Sketch-based Image Retrieval With Multi-clustering Re-ranking

Luo Wang, Xueming Qian*, *Member, IEEE*, Xingjun Zhang, Xingsong Hou

***Abstract*—To improve the performance of sketch-based image retrieval (SBIR) methods, most existing SBIR methods develop brand new SBIR methods. In fact, a re-ranking approach, which can refine the retrieval results of SBIR methods, is also beneficial. Inspired by this, in this paper, an SBIR re-ranking approach based on multi-clustering is proposed. In order to make the re-ranking approach invisible to users and adaptive to different types of image datasets, we made it an unsupervised method using blind feedback. Distinguished from the existing methods, this re-ranking approach uses the semantic information of three types of images: edge maps, object images (images with black background and natural images' foreground objects) and natural images themselves. With the initial retrieval results of an SBIR method, our approach first does the clustering operation for three types of images. Then, we utilize the clustering results to generate a cluster score for each initial retrieval result. Finally, the cluster score is used to calculate the final retrieval scores for the initial retrieval results. The experiments on different SBIR datasets are conducted. Experimental results demonstrate that, by implementing our re-ranking approach, the retrieval accuracy of a variety of SBIR methods is increased. Furthermore, the comparisons between our re-ranking method and the existing re-ranking methods are given.**

***Index Terms*—Sketch-based Image Retrieval, Re-ranking, multi-clustering**

I. Introduction

In order to search the desired images for internet users, text-based image retrieval (TBIR) [1]-[3], [55] [56]-[58] has been widely applied, and content-based image/video retrieval [4]-[9], [54], [59] also emerge. Apart from these two techniques, sketch-based image retrieval (SBIR) has received wide attention. SBIR systems merely need a user to draw a query sketch with simple lines and shapes on the white background. Then, the natural images that are relevant to the query sketch are returned to the user. Sometimes, people desire to search images of a particular object, while he does not know exactly the name of it and does not have an exemplar image at hand. At this very time, SBIR becomes helpful.

The majority of existing SBIR methods focus on developing novel SBIR systems [6], [12], [13]. Nevertheless, to devise an effective SBIR re-ranking algorithm, which is able to rearrange the initial retrieval results of SBIR systems, is also a good choice to improve the performance of SBIR systems [10], [11]. By adding a few steps added at the back of SBIR systems, an SBIR re-ranking algorithm can often boost the retrieval accuracy.

There are several challenges during developing SBIR re-ranking algorithms. First, since the internet users tend not to be disturbed by giving feedback to the SBIR system, the re-ranking algorithms that are invisible to users are welcomed. Second, the re-ranking algorithms should be able to deal with various SBIR systems. Third, the devised approach needs to be effective no matter what the dataset is.

In this paper, in order to face these three challenges, using the principle of blind feedback, we develop an unsupervised SBIR re-ranking method based on multi-clustering. First, the proposed re-ranking method is a re-ranking method using blind feedback, which can fulfill the re-ranking task without noticing users to give feedback. Then, our re-ranking approach can re-rank the initial results of different types of SBIR systems. Finally, experiments show that our re-ranking is effective for various initial SBIR systems on different datasets.

The framework of our proposed SBIR re-ranking method is shown in Figure 1. Given the initial retrieval results of an SBIR system as the inputs, the proposed method outputs the re-ranked retrieval results automatically. The steps for implementing our method are as the following.

1) *Initial SBIR*: With a query sketch and an Imageset of natural images, an initial SBIR system is used to get the initial retrieval results. By the way, the query sketch is merely used in the initial SBIR and does not participate in the following re-ranking process.
2) *Image Expansion*: Generate the edge maps as well as object images (the image with a natural image's foreground object and black background) of initial retrieval results, which makes each initial retrieval result has an edge map, an object image and a natural image.

This work was supported in part by the NSFC under Grant 61772407, and 61732008).

Xueming Qian (corresponding author, qianxm@mail.xjtu.edu.cn) is with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security, School of Information and Communication Engineering, and SMILES LAB, Xi'an Jiaotong University, Xi'an 710049, China.

Luo Wang (E-mail: wangluo@stu.xjtu.edu.cn) is with the Faculty of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China.

Xingjun Zhang (E-mail: xjzhang@mail.xjtu.edu.cn) is with the Department of Computer Science & Engineering of Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China.

Xingsong Hou (E-mail: houxs@mail.xjtu.edu.cn) is with the Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2959875, IEEE Transactions on Circuits and Systems for Video Technology

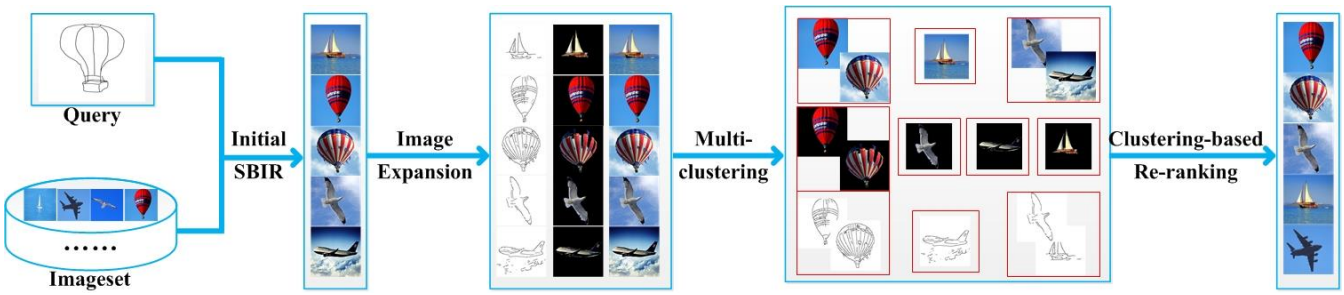


Fig.1 The framework of the proposed SBIR re-ranking method. Query refers to a query sketch, and Imageset contains all the natural images that were needed in the initial SBIR. The query sketch and Imageset are put into an SBIR system to do the initial SBIR and generate the initial retrieval results. Image Expansion transfers the initial retrieval results (natural images) into 3 types of images: edge maps, object images and natural images. Multi-clustering does clustering for each of these three types, where one rectangle corresponds to one cluster. Finally, the clustering results experience the clustering-based re-ranking, thus getting the final re-ranking results.

3) *Multi-clustering*: Extract the features of edge maps, object images and natural images, respectively. Then, three clustering operations are implemented, one for each type of features.
4) *Clustering-based Re-ranking*: The clustering results are then used for re-ranking initial retrieval results. The re-ranked results are the final retrieval results.

There are three main contributions in this paper.

1) We propose an effective unsupervised re-ranking system that is capable of improve the retrieval accuracy of SBIR systems. Our proposed method generates re-ranking results from some top initial retrieval results, and it does not involve the steps of initial SBIR systems and the information of the datasets. So, it can adjust to different initial SBIR systems and retrieval datasets.
2) The proposed re-ranking method leverages the semantic information of edge maps, object images and original natural images. Since the edge maps and object images are two effective forms of representing images, the semantic information of these two types of images benefits the re-ranking performance.
3) The proposed re-ranking method is entirely unsupervised, where no training operations are needed during clustering. Besides, our re-ranking method is based on blind feedback. So, users and developers do not need to do human-computer interaction during re-ranking, which benefits the user experience.

The remainder of this paper is organized as follows. This paper's related works are reviewed in Section II. In Section III, the detailed process of our proposed SBIR re-ranking method is described. We display and analyze the experimental results in Section IV. Discussions are presented in Section V. Finally, a short conclusion is given in Section VI.

II. Related work

In recent years, plenty of SBIR methods have been developed. Most of them focus on developing novel SBIR methods, and a few of them focus on SBIR re-ranking methods. In this paper, the proposed SBIR re-ranking method is implemented on some state-of-the-art SBIR methods. Furthermore, the comparison between the proposed re-ranking method and other re-ranking methods is given. This section briefly describes the existing SBIR methods and SBIR re-ranking methods.

A. SBIR methods

Sketches and natural images are vastly visually different from each other, thus resulting in a large image domain gap between natural images and sketches. In order to make SBIR work, the first task of SBIR is to overcome this image domain gap. In general, there are two strategies to bridge this image domain gap. The first strategy is to extract edge maps from natural images. Like sketches, edge maps are also composed of black lines and white ground, which makes the edge maps and sketches comparable [6], [12]-[26]. The second strategy is to build a common learning framework for both sketches and natural images. Through a machine learning process, the features of the natural images and those of sketches are comparable [27]-[32].

Early SBIR methods frequently utilize the first strategy to overcome the image domain gap. In order to generate edge maps of natural images, edge extraction approaches [33]-[35] are harnessed. Then, a common feature is designed for both sketches and edge maps. The common features between sketches and natural images tend to be geometric features, e.g., Histograms of Oriented Gradients (HoG) [15] and Edgel [12]. Besides, features based on deep learning are used [13], [20], [26], too.

Since edge maps have much less semantic information than natural images themselves, the retrieval performance of the above SBIR methods lags behind that of the SBIR methods using the second strategy to cope with the image domain gap. The SBIR methods using the second strategy allows the natural images themselves to be the inputs of feature extraction [27]-[32], [53], thus providing more semantic information. This makes the SBIR methods based on the second strategy often have a better retrieval performance. In these years, with the rapid development of the deep learning technique, the retrieval effects of SBIR improve a lot. Although most SBIR systems focus on retrieving images whose category is the same as the category of images for training machine learning models [27]-[32], there are also SBIR systems that can cope with the zero-shot SBIR tasks [53].

