

Joint Classification Loss and Histogram Loss for Sketch-Based Image Retrieval

Yongluan Yan, Xinggang Wang^(✉), Xin Yang, Xiang Bai, and Wenyu Liu

School of Electronic Information and Communications,
Huazhong University of Science and Technology,
1037 Luoyu Road, Wuhan 430074, Hubei Province, People's Republic of China
xgwang@hust.edu.cn

Abstract. We study the problem of content-based image retrieval using hand drawn sketches. The problem is very challenging since the low-level visual features of sketch and image have a large variance. Recent studies show that learning deep features that utilize high-level supervision is a feasible solution of this problem. We propose a new network structure with a joint loss by combining a simple classification loss with a robust histogram loss to learn better deep features for both sketch and image. The joint loss method has nearly no parameters to tune; it can not only learn the difference between image/sketch samples from different semantic class but also capture the fine-grained similarity between image/sketch samples in the same semantic class. In the experiments, we show the proposed method obtains excellent performance in real-time on the standard sketch-based image retrieval benchmark.

Keywords: Sketch · Histogram loss · Image retrieval · Deep learning

1 Introduction

Sketch-based image retrieval (SBIR) is an important problem that is attractive in both computer vision and computer graphics. In computer vision, sketch and image have different low-level visual cues; finding visual representation or designing a computer vision system to match sketch and image is a meaningful research topic. In computer graphics, sketch-based image retrieval has many fancy applications, such as MindFinder¹, shoe search, furniture design [1]. Thus, it is important to find a robust method to solve this problem.

Before the era of deep learning, there are two popular methods for SBIR. The first one is to extract sketch/edge from the image, and then perform sketch to image retrieval by shape matching methods. The second one is to extract local image descriptors and encode the image descriptors to generate a vector representation of both image and sketch. The local image descriptors can be SIFT [2] or HOG [3]. But these methods are not robust enough since the variation measured by the low-level visual features between image and sketch is so large.

¹ <http://research.microsoft.com/en-us/projects/mindfinder>.

Deep learning is more robust for this problem because it produces high-level semantic features. But the conventional deep features are not suitable for the sketch-based image retrieval task, such as the deep feature of AlexNet [4] trained in the ImageNet dataset [5]. Deep features for SBIR are learned from sketch and image data with instance-level or category-level annotations. Wang et al. firstly proposed a sketch-image dataset that contains category-level annotations in [6]; then Yu et al. proposed a sketch-image dataset with instance-level annotations in [1]. Using these annotations, the deep network for SBIR is trained using the batch contains both sketches and images; classification loss and triplet loss are utilized by [1,6] respectively. In this paper, we develop a new deep network structure that takes the advantages of both [1,6]; the network is supervised by both classification loss and embedding loss; furthermore, we adopt a more advanced embedding loss named histogram loss - it has nearly no parameters to tune and achieves excellent performance. The histogram loss is recently proposed in [7]; when training the network, histogram loss computes two similarity distributions for positive similarities and negative similarities respectively; the positive similarity means the similarity between two training samples from the same class, and the negative similarity means the similarity between two training samples from different classes; in the end, the histogram loss minimizes the intersection between the positive similarity distribution and the negative similarity distribution.

In the experiments, we validate the proposed method on the standard SBIR benchmark, i.e., M.Eitz-SBIR dataset [8]. The experimental results show that the joint loss function outperforms each of the individual loss, and the proposed deep feature obtains the state-of-the-art performance on this benchmark.

2 Related Work

Since there are more and more tablets and smart phones with pen as input, it is much easier for people to show his idea using sketch. Recognizing sketch is becoming more important in many AI applications. Previous SBIR methods can be divided into three folds: (1) SBIR using shape matching methods aims on matching the query sketch to the edge map of natural image, such as [9–13]. (2) SBIR using image descriptors, such as SIFT [14] and the dense stroke feature [15], and typically a bag of word model is adopted to aggregate the image descriptors into a compact representation for both sketch and image. (3) SBIR using deep learning features, which use supervised deep convolutional neural networks to learn representations for sketch and image, such as [6,16–18]. In detail, the contrastive loss [16,17] is used for sketch to 3D shape retrieval and sketch to image retrieval in [18,19] respectively. In [6], classification loss is used to learn deep representation, and it needs a balanced number of training data for each category. In [1], triplet loss is used since for each instance there is a limited number of training samples. Our work includes both classification loss and histogram loss [7] which is suitable for a small number of training samples per category.

