Journal of Electronic Imaging

JElectronicImaging.org

Making better use of edges for sketch generation

Xingyuan Zhang
Yaping Huang
Qi Zou
Qingji Guan
Junbo Liu

Xingyuan Zhang, Yaping Huang, Qi Zou, Qingji Guan, Junbo Liu, "Making better use of edges for sketch generation," *J. Electron. Imaging* **27**(6), 063006 (2018), doi: 10.1117/1.JEI.27.6.063006.

Journal of Electronic Imaging 27(6), 063006 (Nov/Dec 2018)

Making better use of edges for sketch generation

Xingyuan Zhang, Yaping Huang,* Qi Zou, Qingji Guan, and Junbo Liu
Beijing Jiaotong University, Beijing Key Lab of Traffic Data Analysis and Mining, Computer Science, Haidian District, Beijing, China

Abstract. Sketching has become fashionable with the increasing availability of touch-screens on portable devices. It is typically used for rendering the visual world, automatic sketch style recognition and abstraction, sketch-based image retrieval (SBIR), and sketch-based perceptual grouping. How to automatically generate a sketch from a real image remains an open question. We propose a convolutional neural network-based model, named SG-Net, to generate sketches from natural images. SG-Net is trained to learn the relationship between images and sketches and thus makes full use of edge information to generate a rough sketch. Then, mathematical morphology is further utilized as a postprocess to eliminate the redundant artifacts in the generated sketches. In addition, in order to increase the diversity of generated sketches, we introduce thin plate splines to generate more sketches with different styles. We evaluate the proposed method of sketch generation both quantitatively and qualitatively on the challenging dataset. Our approach achieves superior performance to the established methods. Moreover, we conduct extensive experiments on the SBIR task. The experimental results on the Flickr15k dataset demonstrate that our proposed method leverages the retrieval performance compared with the state-of-the-art methods. [DOI: 10.1117/1.JEI.27.6.063006]

Keywords: freehand sketch; SG-Net; sketch generation; sketch-based image retrieval; thin plate spline.

Paper 180441 received May 17, 2018; accepted for publication Oct. 26, 2018; published online Nov. 19, 2018.

1 Introduction

Sketching comes naturally to humans and is supposed to be essential to communicate with each other. Because of the recent prevalence of touch-screen devices in daily life, e.g., phones, tablets, and smart watches, we can describe what we think by sketching on these devices. Therefore, in recent years, an exploratory study investigating how to generate high-quality sketches has attracted a great deal of interest. In addition, it has a wide application area, such as sketch-based image retrieval (SBIR),[1–6] sketch recognition,[7,8] sketch-based perceptual grouping,[9,10] sketching style recognition,[11] sketching style abstraction,[12,13] and others.

In terms of application examples, it remains surprising that very few people have considered the lack of training data. That is to say, despite the fact that computers have achieved human-level in the recognition of freehand sketches,[7,14,15] the ability to synthesize sketches has not been fully explored. At present, previous studies mainly focus on a special domain: human faces. Although existing methods are successful for synthesizing a face sketch, there exist widespread and important assumptions that we do not render them directly applicable to more categories. The difference regarding the face sketch is that it exhibits well-known, stable structure: (i) traditional handcrafted feature descriptors of faces are sufficient because they are essentially full of structural and appearance variations compared with natural images; (ii) the face sketch is strict in data alignment, and we can obtain new samples by simple patch replacement.

A large number of photographs can be searched to train the model for each visual category on media-sharing or video-sharing sites such as PASCAL VOC,[16] Flickr,[17] and others. As few large-scale photo-sketch datasets are open to the public, this makes the study more difficult in the sketch domain. In particular, our choice of making full use of midlevel and high-level features for sketch generation is specific: (i) sketches exhibit a woeful lack of essential visual cues (images have color and texture, but sketches only have black and white lines); (ii) sketches have an abstract depiction, e.g., stick figures represent human; and (iii) compared with rendered drawings, sketches are one of the simplest forms to represent human visual impressions, and as a result, many testing theories are found in human visual cognition.[1]

In this paper, we address the sketch generation problem by developing a convolutional neural network-based sketch generation model, SG-Net, which aims to convert a real image into a human-drawing-like sketch. We treat the task of sketch generation as a feature learning process. Essentially, our underlying hypothesis is that convolutional neural network is able to find some distinctive information from a clustered background, and thus there remains only important information corresponding to human-drawing-like sketches. In this case, the task of SBIR benefits from the sketch generation technology. The main reason is that traditional methods use the cross-domain (photo-sketch) model to perform SBIR, whereas our method aims to transform the problem of cross-domain into the inner-domain (sketch-sketch). The SBIR system uses sketches as a search query and returns the corresponding images based on similarity distance computation during retrieval. To sum up, if we transform images into freehand sketches, the retrieval accuracy would be higher as expected. Additionally, our model undergoes nonrigid deformation and affine transformation, whereas many traditional methods[18] obtain training images from well-chosen datasets with uniform background.