Different from these methods trying to develop a novel SBIR method, our proposed method aims to design an unsupervised

SBIR re-ranking method to re-rank the initial retrieval results obtained by an SBIR method. By this means, we enhance the retrieval performance of it. Since most of SBIR methods provide an initial similarity between the query sketch and each natural image, we can simply add our re-ranking method at the back of these SBIR methods. Since users are likely not to be disturbed by giving feedback during searching images on the internet, our proposed method is designed to be based on blind feedback.

B. SBIR re-ranking methods

Apart from developing novel SBIR methods, an effective SBIR re-ranking algorithm is also a good choice to improve the performance of SBIR methods.

Some of these methods are re-ranking methods based on explicit feedback [36], [37]. For these methods, the re-ranking method is started or optimized through some human-computer interaction activities, and then the re-ranking methods are implemented with the aid of these activities. Matsui et al. [37] put forward a re-ranking method to re-rank the SBIR task on Japanese manga datasets. After the initial retrieval results are obtained by their retrieval steps, the retrieval system shows the initial retrieval results to users. The users then choose an image from these results (changes can also be made on this stage) to be the basis of the following re-ranking process. In addition, Portenier et al. [36] proposed an SBIR re-ranking method. First, they extract CNN features for each initial retrieval result after an initial SBIR system finishes its retrieval operation. On receiving the initial retrieval results of an SBIR system, it then uses the k-means clustering algorithm to do clustering for these initial results. To achieve the best clustering, the k-means algorithm is initiated for some times, and they choose the best-performing one to be continued until it is convergence. Next, the average score of the initial retrieval results within each cluster is calculated, and the calculated scores are sorted to give ranks to these clusters. Finally, the initial retrieval results are re-ranked by these ranks.

There are also SBIR re-ranking methods that use the blind feedback, where the users are unaware of the entire re-ranking process [38]. In our previous work, we propose a supervised SBIR re-ranking method by CNN semantic re-ranking [38]. To begin with, two CNNs are trained to be the classifiers of sketches and natural images, respectively. The two CNNs are used to extract the classification information for each initial retrieval result, after which the classification information is utilized to calculate the category similarity between the query sketch and each initial retrieval result. Finally, the feature distance of each initial retrieval result, which is obtained through the initial SBIR, and the category similarity together are used to rearrange the initial retrieval results.

Unlike the above supervised SBIR re-ranking method based on CNN classification, one of our previous work is an unsupervised SBIR re-ranking method through re-ranking via visual feature verification (RVFV) and contour-based relevance feedback (CBRF) [6]. The whole re-ranking operation consists of three rounds. The first round of re-ranking is RVFV, which aims to reduce the images that are irrelevant to the query sketch among the high-ranked initial retrieval results. The second round of re-ranking is CBRF. CBRF picks out several highest-ranked initial retrieval results, then extract the edge maps of these images, and then use the edge maps as the new queries to start a new retrieval process using the initial SBIR system. The last round of re-ranking is to repeat the RVFV again.

Sometimes, re-ranking methods for other types of image retrieval can also be useful. [60] proposes a relevance-based ranking algorithm for tag-based image retrieval. Given a query tag and a collection of images, each image has a semantic score representing the similarity degree between the query tag and the tags of the images. There is also a visual similarity matrix denoting the similarity degree between visual features of images. With semantic scores and the visual similarity matrix, a closed-form solution is put forward to get the relevance scores to replace the semantic scores. Afterwards, [57] fine-tunes the ranking algorithm in [60] a bit, and then uses the fine-tuned version to do re-ranking for the initial retrieval results of their image retrieval task. This implies that we can use similar fine-tunes to make SBIR re-ranking methods.

We can see that most of the existing SBIR re-ranking methods need the users or developers to either implement some human-interaction during re-ranking or train some models through a supervised machine learning technique. So, in this paper, we desire to introduce an effective unsupervised SBIR re-ranking method using blind feedback that can achieve the following two goals.

1) Make the re-ranking method invisible to users and does not need developers to do human-interaction activities.
2) Let our proposed method adaptive to different SBIR datasets without training process using labeled datasets.

Like our work, the just mentioned re-ranking method that utilizes RVFV and CBRF [6] is also an unsupervised method that uses blind feedback. Besides, the relevance ranking algorithm in [60] is fine-tuned to be a blind-feedback-based SBIR re-ranking algorithm. So, comparisons are made between the proposed method, the re-ranking method in [6] and our fine-tuned version of [60].

III. METHODOLOGY

Section III focuses on the methodology of this paper. In Section I and Fig. 1, we describe that our proposed SBIR re-ranking method is composed of 4 steps. In this Section, we describe the detailed principle of each step. Subsection A is a description of the flow of the existing SBIR systems, while the other subsections are the steps of the proposed method.

A. Initial SBIR

Given a query sketch q and an imagest $A = \{a_i\}_{i=1}^{I}$ with I natural images, an initial SBIR system performs an initial SBIR process for q. Most SBIR systems provide each natural image with an initial retrieval distance, and the distances are sorted in ascending order to be the initial retrieval results.

The general SBIR process for most of the existing SBIR systems is through the following steps:

1) *Feature Extraction*

Set a common feature for both the query sketch q and I natural images $A = \{a_i\}_{i=1}^{I}$, which makes the query sketch and natural images comparable. Then, features of the query sketch and each natural image are extracted.

2) *Feature Similarity Measurement*

After the features are extracted, feature similarity measurement is conducted to measure the similarity between the query sketch and each natural image. The majority of present SBIR systems' common feature is in the form of vectors. After this, feature distance between these vectors is calculated, as shown in Equation (1).

$$D(i) = \mathrm{dist}(f_q, f_i), i = 1, \dots, I \quad (1)$$

Where f_q is the feature vector of the query sketch q; f_i is the feature vector for the natural image a_i in the imageset $A = \{a_i\}_{i=1}^{I}$; dist(·) is the feature distance measurement function that is determined by a specific SBIR system; $D(i)$ is the feature distance between the query sketch q and the natural image a_i.

3) *Feature Similarity Ranking*

Now we have the feature distance between the query sketch q and every natural image in the imageset $A = \{a_i\}_{i=1}^{I}$. Most SBIR systems use Euclidean distance or hamming distance to measure the feature distance. For these two distance metric, the similarity degree of two features decreases while the feature distance rises. So, we obtain the initial retrieval results by means of sorting I feature distances obtained through Equation (1). The resulting sequence are marked as $R = \{r_i\}_{i=1}^{I}$, where r_i represents the feature distance between q and the i-th initial retrieval result. $B^N = \{b_i^N\}_{i=1}^{I}$ is the sorted version of $A = \{a_i\}_{i=1}^{I}$, where b_i^N is the natural image (i-th initial retrieval result) r_i refers to.

When the initial SBIR is done, the I natural images in the imageset are sorted to be the initial retrieval results. As far as re-ranking is concerned, we choose M highest-ranked initial retrieval results $B^N = \{b_i^N\}_{i=1}^{M}$ instead of all I natural images participating in the re-ranking process.

The reason for doing so is as the following. In image retrieval tasks, it is frequent that users focus more on the high-ranked images and want them to be the query sketch's relevant images. Considering that 1) users pay less attention to the low-ranked results, and 2) the query sketch's relevant images seem not to appear in the low-ranked retrieval results, we only choose the M highest-ranked initial retrieval results to participate in the re-ranking process.

B. Image Expansion

When the initial SBIR is done, we get M natural images $B^N = \{b_i^N\}_{i=1}^{M}$ as the top M initial retrieval results. These natural images go through the re-ranking process. Before we conduct feature extraction for the M initial retrieval results, we extract the edge maps and object images of them. This operation makes each initial retrieval result corresponds to three types of images: edge map, object image and natural image.

1) Extracting edge maps

Edge maps consist of black lines and white background, which represent the main outlines and contours for an object. One example of the edge map is shown in Fig. 2(a). We assume that when two objects belong to the same category, their main outlines and contours are also alike. Under this consideration, we extract edge maps of the natural images $B^N = \{b_i^N\}_{i=1}^{M}$. The resulting edge maps are marked as $B^S = \{b_i^S\}_{i=1}^{M}$, where b_i^S is the edge map of the i-th initial retrieval result's natural image b_i^N.

There are plenty of edge map extraction algorithms, such as Canny detector [34], Berkeley detector [33] and image token detector [35]. All these detectors are able to transfer a natural image into the form of edge map.

In this paper, Berkeley detector [33] is used. Berkeley detector provides each pixel with a probability of this pixel being an edge pixel. The higher the probability is, the more likely the pixel represents an edge pixel. We set 0.5 as a threshold. That is, pixels whose probability is no less than 0.5 are regarded as edge pixels by us. Thus, the resulting images, which are composed of black lines and white background, are edge maps used in this paper.

2) Generating object images

Most images have foreground information and background information. When people try to retrieve images that contain a particular object, they tend to think that the desired object appeared as a foreground object. Considering this, we generate object images $B^O = \{b_i^O\}_{i=1}^{M}$, which consist of only the salient foreground object and the black background, for the natural images $B^N = \{b_i^N\}_{i=1}^{M}$ of the initial retrieval results. b_i^O denotes the object image of the i-th initial retrieval result's natural image b_i^N. One example of the object images is shown in Fig. 2(b).

In order to generate an object image out of a natural image, we first extract the saliency map of the natural image. The function of a saliency map is to show the possibility of each pixel being the foreground pixel. There are many algorithms designed to obtain such saliency maps. For example, Liu et al. [39] proposed a salient object image termed as Saliency Tree; Cheng et al. [40] put forward a regional contrast based salient object detection algorithm.

In this paper, the salient object detection algorithm put forward in [40] is used as the detector to extract saliency maps for the initial retrieval results. In the rest of this paper, we denote this algorithm as RC algorithm. Saliency maps generated by RC algorithm are grayscale images, where the pixel value indicates how likely a pixel belongs to the foreground object. This value is an integer in the interval [0,255]. With the growth of the pixel value, the confidence of this pixel belonging to the salient object increases.