Learning deep embeddings is widely used in many visual applications beyond of SBIR, for example, face recognition, person re-identification, fine-grained recognition, online product search etc. The histogram loss method [7] adopted has been proven to be more powerful than triplet loss [20,21] and the lifted structure similarity softmax (LSSS) loss [22].

3 Methodology

In this section, we give detailed descriptions of the proposed method for SBIR. Inspired by the histogram loss method in [7], our method estimates two distributions of similarities for a pair of positive and negative samples, then minimizes the probability of the case that the similarity score in a positive pair is lower than that in a random negative pair. We follow the trend of recent work which utilizes the CNN model combines it with the classification loss and the histogram loss to learn a cross-domain embedded feature [1,6,18,19,23].

3.1 Network Architecture

Given a query sketch s and a set of candidate nature images $\{p_j\}_{j=1}^M \in P$, SBIR focuses on computing the similarity between s and p . We use deep CNN model as feature extractor for both s and p . Without loss of generality, we denote the CNN feature extractor as $f_\theta(\cdot)$.

Due to the semantic gap between nature images and sketches, it is difficult to train a CNN model which works well in both sketch and image domains. In recent work, [18,23] focused on using a network with heterogeneous branches for the two domains, whereas [1,19] trained a network with multiple identity branches for free-hand sketches and edge maps extracted from nature images by the structural edge detector [24]. Unlike the previous works, we propose to learn deeply embedded features that represent both nature images and sketches well.

TU-Berlin dataset [14] is a widely-used sketch dataset containing 20,000 sketches, and HUST SI dataset [6] is composed of 31,824 nature images from the web. We extract edge maps from these nature images by using the edge detector in [24]. Then we obtain a mixed dataset with 20,000 sketches, 31,824 nature images, and 31,824 edge maps from nature images to train our model.

Deep models have the ability to learn high-level descriptions for input data. Here AlexNet [4], which is trained in the ImageNet dataset [5] consisting of 1,000 common object images, is adopted as our pre-trained model. Different from [6], we not only use the traditional ‘softmax’ classification loss but also adopt the histogram loss that encourages our model to be more sensitive to fine-grained sketch-to-image similarities. The framework is illustrated in Fig. 1.

3.2 Classification Loss

We define a batch of examples as $X = \{x_1, x_2, \dots, x_n\}$ and a deep convolutional network as $f_\theta(\cdot)$, where θ represents learnable parameters of the network. As our

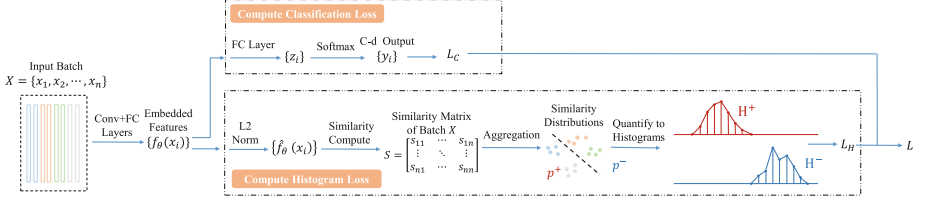


Fig. 1. The proposed deep convolutional network supervised by the classification loss and the histogram loss.

training dataset is composed of C common categories, the embedded features $\{f_\theta(x_i)\}$ can be classified by a fully-connected layer (we can regard it as a classifier) with C neurons. The output of the fully-connected layer z_i is fed into a C -way softmax layer which produces a distribution y_i over the C class in Eq. (2). The classification loss L_C over a batch of n examples can be calculated as in Eq. (1), where t_i is the class label of i th example.

