang, K., Li, K., Yang, Y., Zhang, H., Hospedales, T.M., Xiang, T., Song, Y.Z.: Generalising fine-grained sketch-based image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 677–686 (2019)
25. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)* **35**(4), 1–12 (2016)
26. Song, J., Song, Y.Z., Xiang, T., Hospedales, T.M., Ruan, X.: Deep multi-task attribute-driven ranking for fine-grained sketch-based image retrieval. In: BMVC. vol. 1, p. 3 (2016)
27. Song, J., Yu, Q., Song, Y.Z., Xiang, T., Hospedales, T.M.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5551–5560 (2017)
28. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2818–2826 (2016)
29. Toliás, G., Chum, O.: Asymmetric feature maps with application to sketch based retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2377–2385 (2017)
30. Tripathi, S., Nittur Sridhar, S., Sundaresan, S., Tang, H.: Compact scene graphs for layout composition and patch retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 676–683 (2019)
31. Wang, R., Yan, J., Yang, X.: Learning combinatorial embedding networks for deep graph matching. arXiv preprint arXiv:1904.00597 (2019)
32. Xie, Y., Xu, P., Ma, Z.: Deep zero-shot learning for scene sketch. arXiv preprint arXiv:1905.04510 (2019)
33. Xu, P.: Deep learning for free-hand sketch: A survey. arXiv preprint arXiv:2001.02600 (2020)
34. Xu, P., Huang, Y., Yuan, T., Pang, K., Song, Y.Z., Xiang, T., Hospedales, T.M., Ma, Z., Guo, J.: Sketchmate: Deep hashing for million-scale human sketch retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8090–8098 (2018)
35. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 192–199 (2014)
36. Yu, Q., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 799–807 (2016)

37. Yu, Q., Yang, Y., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M.: Sketch-a-net: A deep neural network that beats humans. *International journal of computer vision* **122**(3), 411–425 (2017)
38. Zhang, J., Shen, F., Liu, L., Zhu, F., Yu, M., Shao, L., Tao Shen, H., Van Gool, L.: Generative domain-migration hashing for sketch-to-image retrieval. In: *Proceedings of the European Conference on Computer Vision*. pp. 297–314 (2018)
39. Zhang, T., Liu, B., Niu, D., Lai, K., Xu, Y.: Multiresolution graph attention networks for relevance matching. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. pp. 933–942 (2018)
40. Zou, C., Yu, Q., Du, R., Mo, H., Song, Y.Z., Xiang, T., Gao, C., Chen, B., Zhang, H.: SketchyScene: Richly-annotated scene sketches. In: *Proceedings of the European Conference on Computer Vision*. pp. 421–436 (2018)