We then use Saliency maps to acquire the object images. To do this, we set a threshold to determine which pixels are foreground pixels. To be specific, those pixels in a saliency map whose pixel value is more than this threshold are taken as the foreground pixels; the other pixels are background pixels. For the foreground pixels, we make them the same as the corresponding ones in the form of natural image. For the background pixels, they all become black. In the experiments of this paper, the threshold value is 100.

For each initial retrieval result, Image Expansion now makes us have an edge map, an object image and a natural image. Fig. 2 is an example of such an expansion. Accordingly, the number of feature vectors triples. We can see from Fig. 2 that there are

(a) edge map (b) object image (c) natural image

Fig. 2 An example of the results of image expansion. The three subimages are a candle image's edge map, object image and natural image, respectively.

image domain differences between an edge map, an object image and a natural image.

Image Expansion adds two more images to each initial retrieval result. As a result, we have three image datasets for these initial retrieval results: the edge map image dataset $B^S = \{b_i^S\}_{i=1}^M$, the object image dataset $B^O = \{b_i^O\}_{i=1}^M$ and natural image dataset $B^N = \{b_i^N\}_{i=1}^M$.

C. Multi-clustering

At this time, each initial result corresponds to three images: one edge map, one object image and one natural image. In order to operate the clustering-based re-ranking, we extract features from these three image domains, and implement an unsupervised clustering algorithm for each image domain. The outputs of multi-clustering are clustering results for edge maps, object images and natural images, respectively.

1) Feature extraction

We first extract feature vectors for edge maps $B^S = \{b_i^S\}_{i=1}^M$, object images $B^O = \{b_i^O\}_{i=1}^M$ and natural images $B^N = \{b_i^N\}_{i=1}^M$. The aim is to give three feature vectors to each initial retrieval result. For each initial retrieval result, the three feature vectors are for edge map, object image and natural image, respectively.

There are plenty of features trying to represent the characteristics of images. Early image feature extraction methods use geometric features to represent images, such as HoG feature [15], ARP feature [41], AROP feature [18], [19], SIFT feature [42]. Recently, with the rapid development of deep learning techniques, image features acquired from deep learning methods exhibit great performance on various image processing tasks, such as image recognition, image retrieval, and object detection. Among the deep learning techniques, Convolutional neural network (CNN) performs well on image-related works. AlexNet [43], VGGNet [44], GoogLeNet [45] and ResNet [46], as state-of-the-art CNNs, ranked at the top in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) for its image classification tasks.

In this paper, we propose an unsupervised SBIR re-ranking method. Given that we cannot train a machine learning model specially fitting the target dataset, choosing a feature extraction method generally performing well is a good choice. Although the geometric image features are available, deep learning models pre-trained on the ImageNet or other datasets are preferred to be the image feature extraction tool.

Once the image expansion is done, the features $F^S = \{f_i^S\}_{i=1}^M$, object image features $F^O = \{f_i^O\}_{i=1}^M$ and natural image features $F^N = \{f_i^N\}_{i=1}^M$ are gained. f_i^S denotes the feature vector of the edge map of the *i*-th initial retrieval result. Similar ways are used for f_i^O and f_i^N.

2) Multi-clustering

At this step, we implement clustering methods on the just extracted features of the top M initial retrieval results. Three clustering methods are used. One is for M edge map features, one is for M object images, and one is for M natural images. Finally we have three clustering results, each for an image domain.

Clustering, which is a commonly used unsupervised machine learning method, can automatically divide some features into several groups (clusters). Clustering algorithms assume that images that are in the same clusters tend to be similar, and they assume that images that are from different clusters tend to be dissimilar. Under this assumption, we use clustering methods to divide the initial results into several clusters.

The Multi-clustering is conducted through the following steps:

Step 1: Clustering methods selection

In this step, we choose three clustering methods to do the clustering for edge maps, object images and natural images, respectively. The three clustering methods can either be identical or different methods.

Some of the clustering methods require users or developers to give an exact number of the desired clusters, while the others do not require the users or developers to do so. We desire to propose an SBIR re-ranking method that uses blind feedback, so the users and developers should not participate in the re-ranking process. Therefore, the number of clusters should automatically adjust to the characteristics of the initial retrieval results and the selected clustering methods. So, the clustering methods that can automatically get its number of clusters during its clustering process are selected. In other words, in Multi-clustering of this paper, the number of clusters is decided automatically by the clustering method.

Step 2: Multi-clustering

Three clustering methods are then used to do clustering for each image domain. After we do Image Expansion and Feature Extraction, we have M feature vectors $F^S = \{f_i^S\}_{i=1}^M$ for edge maps, M feature vectors $F^O = \{f_i^O\}_{i=1}^M$ for object images and M feature vectors $F^N = \{f_i^N\}_{i=1}^M$ for natural images for M initial retrieval results. The selected clustering method for edge maps are implemented on the M edge map features. And the other two selected clustering methods are implemented correspondingly to their image domains.

The clustering method conducted on the edge map features $F^S = \{f_i^S\}_{i=1}^M$ brings edge clusters $C^S = \{c_k^S\}_{k=1}^{l^S}$. Each cluster c_k^S contains some edge maps of the initial retrieval results; l^S is the number of clusters acquired from the clustering method used for edge map features. Likewise, the other two clustering methods provide object clusters $C^O = \{c_k^O\}_{k=1}^{l^O}$ to object image features $F^O = \{f_i^O\}_{i=1}^M$ and natural clusters $C^N = \{c_k^N\}_{k=1}^{l^N}$ to natural image features $F^N = \{f_i^N\}_{i=1}^M$. l^O is the number of clusters acquired from the clustering method used for object features, and l^N is the number of clusters acquired from the clustering method used for natural image features. Each cluster c_k^Z ($Z \in \{S, O, N\}$) has $|c_k^Z|$ initial retrieval results, and we have $\sum_{k=1}^{l^Z} |c_k^Z| = M$.

D. Clustering-based Re-ranking

This subsection describes the SBIR re-ranking process based on the just obtained clusters for edge maps, object images and natural images. The initial retrieval results are rearranged here.

As a result, we get the final SBIR retrieval results. Algorithm 1 gives the summarization of this subsection's algorithms.

The whole re-ranking process is with the following three steps: cluster importance evaluation, cluster score calculation and multi-modal re-ranking.

Step 1: Cluster importance evaluation

Cluster score evaluation evaluates the importance of each cluster of the edge clusters $C^S = \{c_k^S\}_{k=1}^{l^S}$, object clusters $C^O = \{c_k^O\}_{k=1}^{l^O}$ and natural clusters $C^N = \{c_k^N\}_{k=1}^{l^N}$. In multi-clustering, the top *M* initial retrieval results are divided into several clusters in each image domain. Since each image domain has several clusters, the clusters which are similar to the query sketch q are more important than others during re-ranking. So, we provide a cluster importance to each cluster.

For each cluster c_k^Z ($Z \in \{S, O, N\}$, $k \in \{1,2,\dots,l^Z\}$), its cluster importance p_k^Z is set as the reciprocal of the mean value of the initial feature distance of the first half of initial retrieval results inside this cluster, as shown in Equation (2).

$$p_k^Z = 1/\frac{\sum_{i=1}^{\left\lceil |c_k^Z|/2+1 \right\rceil} x_{i,k}^Z}{\left\lceil |c_k^Z|/2+1 \right\rceil} = \frac{\left\lceil |c_k^Z|/2+1 \right\rceil}{\sum_{i=1}^{\left\lceil |c_k^Z|/2+1 \right\rceil} x_{i,k}^Z} \quad (2)$$

where $|c_k^Z|$ represents the number of initial retrieval results inside c_k^Z. $x_{i,k}^Z$ denotes the initial feature distance of the initial retrieval result that is ranked *i*-th among all the initial retrieval results in the cluster c_k^Z. That is, $x_{i,k}^Z \in (R = \{r_i\}_{i=1}^{l})$, and it is the initial retrieval distance outputted from an SBIR system.

$1/p_k^Z$ is the mean value of the initial feature distance of the first half of initial retrieval results inside c_k^Z. The reason for only dealing with the first half rather than all the initial retrieval results is that the first half of initial retrieval results of a cluster can often represents the characteristics of this cluster well. We can see that the lower $1/p_k^Z$ is, the lower the mean value of the initial feature distance is. It is likely that the lower initial feature distance means greater similarity, so the cluster importance increases as $1/p_k^Z$ decreases. Considering that we tend to think that higher values represent greater importance, the reciprocal of $1/p_k^Z$ is finally set to be the cluster importance of c_k^Z.

After Eq. (2) is conducted on all clusters, we have the edge cluster importance values $P^S = \{p_k^S\}_{k=1}^{l^S}$, the object cluster importance values $P^O = \{p_k^O\}_{k=1}^{l^O}$ and natural cluster importance values $P^N = \{p_k^N\}_{k=1}^{l^N}$.

Step 2: Cluster score calculation

For every initial retrieval result, the re-ranking process takes both its initial feature distance and its cluster score into account. With cluster importance, this step focuses on generating the cluster score for every initial retrieval result.

An initial retrieval result has three cluster importance values: edge cluster importance, object cluster importance and natural cluster importance. Generally speaking, natural images and object images has more semantic information than edge maps does. So, we should pay different attention to clusters of different image domains during re-ranking. In addition, since different clusters have different cluster importance, the clusters whose cluster importance is higher need to hold a cluster score that can influence the re-ranking process more.