$$L_c(T, Y) = - \sum_{i=1}^n L_c(t_i, y_i) = - \sum_{i=1}^n \sum_{c=1}^C t_{ic} \log(y_{ic}) \quad (1)$$

$$y_{ic} = \text{softmax}(z_{ic}) = \frac{e^{z_{ij}}}{\sum_{t=1}^C e^{z_{it}}} \quad (2)$$

3.3 Histogram Loss

Histogram loss is inspired by the Quadruplet-based loss [25, 26] which computes similarities/distances of positive pairs and negative pairs. However, the potential problem of the quadruplet-based loss is that there are an even larger number of all quadruplets than of all triplets for large-scale training dataset, leading to rare usage of this loss. As a consequence, [7] proposed a histogram loss which is regarded as an efficient quadruplet-based loss and almost in a parameter-free manner.

We use $f_\theta(\cdot)$ to convert the input samples $X = \{x_1, x_2, \dots, x_n\}$ to the embedded features $\{f_\theta(x_i)\}$. After performing L_2 -normalization, the embedded features can be represented as $\{\hat{f}_\theta(x_i)\}$. We match elements to each other to form a dense similarity matrix $S = \{s_{ij} = \langle \hat{f}_\theta(x_i), \hat{f}_\theta(x_j) \rangle\}$, which shows how similar they are. And the label $m_{ij} \in \{+1, -1\}$ indicates whether x_i and x_j is similar (from the same class) or not. Two similarity sets $S^+ = \{s_{ij} | m_{ij} = +1\}$ and $S^- = \{s_{ij} | m_{ij} = -1\}$ for positive and negative pairs are aggregated to estimate the probability distributions p^+ and p^- , respectively. In addition, the elements of two sets are bounded to $[-1; +1]$ for the sake of task simplification. By this means, we can acquire two distributions and convert them to R -dimensional histograms H^+ and H^- with uniformly spaced bins (the step of bins Δ is $\frac{2}{R-1}$).

The r th bin h_r^+ of the histogram H^+ is computed as in Eq. (3).

$$h_r^+ = \frac{1}{|S^+|} \sum_{(i,j):m_{ij}=+1} \delta_{i,j,r} \quad (3)$$

where (i, j) is a positive pair in the batch and $\delta_{i,j,r}$ can be assigned as in Eq. (4).

$$\delta_{i,j,r} = \begin{cases} (s_{ij} - t_{r-1})/\Delta, & \text{if } s_{ij} \in [t_{r-1}; t_r], \\ (t_{r+1} - s_{ij})/\Delta, & \text{if } s_{ij} \in [t_r; t_{r+1}], \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Meanwhile, H^- is estimated in the same way.

Consequently, we employ two distributions p^+ and p^- to calculate the probability of the case that the similarity score in a random negative pair is higher than that in a random positive pair, and it can be represented as in Eq. (5).

$$L_h(X) = \sum_{r=1}^R (h_r^- \sum_{q=1}^r h_q^+) \quad (5)$$

where L_h is our histogram loss for the batch X .

3.4 Optimization

We adopt the joint supervision of the classification loss L_c and the histogram loss L_h for learning deep features. The formulation is given in Eq. (6).

$$L = L_c + L_h \quad (6)$$

In order to optimize this network, it is crucial to update weights of this proposed network. Indeed, it is easy to do the back-propagation of classification loss. Here, the gradients of the histogram loss L_h with respect to h_r^+ and h_r^- are computed in Eq. (7):

$$\begin{aligned} \frac{\partial L_h}{\partial h_r^+} &= \sum_{q=1}^r h_q^+ \\ \frac{\partial L_h}{\partial h_r^-} &= \sum_{q=1}^r h_q^- \end{aligned} \quad (7)$$

Furthermore, the back-propagation gradients for similarities s_{ij} can be derived as shown in Eq. (8).

$$\frac{\partial h_r^+}{\partial s_{ij}} = \begin{cases} \frac{+1}{\Delta|S^+|}, & \text{if } s_{ij} \in [t_{r-1}; t_r], \\ \frac{-1}{\Delta|S^-|}, & \text{if } s_{ij} \in [t_r; t_{r+1}], \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

For $\frac{\partial h_r^-}{\partial s_{ij}}$, it is computed in a similar way. Finally, given $\frac{\partial s_{ij}}{\partial \hat{f}_\theta(x_i)} = \hat{f}_\theta(x_j)$ and $\frac{\partial s_{ij}}{\partial \hat{f}_\theta(x_j)} = \hat{f}_\theta(x_i)$, the loss can be back-propagated to the individual samples, and then further into the deep embedding network.

4 Experiment

In this section, we perform experiments to test the proposed method on the task of SBIR and compare it to the previous methods.

4.1 Dataset

TU-Berlin dataset [14] has 20,000 non-expert sketches, divided into 250 common object categories, e.g., bicycle, car. In each category, there are 80 sketches of 1111×1111 size are drawn by the human.

HUST SI dataset [6] contains 31,824 nature images of 250 categories, corresponding to TU-Berlin sketch classes. For each category, there are at least 80 nature images in various sizes.

M.Eitz-SBIR [8] dataset is the retrieval set. This dataset contains 31 sketches and each sketch is associated with 40 nature images. We query a sketch in its 40 related nature images and obtain a ranking list of the nature images. And we compute the Kendall's rank correlation coefficient [8] to evaluate the performance of the proposed method. This evaluation criteria will be briefly reviewed in Sect. 4.3.

In order to learn deep features suitable for sketches and nature images, we combine TU-Berlin, HUST SI, and edge maps of HUST SI as mixed training data. M.Eitz-SBIR dataset is served as the retrieval dataset.

4.2 Experimental Settings

Our deep learning model is based on a widely used pre-trained model AlexNet [4], which is designed for 1000 common object categories of the ImageNet classification task [5]. AlexNet contains 8 layers: the first five ones are convolutional layers and the remaining layers are fully-connected layers. Each convolutional layer is followed by a ReLU layer, and some of them are also followed by a local normalization layer and max-pooling layer. The 6th layer and the 7th layer both includes 4096 neurons, and the 8th layer outputs 1000 nodes for 1000-category classification. As is referred in Sect. 3.3, only the first 7 layers of AlexNet are employed in our model to extract deep features, and we use these features to compute the histogram loss. With deep features as input, a new fully-connected layer with 250 neurons outputs the class distributions. And the classification loss is calculated based on these class distributions. No matter it is sketch or image, it is resized into 227×227 to extract its deep feature.

We use the open source Keras [27] toolbox to train our models. Weights of the last fully-connected layer are initialized using a glorot-uniform distribution [28]. Biases are all initialized to be 0. The initial learning rate is set to 0.001, divided it by 10 at the 15th epoch and the 30th epoch. And mini-batch size is assigned to 256. The size of histogram bins is set to 100. Meanwhile, the weight decay is 0.0005.

Table 1. Comparison with the results (using Kendall’s rank correlation coefficient) of different data pre-process strategies on M.Eitz-SBIR dataset. The best result is in bold.

Training data	Without data-aug	With data-aug
Sketch	0.373	0.347
Sketch + image	0.434	0.471
Sketch + image + edge-map	0.479	0.497

4.3 Evaluation

We perform our retrieval experiments on the M.Eitz-SBIR dataset with the Kendall’s rank correlation coefficient [8] as the evaluation criteria. As is referred in [8], the Kendall’s rank correlation coefficient τ is proposed to determine how similar two ranking lists are. So we compare the ranking list of our proposed method to the user ranking which is regarded as “ground-truth”. τ can take values in the range $[-1, 1]$, with -1 indicating a reversed list, 0 indicating two ranking lists are independent, and 1 indicating two lists are in a same order.