Therefore, a domain weight is set to each image domain to represent the significance of this domain, and a domain cluster score is set to each cluster based on its cluster importance. The cluster score for the initial retrieval results is calculated with the aid of the domain weight and the domain cluster score.

Domain Weight: We set the edge domain weight as w_S, the object domain weight as w_O and the natural domain weight as w_N, and we set $w_N + w_O + w_S = 1$. Since natural images have the most semantic information and edge maps have the least, let $w_N \geq w_O \geq w_S$ would be a proper choice.

Domain Cluster Score: We assume that an initial retrieval result is clustered into the edge cluster c_α^S, the object cluster c_β^O and the natural cluster c_γ^N, respectively. Consequently, the corresponding cluster importance values are p_α^S, p_β^O and p_γ^N. Each of these three cluster importance values is used to get its corresponding domain cluster score.

Under these assumptions, we first sort the cluster importance of each image domain in descending order. That is, we sort the elements of importance values $P^Z = \{p_k^Z\}_{k=1}^{l^Z}$ of each image domain Z ($Z \in \{S, O, N\}$) in descending order.

After sorting, if the initial retrieval result is clustered into c_θ ($c_\theta \in \{c_\alpha^S, c_\beta^O, c_\gamma^N\}$), we find the rank of its cluster importance $p_\theta \in \{p_\alpha^S, p_\beta^O, p_\gamma^N\}$ in the sorted P^Z. If p_θ is ranked t^Z-th, we use the Equation (3) to get its edge domain cluster score e^Z.

$$e^Z = \begin{cases} 1 + (t^Z - 1)/(l^Z - 1), & l^Z > 1 \\ 1, & l^Z = 1 \end{cases} \quad (3)$$

Then, Eq. (3) is used for three image domains, and this initial retrieval result has three domain cluster scores: e^S, e^O and e^N.

The consideration behind using Eq. (3) is as the follows.

First, for an initial retrieval result, Eq. (4) and Eq. (5) implies that e^Z is multiplied by its initial feature distance r_i to get the final retrieval score. Thus, to fit the monotony of the initial feature distance $R = \{r_i\}_{i=1}^{l}$, the smaller value of e^Z is preferred. So, we let the value of e^Z increases as t^Z grows.

Second, to balance the influence of cluster scores against that of the initial feature distance, the range of e^Z is crucial. The lower limit of this interval should not be too small, and the upper limit should not be too great. Otherwise, the initial feature distance, which is an important element during re-ranking, may become useless. To avoid this problem, in our paper, the range of e^Z is the interval [1,2]. In Eq. (3), $(t^Z - 1)/(l^Z - 1)$ is a value in the interval [0,1], and the constant "1" is set to change the range of e^Z from [0,1] to [1,2]. With $w_N + w_O + w_S = 1$, the final retrieval score belongs to $[r_i, 2r_i]$. In this way, the cluster score does not change the value of the initial feature distance overly. Thus, the proposed re-ranking method makes the final retrieval score give consideration to both the role of cluster scores and that of the initial feature distance, which makes the re-ranking method a stable one that is able to increase the performance of different initial retrieval results generated by different SBIR systems.

Third, given that l^Z (the number of clusters inside an image domain) is decided automatically by the clustering method, the smaller l^Z often implies that the difference between the clusters of this domain is greater. So, we require the differences between the candidate values of e^Z increase as l^Z reduces. In order to deal with this issue, Eq. (3) makes the candidate values of e^Z be an arithmetic progression $\{1, 1 + \frac{1}{l^Z - 1}, \dots, 1 + \frac{l^Z - 2}{l^Z - 1}, 2\}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2959875, IEEE Transactions on Circuits and Systems for Video Technology

Thus, the differences between the candidate values of e^Z can be automatically adjusted to l^Z.

Cluster Score: We take the domain weights and the domain cluster scores of three image domains into account, and Equation (4) is to calculate the cluster score g_i of the *i*-th initial retrieval result.

$$g_i = w_S \cdot e^S + w_O \cdot e^O + w_N \cdot e^N \quad (4)$$

Using the form of weighted sum, Eq. (4) gives consideration to all three domain clusters scores. When one domain cluster score is not satisfying, the other two domain cluster scores can often help alleviate the influence of the unsatisfying cluster score. As a result, our re-ranking method becomes more stable and can adjust to different SBIR systems and different initial retrieval results.

Step 3: Multi-modal Re-ranking

The initial feature distance and its corresponding cluster score are used to gain the final retrieval score of the initial retrieval results. For the *i*-th initial retrieval result, its initial feature distance is r_i, and its cluster score is g_i. Equation (5) is used to calculate the final retrieval score u_i.

$$u_i = r_i \cdot g_i, i = 1, \dots, M \quad (5)$$

After the final retrieval scores of all the *M* initial retrieval results are calculated, we have a sequence that contains *M* final retrieval scores. The sequence is sorted in ascending order to be the final re-ranking results.

Algorithm 1 Clustering-based Re-ranking

Input: Edge clusters $C^S = \{c_k^S\}_{k=1}^{l^S}$; object clusters $C^O = \{c_k^O\}_{k=1}^{l^O}$; natural clusters $C^N = \{c_k^N\}_{k=1}^{l^N}$; initial retrieval results distances $R = \{r_i\}_{i=1}^{I}$.

Output: The final retrieval results.

1: Cluster importance evaluation. Provide a cluster importance to each cluster in C^S, C^O and C^N.
  1.1: **For** $k = 1, \dots, l^s$ **do**
  1.2: Get the cluster importance p_k^S of the cluster c_k^S through Eq. (2).
  1.3: **End**
  1.4: Implement similar actions for each cluster in C^O and C^N.
  1.5: Record edge cluster importance $P^S = \{p_k^S\}_{k=1}^{l^S}$, object cluster importance $P^O = \{p_k^O\}_{k=1}^{l^O}$ and natural cluster importance $P^N = \{p_k^N\}_{k=1}^{l^N}$.
2: Cluster score calculation. Provide each initial retrieval result $r_i \in R$ a cluster score.
  2.1: Set domain weights w_N, w_O and w_S satisfying $w_N + w_O + w_S = 1$ and $w_N \geq w_O \geq w_S$.
  2.2: Sort the elements of importance values P^S in descending order. Similarly, sort the elements of P^O and P^N.
  2.3: **For** $i = 1, \dots, M$ **do**
  2.4: The *i*-th initial retrieval result has cluster importance for three image domains: $p_\alpha^S, p_\beta^O, p_\gamma^N$. Find the rank of $p_\alpha^S, p_\beta^O, p_\gamma^N$ inside the sorted P^S, P^O and P^N, respectively. The ranks are marked as t^S, t^O and t^N.
  2.5: Use Eq. (3) to get the domain cluster scores e^S, e^O and e^N.
  2.6: Use Eq. (4) to get the cluster score g_i for the *i*-th initial retrieval result.
  2.7: **End**
3: Re-ranking.
  3.1: **For** $i = 1, \dots, M$ **do**
  3.2: Use Eq. (5) to get the final retrieval score u_i for the *i*-th initial retrieval result.
  3.3: **End**
  3.4: Sort the sequence $\{u_i\}_{i=1}^{M}$ in ascending order. The images that the resulting sequence refers to are the final retrieval results.

IV. Experiments

In this section, experiments are conducted to show the effectiveness of our SBIR re-ranking method. To be specific, three initial SBIR systems are used to implement initial SBIR, and our re-ranking method rearranges the initial retrieval results of each initial SBIR system. Besides, we compare the re-ranking performance of our SBIR re-ranking method to that of another two unsupervised blind-feedback-based SBIR re-ranking methods proposed in [6] and [60]. Experimental results demonstrate that our SBIR re-ranking is helpful.

A. Initial SBIR Systems and Comparative Methods

There are four SBIR systems used for initial SBIR: TripAlex, GN Triplet, DSH and SCMR. The natural images' feature distances are sorted in ascending order to get the initial retrieval results. DSH uses Hamming distance to calculate the feature distance, while the others use Euclidean distance.

- **TripAlex.** This network is trained by feeding into a lot of triplets. A triplet has a query image, a positive image and a negative image. A positive image is a natural image that is relevant to the query sketch; a negative image is a natural image that is irrelevant to the query sketch. The network of TripAlex is composed of three identical AlexNets (marked as Alex_S, Alex_P and Alex_N). Alex_S is for sketches, Alex_P is for positive images and Alex_N is for negative images. During training, the total loss function is $L = L_C(\text{Alex\_S}) + L_C(\text{Alex\_P}) + L_C(\text{Alex\_N}) + 0.01L_T$. $L_C(\cdot)$ is the classification loss; L_T is the Triplet loss provided in [61]. We train one caffemodel for Sketchy Extension and another caffemodel for TU-Berlin Extesnion. After training, the output of 'fc7' layer of the Alex_S and Alex_P is taken as the feature for sketches and natural images, respectively.
- **GN Triplet.** GN Triplet is the top-performing SBIR method proposed in [30]. The GN Triplet caffemodel released by the authors is used to extract the feature vectors of both sketches and natural images. A triplet loss and three classification losses are used to train this model three GoogLeNets. One GoogLeNet is for sketches, the other two are for natural images. [30] proposes an large s-cale fine-grained SBIR dataset named Sketchy. Considering that one of our testing dataset is the extension version of Sketchy, our SBIR re-ranking method has to be effective for the top-performing SBIR method for Sketchy dataset in [30].
- **DSH.** It is a hashing-based SBIR method [31]. Three CNNs are trained for this method. One is for sketches, one

is for sketch tokens (an alternative of edge maps) and one is for natural images. The hash codes of sketches are extracted from the CNN for sketches, and the other two CNNs work together to generate the hash codes for natural images. The authors of [31] released two caffemodels. One is trained for Sketchy Extension dataset, and another one is trained for TU-Berlin Extension dataset. Since these two datasets are exactly the datasets used in this paper, DSH on each dataset uses the corresponding caffemodel to extract the 128-bit hash codes to be the features for sketches and natural images.