A SBIR system may produce the same score for two images, thus possibly producing tied pairs. Therefore, a variation of the Kendall’s rank correlation coefficient is used. This variation is denoted as τ_b and defined as Eq. (9):

$$\tau_b = \frac{n_c - n_d}{[(N - U)(N - V)]^{\frac{1}{2}}} \quad (9)$$

where n_c and n_d denote the number of concordant and discordant pairs, respectively. $N = n(n - 1)/2$ means the number of possible pairs in a set of n distinct elements, $U = \frac{1}{2} \sum_{i=1}^t t_i(t_i - 1)/2$ means the number of tied pairs in the baseline ranking list, and $V = \frac{1}{2} \sum_{i=1}^u u_i(u_i - 1)/2$ represents the number of tied pairs in the ranking list of our proposed method.

4.4 Results

Training data pre-process strategies. We employ different strategies of data pre-process to train our model. Firstly, we discuss on the influence of different mixing strategies of training data. (1) *Only using Sketches*: 20,000 sketches in TU Berlin are used to train our model directly. (2) *Mixing the sketches and nature images*: We combine TU Berlin with HUST SI dataset as the training data. (3) *Mixing the sketches, nature images, and edge maps*: Following [6], edge maps of the HUST SI dataset are added to training data. We extract edge maps by the structural edge detector and binary them with the threshold of 0.9.

Data augmentation (data-aug) is widely used to boost the performance of deep learning system. Here, we compare the experimental results with data augmentation to those without data augmentation. The data augmentation method applied in this article is briefly introduced as follows: For training data, we resize a image/sketch to the size 256×256 , then randomly crop a 227×227

Table 2. Comparison with the results of different size of histogram bins on M.Eitz-SBIR dataset.

Size of histogram bins	Correlation coefficient
100	0.497
50	0.494
25	0.492
10	0.483

patch from the resized image and randomly horizontally flip the patch during the network training. To obtain fast retrieval speed, for retrieval images, no data-augmentation is done.

As is shown in Table 1, performing data augmentation for mixed data of TU Berlin, HUST SI and edge maps of HUST SI achieves the best performance. Due to the similar appearance between sketches and edge maps from the same category, edge maps can bridge the common characteristic of them. In contrast, if a model is trained without edge maps, it is hard to fill the semantic gap and learn the same feature extractor for sketches and images. When only applying sketches as the input, the features extracted by the model are only suitable for sketches but not suitable for nature images. So mixing sketches, nature images, and edge maps are superior to other two strategies of using training data.

Parameter study of the histogram loss. As is referred in Sect. 3.3, the size of histogram bins is the only tunable parameter of the histogram loss. It determines whether histograms are coarse or fine-grained. In order to estimate similarity distributions p^+ and p^- , we transform a training batch of examples to a dense matrix of pairwise similarities. Then the distributions are quantized to fixed-length vectors. If the batch size is too small, the similarity distributions may not model the real distributions. Only keeping large batch size can provide the data independence.

We perform experiments when the size of histogram bins is equal to 100, 50, 25, 10 ($\Delta = 0.01, 0.02, 0.04, 0.1$). And the batch size is fixed to 128. Table 2 illustrates that we get similar results if the number of histogram bins is set to 100, 50 and 25. However, if the size of histogram bins is too small, the correlation coefficient value will drop. Therefore, we set the size of histogram bins as 100 and the batch size as 128.

Comparison with the state-of-the-art methods. We make a comparison with previous work in Table 3. As is mentioned in Sect. 4.4, it discusses the data pre-process of training data and the size of histogram bins. Following our best setting, our proposed method achieves the state-of-the-art result, which outperforms SHOG [8], Key-shape [29], cross-domain method [6], etc. Meanwhile, we conduct comparison experiments that the model trains only with classification loss or histogram loss.

Table 3. Comparison with results of different method on M.Eitz-SBIR dataset. In the table, hist, classification and joint respectively represents the network trained only with histogram loss, only with histogram loss and with the joint loss.

Method	Correlation coefficient
SHOG [8]	0.277
Words-of-Interest [30]	0.313
Key-shape [29]	0.289
Min-Hash [31]	0.336
Content-aware [32]	0.352
Cross-domain [6]	0.477
Ours (hist)	0.451
Ours (classification)	0.441
Ours (joint)	0.497

Table 3 explains that our model with the joint loss is superior to that with a single classification or histogram loss. The advantage of joint loss is that it learns the category semantics and fine-grained details at the same time. Moreover, some samples of the SBIR results are shown in Fig. 2. We list the top 10 retrieval results of our proposed method for 8 query sketches. The first column denotes the Kendall’s rank correlation coefficient score τ_b of each query sketch which is given in second column. If τ_b is closer to 1, it means that this retrieval result is in better performance. Conversely, it represents a bad retrieval result. In Fig. 2, we can find that the correlation coefficient score of second row and fifth row are lower than other rows. In second row, query sketch is an image of two people and should retrieval group photos. However, it is easily confused by images that contain many people, like 2th and 3th retrieval results. Query sketch in fifth row is a building with some windows, but we retrieval several images that contain a sign or a door actually. It indicates that our proposed method can not distinguish buildings from other rectangular objects. However, it still makes sense, because buildings have rectangular shape inside.

4.5 Running Cost

All of our experiments run on a PC with a NVIDIA GeForce GTX 1080 GPU (8 GB) and 32 GB RAM. It takes about 7 h for training (nearly 25,000 iterations, 45 epochs). During testing, it will take about 17 ms to preform one retrieval, including the time of feature extraction of query sketch and feature comparison. It meets the requirement of real-time application. The source code for reproducing results is available upon acceptance.

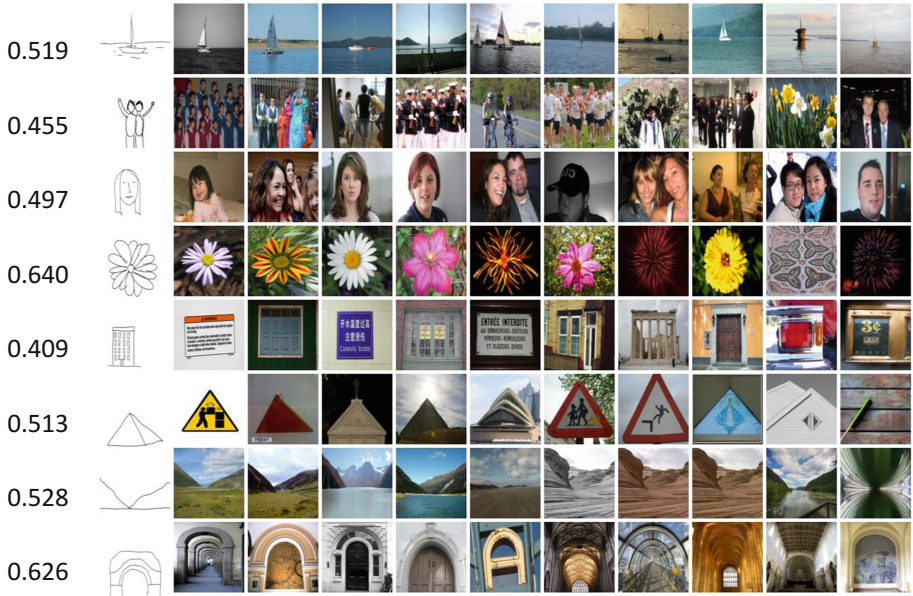


Fig. 2. The top 10 retrieval results of the proposed method. Each row shows a retrieval result. The first column and second column means the correlation coefficient score and query image respectively.

5 Conclusion and Future Work

For the problem of SBIR, the proposed deep network jointly trained using classification loss and histogram loss is able to learn superior deep features. The joint loss is better than the single loss; it could be applied in tasks which need to learn deep embeddings. In addition, we will study how to design network structure that is more suitable for sketch recognition and generating an image from a sketch using generative adversarial networks in the future.