- **SCMR.** It is a deep framework that uses a hybrid multi-stage training networks [47]. For training this final caffemodel for extracting image features, four stages of training is implemented. The first stage trains two CNNs separately with softmax losses. In the second stage, the shared layers of two branches are trained by softmax losses and a contrastive loss. The third stage trains the frozen layers with softmax losses and a triplet loss. The public caffemodel released by the authors are taken as the feature extraction tools to extract the features of sketches and natural images.

Besides, the comparisons between the following re-ranking methods are made.

- **R.** Our proposed re-ranking method.
- **IRC.** The SBIR re-ranking method proposed in [6] presented in Section II.
- **BR.** The fine-tuned version of the relevance ranking method in [60]. In [60], $Y = \{y_i\}_{i=1}^n$ is set as the semantic scores between a query tag and images $\{x_i\}_{i=1}^n$, and $W_{ij} = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$ is set as the similarity degree between visual features of the images x_i and x_j. The closed-form solution, $F^* = \frac{C}{1+C}\left(I - \frac{1}{1+C}D^{-\frac{1}{2}}WD^{-\frac{1}{2}}\right)^{-1}Y$, is used to get the relevance scores $F^* = \{f_i\}_{i=1}^n$ to replace Y. $D_{ii} = \sum_{j=1}^n W_{ij}$; C is a constant. Finally, F^* is sorted to be the final retrieval result.

  To convert this algorithm into an SBIR re-ranking method to re-rank the *M* initial retrieval results of an initial SBIR systems, we convert Y into $Y = \{exp\left(-\frac{\|r_i\|^2}{2\sigma^2}\right)\}_{i=1}^M$, where $r_i \in R$ is the initial feature distance of the *i*-th initial retrieval result of an initial SBIR system. Then, the closed-form solution can be used for SBIR re-ranking.

  In this paper, we let $C = 0.3$; 'pool5/7x7_s1' features of the GoogLeNet caffemodel that has been pre-trained on ImageNet dataset [51] are used to calculate the visual similarity degree W_{ij}.

B. Datasets

The experiments are conducted on two datasets: Sketchy Extension [30], [31] and TU-Berlin Extension [48], [49].

1) Sketchy Extension

Sketchy [30] is a newly released fine-grained SBIR dataset. It contains 125 image categories, such as airplane, apple, bear, dolphin, eyeglasses, pig, sheep and strawberry. Each category has 100 natural images, which forms a natural image dataset with in total 12,500 natural images. In order to build the sketch dataset, volunteers were invited to draw 75,471 hand-made sketches exactly for these 12,500 natural images.

After this, [31] collect another 60,502 natural images, making the number of natural images rise to 73,002. Thus, the Sktechy dataset is expanded into Sketchy Extension dataset.

In this paper, 7,583 sketches (roughly 60 per category) and 14,600 natural images (roughly 117 per category) are randomly selected to be the dataset for our testing. The other sketches and natural images are used for training TripAlex caffemodel for this dataset.

2) TU-Berlin Extension

TU-Berlin dataset [48] is a benchmark sketch dataset. It has 250 categories, including axe, harp, owl, pistol, ray, seal and so on. Each category has 80 sketches, and in total there are 20,000 sketches.

In addition, [49] allocates 204,489 natural images for these categories (in average about 818 per category). This makes the TU-Berlin dataset expand to the TU-Berlin Extension dataset.

We randomly choose 2,000 sketches and 40,898 natural images from these images to be the testing dataset. As a result, each category has 8 sketches and approximately 163 natural images. The other sketches and natural images are used for training TripAlex caffemodel for this dataset. To increase the number of sketches for training, the edge maps of those natural images used for training are used.

C. Implementation Details

The open source Caffe [50] deals with extracting image features from CNN models, and Matlab2014a realizes re-ranking. Both Caffe and Matlab are implemented on the Ubuntu 14.04.

During Multi-clustering, the GoogLeNet caffemodel that has been pre-trained on ImageNet dataset [51] is used to extract the 'pool5/7x7_s1' features from edge maps, object images and natural images in feature extraction.

The clustering method used in this paper for multi-clustering is Affinity propagation Clustering Algorithm [52]. In addition, the domain weights w_S, w_O and w_N in cluster score calculation in Clustering-based Re-ranking are 0.1, 0.3 and 0.6, respectively.

For R method and BR method, the 100 highest-ranked initial retrieval results participate in the re-ranking process. That is, the *M* is set as 100.

D. Performance Evaluation

Just like our previous work [6], [18], [19], [38], we use the precision under depth *x* (denoted as *Precision@x*) to measure the performance of all the methods. *Precision@x* is defined as follows:

$$Precision@x = \frac{1}{L}\sum_{m=1}^{L}\frac{1}{x}\sum_{i=1}^{x}R_m(i) \quad (6)$$

where $R_m(i)$ is the relevance of the *i-th* result for query m, $i \in [1,2,\cdots,x]$, and $m \in [1,2,\cdots,L]$. If the *i-th* result is relevant to the query sketch, $R_m(i) = 1$. Otherwise, $R_m(i) = 0$.

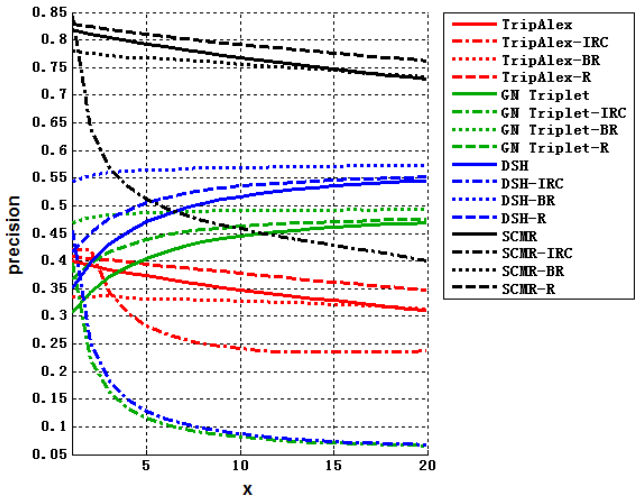
E. Objective Comparisons

In order to make a fair comparison, the relative parameters d-

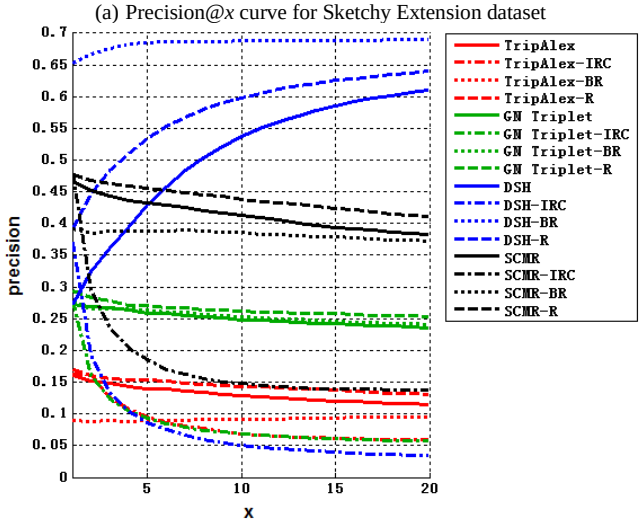
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2959875, IEEE Transactions on Circuits and Systems for Video Technology

uring the running of GN Triplet [30], DSH [31] and SCMR [47] are set the same as the ones in [30], [31] and [47], respectively.

Precision@x curves for depth in the range of [1,20] in Fig. 3 show the performance of R method and other comparative methods. It contains the average retrieval precision of the top-20 results.



(a) Precision@*x* curve for Sketchy Extension dataset



(b) Precision@*x* curve for TU-Berlin Extension dataset

Fig. 3 Precision comparison for two datsets. 'TripAlex', 'GN Triplet', 'DSH' and 'SCMR' denotes the accuracy of initial retrieval results of the initial SBIR based on each corresponding method. '-IRC' is the result of using the re-ranking method 'IRC'. '-BR' is the result of using the re-ranking method 'BR'. '-R' is the result of using the re-ranking method of this paper.

Fig. 3 shows that our R method is effective on both datasets. It can be seen that after our re-ranking operation, the retrieval performance of all the top-20 results increases. The most distinguish rise occurs at the DSH for TU-Berlin dataset, where the top-1 retrieval accuracy increases over 10% by the R method. Given that the SCMR model uses the Sketchy dataset to train its model, all the top-20 retrieval accuracy of initial SBIR of SCMR on Sketchy Extension (the red solid line of Fig. 3 (a)) is over 75%. Although this performance is very good, R method still receives a performance improvement.

As for BR method, the re-ranking performance is unstable. In terms of these 8 initial SBIR systems, R method performs better than BR in 5 of them. For these 5 groups, the retrieval accuracy of BR re-ranking is often worse than that of initial retrieval results. For the other 3 groups, BR performs better. The reasons for this phenomenon lie in the optimization logic behind BR. BR prefers those initial retrieval results that have greater similarity to all the other initial retrieval results. As a result, no matter whether these initial retrieval results are relevant images, they are preferred by BR. As the output relevance scores of the optimization are not kept within a limited range according to initial feature distances, we can often see that the influence of the initial feature distances is weaken, while the initial retrieval results that have greater similarity to all the other initial retrieval results are brought to the front. So, if there are enough relevant images in the initial retrieval results participating in re-ranking, the re-ranking performance of BR tends to be good. Otherwise, the re-ranking performance is often not satisfying. For DSH on both datasets and GN triplet on Sketchy Extension, more than half of the initial retrieval results participating in re-ranking are relevant images, which makes their performance good. With respect to the corresponding performance of R method, as the final retrieval scores are kept within the range $[r_i, 2r_i]$, the rearrangement extent of R method is weaker than that of BR. As a result, the corresponding performance of our re-ranking method is not as good as that of BR. Nevertheless, that the final retrieval scores of our re-ranking method are inside an exact range helps us alleviate the problem of rearranging overly, which makes our re-ranking method a stable one. We can see that R method benefits all the 8 initial retrieval instances.

We can see from Fig. 3 that the performance of the IRC is not satisfying. Although the retrieval performance of top-1 has a small rise, the precision curves of the re-ranked results on the other ranks experience somewhat decrease compared with their corresponding ones of initial SBIR. The RVFV re-ranking method of IRC first extracts the edge map of the several top retrieval results of initial SBIR, and then the feature similarity of all the initial retrieval results and the selected several top retrieval results are compared. So, the performance of the top-1 re-ranked results does not vary a lot. However, since IRC needs to uses the features of the edge maps to do feature similarity measurement, the irrelevant initial retrieval results whose edge maps are similar to the ones of the relevant initial retrieval results are also preferred. Therefore, some irrelevant initial retrieval results are brought to the front.

V. Discussion

In this section, we discuss the influences of the parameter values during the re-ranking process. The following factors are considered:

1) the feature extraction method in Multi-clustering
2) parameter *M*: the number of initial retrieval results participating in our SBIR re-ranking method
3) parameters w_S, w_O and w_N: the domain weights in Cluster score calculation in Clustering-based Re-ranking

In addition to *Precision@x*, a performance indicator *AP(K)*, which is the average of *K* top-ranked points in *Precision@x* curve, is used as an SBIR performance indicator, as shown in Equation (7).

$$AP(K) = \frac{1}{K}\sum_{x=1}^{K} Precision@x \qquad (7)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2959875, IEEE Transactions on Circuits and Systems for Video Technology

TABLE I
THE IMPACT OF USING DIFFERENT CNN MODELS AS FEATURE EXTRACTION TOOLS ON THE PERFORMANCE OF IMPLEMENTING RE-RANKING ALGORIHTM ON DIFFERENT SBIR SYSTEMS

CNN Model for Feature Extraction	Initial Retrieval Method	AP(10)	
		Sketchy Extension	TU-Berlin Extension
AlexNet	TripAlex	38.8%	14.2%
	GN Triplet	43.4%	26.8%
	DSH	50.0%	52.9%
	SCMR	80.1%	44.8%
VGG-16	TripAlex	39.0%	14.5%
	GN Triplet	43.0%	26.9%
	DSH	49.7%	52.2%
	SCMR	80.2%	44.8%
GoogLeNet	TripAlex	39.3%	15.3%
	GN Triplet	43.3%	27.2%
	DSH	49.7%	52.6%
	SCMR	80.9%	45.5%
ResNet-50	TripAlex	39.0%	15.5%
	GN Triplet	43.4%	27.5%
	DSH	50.4%	53.1%
	SCMR	79.9%	45.5%

A. The feature extraction method in Multi-clustering

CNN caffemodels pre-trained on ImageNet dataset are used to be the feature extraction tool in Multi-clustering. The classic CNNs contain AlexNet, VGGNet, GoogLeNet, ResNet and so on. The Caffemodels for these CNNs all achieve top performance for image classification tasks on the ImageNet Large Scale Visual Recognition Competition (ILSVRC). So, the caffemodels, which are the ones pretrained by ImageNet dataset, for AlexNet [62], VGG-16 [63] and ResNet-50 [64] are also used to extract image features in Table I. The features of AlexNet and the VGG-16, which are used for multi-clustering, are the outputs of the 'fc7' layer; the ResNet-50 uses the features of 'pool5' layer.

From Table I, we can observe that the re-ranking performance of using three CNN models is not so different from each other. Generally speaking, the semantic understanding ability of GoogLeNet, VGG-16 and ResNet-50 is better than that of AlexNet. So, when extracting features of natural images, GoogLeNet and VGG-16 have a better semantic understanding. Nevertheless, AlexNet is also a CNN that has strong ability to understand semantic information of images. When it comes to edge maps and object images, since the visual structures of edge maps and object images are simpler than those of natural images, the semantic understanding ability of AlexNet, VGG-16 and GoogLeNet are at a similar level. Considering that edge maps and object images also play important roles in re-ranking, the importance of natural images lowers. Consequently, the re-ranking performance of using four different CNN caffemodels is not so different from each other.

B. The number of initial retrieval results in re-ranking

Instead of taking all the initial results into consideration during our re-ranking method, we only let the *M* highest-ranked initial retrieval results participate re-ranking. In this section, the effects of changing the value of *M* are discussed in Table II.

TABLE II
THE PERFORMANCE OF RE-RANKING DIFFERENT SBIR METHODS UNDER DIFFERENT *M*

Dataset	Method	AP(10)			
		Without Re-ranking	*M*=20	*M*=50	*M*=100
Sketchy Extension	TripAlex	37.1%	38.2%	38.9%	39.3%
	GN Triplet	39.8%	41.1%	42.4%	43.3%
	DSH	46.1%	48.1%	49.0%	49.7%
	SCMR	78.9%	79.6%	80.3%	80.9%
TU-Berlin Extension	TripAlex	14.1%	15.0%	15.3%	15.7%
	GN Triplet	25.8%	26.4%	26.8%	27.2%
	DSH	42.9%	47.8%	50.2%	52.6%
	SCMR	43.1%	44.0%	44.6%	45.5%

Table II shows that with the increase of *M*, the AP(10)s of re-ranking different SBIR systems experience a growth. The quantity of images in both datasets is relatively large, where we have hundreds of images for per image category. As a result, the number of relevant images in the initial retrieval results rises as *M* grows. Since multi-clustering needs as many relevant images as possible in initial retrieval results, the rise of *M* results in the improvement of the re-ranking performance.

C. The domain weights in Cluster score calculation

During clustering-based re-ranking, clustering results of three image domains (edge maps, object images and natural images) are used to rearrange the initial retrieval results. The domain weights w_S, w_O and w_N judge the importance of every image domain. In order to get a better re-ranking method, we need to set proper values for w_S, w_O and w_N.

Generally speaking, edge maps have less semantic information than object images and natural images, and object images' semantic information is less than natural images'. Intuitively, letting $w_S \leq w_O \leq w_N$ is a reasonable choice.

Taken the initial SBIR system 'DSH' as an example, we display the re-ranking performance under different domain weights in Table III. In Table III, we first gives the AP(10) of only using cluster results of natural images to re-rank the initial retrieval results in the second column. With $w_S \leq w_O \leq w_N$,

TABLE III
THE PERFORMANCE OF RE-RANKING INITIAL RETRIEVAL RESULTS OF 'DSH' UNDER DIFFERENT w_S, w_O *and* w_N

	AP(10)		AP(10)				
	Only Natural Images ($w_S=0, w_O=0, w_N=1$)		$w_O=0.1$	$w_O=0.2$	$w_O=0.3$	$w_O=0.33$	$w_O=0.4$
Sketchy Extension	48.2%	$w_S=0.1$	49.3%	49.5%	49.7%	/	49.8%
		$w_S=0.2$	/	49.7%	49.8%	/	49.8%
		$w_S=0.33$	/	/	/	49.9%	/
TU-Berlin Extension	50.3%	$w_S=0.1$	51.9%	52.2%	52.6%	/	52.7%
		$w_S=0.2$	/	52.5%	52.8%	/	52.8%
		$w_S=0.33$	/	/	/	53.1%	/

we give the AP(10) of re-ranking under different combinations of domain weights in the columns thereafter; in order to make the table concise, we merely present the values of w_S and w_O in these columns, and w_N can be calculated out by $w_N = 1 - w_S - w_O$.

Table III reveals that the performance of re-ranking varies with different image domain weights. When there is a proper combination of $w_S \leq w_O \leq w_N$, our SBIR re-ranking method performs better. In comparison to using clustering results of natural images only ($w_S = w_O = 0, w_N = 1$) to do re-ranking, the introduction of clustering results of edge maps and natural images benefits the retrieval accuracy of re-ranking. The top-performed results for both datasets appear when three image domains are of the same importance ($w_S = w_O = w_N$). Accordingly, setting $w_S = w_O = w_N$ is a proper strategy for our SBIR re-ranking method.

D. Subjective Comparisons

We applied our proposed SBIR re-ranking method on two image datasets and four initial SBIR systems. Besides, the IRC method and BR method are also conducted to be comparative methods. Fig. 4 and Fig. 5 give the initial retrieval results and re-ranking results on the Sketchy Extension dataset and TU-Berlin Extension dataset, respectively.

Fig. 4 and Fig. 5 display that our SBIR re-ranking is effective for various initial SBIR systems. After our re-ranking method is implemented, the two highest-ranked ones of the majorities of re-ranked results are correct. When the initial retrieval results are good (7 or more relevant images in top-10 initial retrieval results), our re-ranking method can make the performance even better. At this time, our re-ranking method can often make all the top-10 retrieval results correct.

It can also be seen that the performance of IRC method is not ideal. The reason is that the IRC method leverages the features of the edge maps of natural images. Since the edge maps contain much less semantic information than the natural images, the incorrect images whose edge maps are similar to correct ones are often put in front of the correct images.

The performance of BR method is not stable. Sometimes, it benefits the initial SBIR. Sometimes, it does not. As stated in Section IV E, the reasons lie in the optimization logic of BR.