References

1. 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)
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)
4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)

5. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015)
6. Wang, X., Duan, X., Bai, X.: Deep sketch feature for cross-domain image retrieval. *Neurocomputing* **207**, 387–397 (2016)
7. Ustinova, E., Lempitsky, V.: Learning deep embeddings with histogram loss. In: *Advances in Neural Information Processing Systems*, pp. 4170–4178 (2016)
8. Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: Sketch-based image retrieval: benchmark and bag-of-features descriptors. *IEEE Trans. Vis. Comput. Graph.* **17**(11), 1624–1636 (2011)
9. Olsen, L., Samavati, F.F., Sousa, M.C., Jorge, J.A.: Sketch-based modeling: a survey. *Comput. Graph.* **33**(1), 85–103 (2009)
10. Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: A descriptor for large scale image retrieval based on sketched feature lines. In: *SBIM*, pp. 29–36 (2009)
11. Cao, Y., Wang, C., Zhang, L., Zhang, L.: Edgel index for large-scale sketch-based image search. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 761–768 (2013)
12. Parui, S., Mittal, A.: Similarity-invariant sketch-based image retrieval in large databases. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014*. LNCS, vol. 8694, pp. 398–414. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_26
13. You, X., Peng, Q., Yuan, Y., Cheung, Y.-M., Lei, J.: Segmentation of retinal blood vessels using the radial projection and semi-supervised approach. *Pattern Recogn.* **44**(10), 2314–2324 (2011)
14. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? *ACM Trans. Graph.* **31**(4), 44 (2012)
15. Ma, C., Yang, X., Zhang, C., Ruan, X., Yang, M.-H., Omron Coporation: Sketch retrieval via dense stroke features. In: *British Machine Vision Conference (BMVC)*, vol. 2, p. 3 (2013)
16. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1735–1742. IEEE (2006)
17. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 1, pp. 539–546. IEEE (2005)
18. Wang, F., Kang, L., Li, Y.: Sketch-based 3D shape retrieval using convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1875–1883 (2015)
19. Qi, Y., Song, Y.-Z., Zhang, H., Liu, J.: Sketch-based image retrieval via Siamese convolutional neural network. In: *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 2460–2464. IEEE (2016)
20. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1386–1393 (2014)
21. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: Feragen, A., Pelillo, M., Loog, M. (eds.) *SIMBAD 2015*. LNCS, vol. 9370, pp. 84–92. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24261-3_7

22. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4004–4012 (2016)
23. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. *ACM Trans. Graph. (TOG)* **35**(4), 119 (2016)
24. Dollár, P., Zitnick, C.L.: Fast edge detection using structured forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(8), 1558–1570 (2015)
25. Law, M.T., Thome, N., Cord, M.: Quadruplet-wise image similarity learning. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 249–256 (2013)
26. Zheng, W.-S., Gong, S., Xiang, T.: Reidentification by relative distance comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(3), 653–668 (2013)
27. Chollet, F.: Keras (2015). <https://github.com/fchollet/keras>
28. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS, vol. 9, pp. 249–256 (2010)
29. Saavedra, J.M., Bustos, B.: Sketch-based image retrieval using keyshapes. *Multimedia Tools Appl.* **73**(3), 2033–2062 (2014)
30. Luo, X., Guo, W.-J., Liu, Y.-J., Ma, C.-X., Song, D.: A words-of-interest model of sketch representation for image retrieval
31. Bozas, K., Izquierdo, E.: Large scale sketch based image retrieval using patch hashing. In: Bebis, G., et al. (eds.) ISVC 2012. LNCS, vol. 7431, pp. 210–219. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33179-4_21
32. Liang, S., Zhao, L., Wei, Y., Jia, J.: Sketch-based retrieval using content-aware hashing. In: Ooi, W.T., Snoek, C.G.M., Tan, H.K., Ho, C.-K., Huet, B., Ngo, C.-W. (eds.) PCM 2014. LNCS, vol. 8879, pp. 133–142. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13168-9_14