VI. Conclusion

We propose an unsupervised blind-feedback-based SBIR re-ranking method to improve the retrieval performance of various SBIR systems. First, initial SBIR is conducted to get the initial retrieval results. Then, image expansion is used to get the image features of three image domains: edge map, object image and natural image. Next, we perform multi-clustering to cluster the features of the edge map domain, the object image domain and the natural image domain. Finally, clustering-based re-ranking re-ranks the initial retrieval results according to the outputs of multi-clustering. Experiments on different datasets and different SBIR systems reveal that our proposed re-ranking method is valid and effective.

References

[1] R. Hong, Y. Yang, M. Wang, et al., "Learning Visual Semantic Relationships for Efficient Visual Retrieval," IEEE Trans. Big Data, 2015, 1(4): 152-161.

[2] Z. Lin, G. Ding, J. Han, et al., "Cross-view retrieval via probability-based semantics-preserving hashing," IEEE transactions on cybernetics, 2017, 47(12): 4342-4355.

[3] Z. Guan, F. Xie, W. Zhao, et al., "Tag-based weakly-supervised hashing for image retrieval," International Joint Conference on Artificial Intelligence (IJCAI), 2018, 3776-3782.

[4] L. Zhu, J. Shen, L. Xie, "Unsupervised Visual Hashing with Semantic Assistant for Content-based Image Retrieval," IEEE Transactions on Knowledge & Data Engineering, 2017, 29(2), 472-486.

[5] Z. Xia, X. Wang, L. Zhang, et al., "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," IEEE Transactions on Information Forensics and Security, 2016, 11(11): 2594-2608.

[6] X. Qian, X. Tan, Y. Zhang, R. Hong, and M. Wang, "Enhancing Sketch-Based Image Retrieval by Re-ranking and Relevance Feedback," IEEE Trans. Image Processing, 2016. pp.195-208.

[7] R. Hong, L. Li, J. Cai, et al., "Coherent Semantic-Visual Indexing for Large-Scale Image Retrieval in the Cloud," IEEE Trans. Image Processing, 2017, 26(9): 4128-4138.

[8] R. Hong, Z. Hu, R. Wang, et al., "Multi-View Object Retrieval via Multi-Scale Topic Models," IEEE Trans. Image Processing, 2016, 25(12): 5814-5827.

[9] Y. Guo, G. Ding, J. Han, "Robust Quantization for General Similarity Search," IEEE Trans. Image Processing, 2018, 27(2): 949-963.

[10] P. Liu, J. Guo, C. Wu, et al., "Fusion of Deep Learning and Compressed Domain Features for Content-Based Image Retrieval," IEEE Transactions on Image Processing, 2017, 26(12): 5706-5717.

[11] Y. Matsui, K. Ito, Y. Aramaki, et al., "Sketch-based manga retrieval using manga109 dataset," Multimedia Tools and Applications, 2017, 76(20): 21811-21838.

[12] Y. Cao, C. Wang, L. Zhang, and L. Zhang, "Edgel index for large-scale sketch-based image search," IEEE CVPR, 2011, pp. 761-768.

[13] Y. Qi, Y. Song, H. Zhang, et al, "Sketch-based image retrieval via Siamese convolutional neural network," Image Processing (ICIP), 2016 IEEE International Conference on. IEEE, 2016: 2460-2464.

[14] R. Zhou, L. Chen, and L. Zhang, "Sketch-based image retrieval on a large scale database," ACM MM, 2012, pp. 973-976.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in CVPR, 2005, pp.886-893.

[16] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "An evaluation of descriptors for large-scale image retrieval from sketched feature lines," Comput. Graph., vol. 34, no. 5, pp. 482-498, 2010.

[17] K. Hirat, and T. Kato, "Query by visual example," Advances in Database Technology, EDBT'92. Springer Berlin Heidelberg, 1992, pp. 56-71.

[18] Y. Zhang, X. Qian, and X. Tan, "Sketch-based Image Retrieval Using Contour Segments ," in Proc. IEEE MMSP, 2015, pp.1-6.

[19] Y. Zhang, X. Qian, X. Tan, J. Han, Y. Tang, "Sketch-based Image Retrieval by Salient Contour Reinforcement," IEEE Trans. Multimedia, 2016, 10.1109/TMM.2016.2568138.

[20] Q. Yu, F. Liu, Y. Song, et al., "Sketch me that shoe," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[21] M. Eitz, K. Hildebrand, T. Boubekeur, et al., "A descriptor for large scale image retrieval based on sketched feature lines," in Proc. 6th Eurographics Symp. Sketch-Based Interfaces Model, 2009, pp. 29–36.

[22] M. Eitz, K. Hildebrand, T. Boubekeur, et al., "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," IEEE Trans. Vis. Comput. Graph., vol. 7, no. 11, pp. 1624–1636, Nov. 2011.

[23] T. Bui, J. Collomosse, "Scalable sketch-based image retrieval using color gradient features," In ICCV, 2015.

[24] X. Sun, C. Wang, C. Xu, et al., "Indexing billions of images for sketch-based retrieval," In ACM Multimedia, 2013.

[25] S. Parui, A. Mittal, "Similarity-invariant sketch-based image retrieval in large databases," In ECCV, pages 398–414. Springer, 2014.

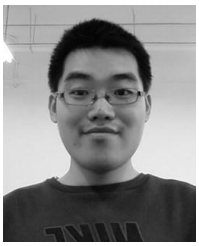
[26] T. Bui, L. Ribeiro, M. Ponti, et al, "Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network," Computer Vision and Image Understanding, 2017.

[27] P. Xu, Q. Yin, Y. Qi, et al, "Instance-Level Coupled Subspace Learning for Fine-Grained Sketch-Based Image Retrieval," European Conference on Computer Vision. Springer International Publishing, 2016: 19-34.

[28] P. Xu, Q. Yin, Y. Huang, et al., "Cross-modal Subspace Learning for Fine-grained Sketch-based Image Retrieval," arXiv preprint arXiv: 1705.09888, 2017.

[29] K. Li, K. Pang, Y. Song, et al, "Synergistic Instance-Level Subspace Alignment for Fine-Grained Sketch-Based Image Retrieval," IEEE Tran-

sactions on Image Processing, 2017, 26(12): 5908-5921.
[30] P. Sangkloy, N. Burnell, C. Ham, J. Hays, "The sketchy database: learning to retrieve badly drawn bunnies," ACM Transactions on Graphics (TOG), 2016, 35(4): 119.
[31] L. Liu, F., Shen, Y. Shen, et al, "Deep sketch hashing: Fast free-hand sketch-based image retrieval," In Proc. CVPR, 2017, (pp. 2862-2871).
[32] O. Seddati, S. Dupont, S. Mahmoudi, "Quadruplet Networks for Sketch-Based Image Retrieval," Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval. ACM, 2017: 184-191.
[33] D. Martin, C. Fowlkes, J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," IEEE TPAMI., vol. 26, no. 5, pp. 530-549, 2004.
[34] J. Canny, "A Computational Approach to Edge Detection," IEEE Computer Society, 1986.
[35] J. Lim, C. Zitnick, P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," In Proc. CVPR, 2013.
[36] T. Portenier, Q. Hu, P. Favaro, et al, "SmartSketcher: sketch-based image retrieval with dynamic semantic re-ranking," Proceedings of the Symposium on Sketch-Based Interfaces and Modeling. ACM, 2017.
[37] Y. Matsui, K. Ito, Y. Aramaki, et al., "Sketch-based manga retrieval using manga109 dataset," Multimedia Tools and Applications, 2017, 76(20): 21811-21838.
[38] L. Wang, X. Qian, Y. Zhang, et al., "Enhancing Sketch-Based Image Retrieval by CNN Semantic Re-ranking," IEEE Transactions on cybernetics, 2019, Online.
[39] Z. Liu, W. Zou, O. Meur, "Saliency tree: A novel saliency detection framework," IEEE Transactions on Image Processing, 2014, 23(5): 1937-1952.
[40] M. Cheng, N. Mitra, X. Huang, et al., "Global Contrast Based Salient Region Detection," IEEE TPAMI, 2015, 37(3): 569-582.
[41] A. Chalechale, G. Naghdy, and A. Mertins, "Edge image description using angular radial partitioning, " IEEE Proceedings-Vision, Image and Signal Processing, vol. 151(2), pp.93–101, April, 2004.
[42] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, 2004, 60(2): 91-110.
[43] A. Krizhevsky, I. Sutskever, G. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems. 2012: 1097-1105.
[44] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Eprint Arxiv, 2014.
[45] C. Szegedy, W. Liu, Y. Jia, et al, "Going deeper with convolutions," 2014:1-9.
[46] K. He, X. Zhang, S. Ren, et al, "Deep residual learning for image recognition," In Proc. Of IEEE Conf Comput Vis Pattern Recognit, pp.1–12
[47] B. Tu, L. Ribeiro, M. Ponti, et al., "Sketching out the details: Sketch-based image retrieval using convolutional neural networks with multi-stage regression," Computers & Graphics 71 (2018): 77-87.
[48] M. Eitz, J. Hays, M. Alexa, "How do humans sketch objects?," ACM Trans. Graph., 2012, 31(4): 44:1-44:10.
[49] H. Zhang, S. Liu, C. Zhang, et al., "Sketchnet: Sketch classification with web images," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1105-1113.
[50] Y. Jia, E. Shelhamer, J. Donahue, et al., "Caffe: Convolutional architecture for fast feature embedding," arXiv preprintarXiv: 1408.5093, 2014.
[51] http://dl.caffe.berkeleyvision.org/bvlc_googlenet.caffemodel
[52] B. Frey, D. Dueck, "Clustering by passing messages between data points," Science, 2007, 315(5814): 972-976.
[53] S. Yelamarthi, S. Reddy, A. Mishra, et al., "A zero-shot framework for sketch based image retrieval," European Conference on Computer Vision. 2018: 316-333.
[54] G. Wu, J. Han, Y. Guo, et al., "Unsupervised deep video hashing via balanced code for large-scale video retrieval," IEEE Transactions on Image Processing. 2018, 28(4): 1993-2007.
[55] G. Wu, J. Han, Z. Lin, et al., "Joint image-text hashing for fast large-scale cross-media retrieval using self-supervised deep learning," IEEE Transactions on Industrial Electronics. 2018, Online.
[56] Y. Wang, L. Zhu, X. Qian, et al., "Joint Hypergraph Learning for Tag-Based Image Retrieval", IEEE TIP. 2018, 27(9): 4413-4451.
[57] X. Qian, D. Lu, Y. Wang, et al., "Image Re-Ranking Based on Topic Diversity," IEEE TIP. 2017, 26(8): 3734-3747.
[58] Y. Wang, H. Yang, X. Qian, et al., "Position Focused Attention Network for Image-Text Matching", IJCAI. 2019, 3792-3798.
[59] C. Kang, L. Zhu, X. Qian, et al., "Geometry and Topology Preserving Hashing for SIFT Feature," IEEE Trans. Multimedia. 2019, 21(6): 1563-1576.
[60] D. Liu, X. Hua, M. Wang, et al., "Boost search relevance for tag-based social image retrieval", IEEE International Conference on Multimedia & Expo. IEEE Press, 2009.
[61] J. Wang, Y. Song, T. Leung, et al., "Learning Fine-Grained Image Similarity with Deep Ranking", Computer Vision & Pattern Recognition. 2014.
[62] http://dl.caffe.berkeleyvision.org/bvlc_alexnet.caffemodel
[63] http://www.robots.ox.ac.uk/~vgg/software/very_deep/caffe/VGG_ILSVRC_16_layers.caffemodel
[64] https://onedrive.live.com/?authkey=%21AAFW2-FVoxeVRck&id=4006CBB8476FF777%2117887&cid=4006CBB8476FF777



Luo Wang is currently working toward the Ph.D. degree with the SMILES Laboratory, Xi'an Jiaotong University, Xi'an, China.

His research interests include sketch-based image retrieval, image content understanding and deep learning.

Xueming Qian (M'09) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, China, in 2008.

From November 2011 to March 2014, he was an Associate Professor with Xi'an Jiaotong University, where he is currently a full Professor. He is also the Director of the SMILES Laboratory, X΄ıan Jiaotong University. He was a Visiting Scholarar with Microsoft Research Asia, Beijing, China, from August 2010 to March 2011. His research interests include social media big data mining and search.

Prof. Qian was the recipient of a Microsoft Fellowship in 2006 and Outstanding Doctoral Dissertations of Xi'an Jiaotong University and Shaanxi Province in 2010 and 2011, respectively.

Xingjun Zhang received the PhD degree in computer architecture from Xi'an Jiaotong University, China, in 2003. From 1999 to 2005, he was a lecturer and associate professor with the Department of Computer Science & Technology, Xi'an Jiaotong University.

From Feb. 2006 to Jan. 2009, he was a research fellow with the Department of Electronic Engineering, Aston University, United Kingdom. He was an associate professor during 2009-2013 with the Department of Computer Science & Engineering, Xi'an Jiaotong University, where he has been a full professor from 2014.

His interests include high performance computer architecture, high performance computing, big data storage system, and computer networks. He is a member of the IEEE.

Xingsong Hou received the Ph.D. degree from Xian Jiaotong University, Xian, China, in 2005. Now, he is a Professor with the School of Electronics and Information Engineering, Xian Jiaotong University. During October 2010-2011, he was a Visiting Scholar at Columbia University, New York, USA.

His research interests include video/image coding, wavelet analysis, sparse representation, sparse representation and compressive sensing, and radar signal processing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2959875, IEEE Transactions on Circuits and Systems for Video Technology

Fig. 4 The top-10 retrieval results on Sketchy Extension dataset for a query sketch 'cup'. The first row is the results of TripAlex method, the second row is the results of IRC method on TripAlex, the third row is the results of using BR method on TripAlex, and the fourth row is the results of using our re-ranking method on TripAlex. The fifth row is the results of GN Triplet method [30], the sixth row is the results of IRC method on GN Triplet, the seventh row is the results of using BR method on GN Triplet, and the eighth row is the results of using our re-ranking method on GN Triplet. The ninth row is the results of DHS method [31], the tenth row is the results of IRC method on DHS, the eleventh row is the results of using BR method on DHS, and the twelfth row is the results of using our re-ranking method on DHS. The thirteenth row is the results of SCMR method [47], the fourteenth row is the results of IRC method on SCMR, the fifteenth row is the results of using BR method on SCMR, and the sixteenth row is the results of using our re-ranking method on SCMR.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2959875, IEEE Transactions on Circuits and Systems for Video Technology

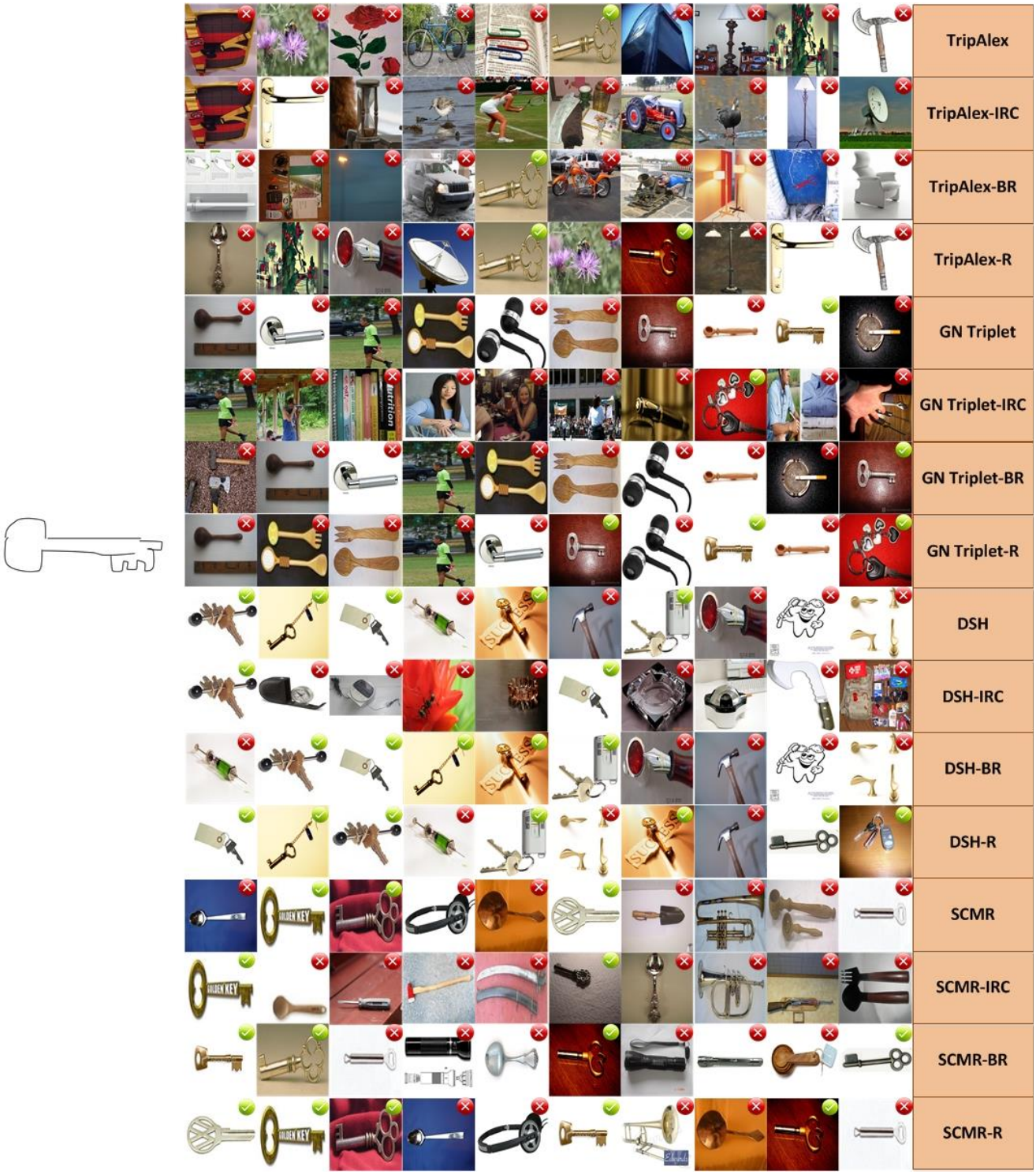


Fig. 5 The top-10 retrieval results on TU-Berlin Extension dataset for a query sketch 'key'. The first row is the results of TripAlex method, the second row is the results of IRC method on TripAlex, the third row is the results of using BR method on TripAlex, and the fourth row is the results of using our re-ranking method on TripAlex. The fifth row is the results of GN Triplet method [30], the sixth row is the results of IRC method on GN Triplet, the seventh row is the results of using BR method on GN Triplet, and the eighth row is the results of using our re-ranking method on GN Triplet. The ninth row is the results of DHS method [31], the tenth row is the results of IRC method on DHS, the eleventh row is the results of using BR method on DHS, and the twelfth row is the results of using our re-ranking method on DHS. The thirteenth row is the results of SCMR method [47], the fourteenth row is the results of IRC method on SCMR, the fifteenth row is the results of using BR method on SCMR, and the sixteenth row is the results of using our re-ranking method on SCMR.