Sketch generation needs to consider both low-level and high-level features, so we consider using multiscale processing. In previous methods, it has been verified that multiscale inputs[19,20] work very well in the image domain. Thus, we use a multiscale convolutional neural network to generate

*Address all correspondence to: Yaping Huang, E-mail: yphuang@bjtu.edu.cn

1017-9909/2018/$25.00

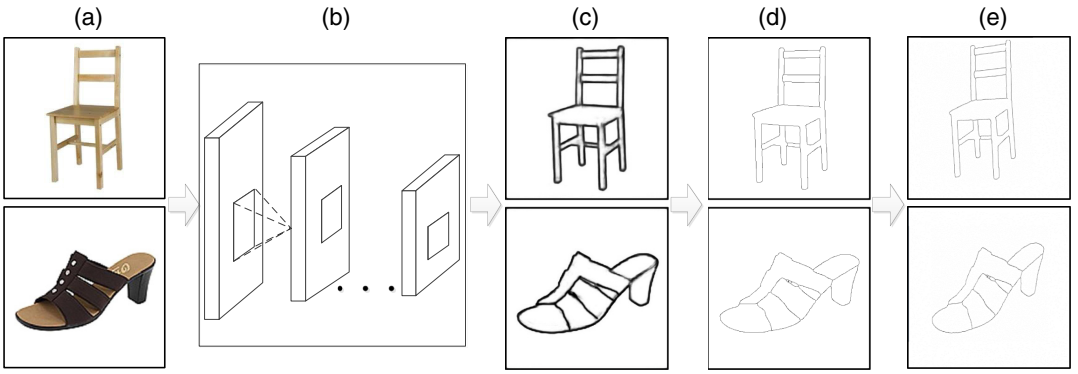

Fig. 1 (a–d) The framework of SG-Net with convolutional neural network and (e) the visualization of sketch deformation. (a) Input image: the images are used as inputs. (b) Training model: the model is pretrained based on our datasets. (c) Sketch generation: rough sketches are generated from the pretrained model in (b). (d) Refinement sketch: the refinement by mathematical morphology generates human-drawing-like sketches. (e) Sketch metamorphosis: we use TPS to perform metamorphosis and thus generate different sketch styles.

sketches, which makes full use of useful information. The entire process of sketch generation is shown in Fig. 1.

Our method aims to obtain a concise, simple, and hand-drawn-like sketch. Figure 1 shows the framework of SG-Net. The main part of SG-Net is (a)–(d), and (e) is the post-processing to improve the performance of SBIR. The first step of the flowchart (a, b) aims to generate rough sketches by RoughSketchNet. The model introduces multiscale and multilevel learning of deep image features via auxiliary cost functions at each convolutional layer to extract abundant information for a rough sketch. Then, as shown in (c), there are some redundant pixels in the sketch, which makes rough sketches very thick. We introduce mathematical morphology as sketch refinement and the result is shown in (d). Different users may have different drawing skills, and we may get a variety of styles for the same object. Therefore, free-form deformation of final sketches by thin plate splines (TPS) is conducted to adapt the condition of different sketch styles and the result is shown in (e).

The main contributions are described as follows:

i. We propose a new model named SG-Net to make full use of image features, which uses a convolutional neural network and mathematical morphology technology to obtain a human-drawing-like sketch;
ii. Compared with traditional methods that generate a sketch from a set of weakly labeled images with contour-based cues, our method generates a sketch from a single image;
iii. We first introduce TPS transformation to obtain a variety of sketch styles for the same object. As demonstrated in the experiment, sketch deformation can improve the performance of SBIR;
iv. We contribute a new pixel-to-pixel dataset consisting of 1400 sketch-photo image pairs from two object categories, e.g., shoe and chair.

The rest of this paper is organized as follows. Section 2 describes the related work. In Sec. 3, the arithmetic of the system and implementations of the model are described. Experimental results and comparisons with other methods are presented in Secs. 4 and 5. Finally, we conclude the paper in Sec. 6.

2 Related Work

In this section, we first make a survey of recent works on sketch generation. In fact, in order to generate the sketch from one image, we usually extract the edges first, and then further employ high-level semantic information to generate the corresponding sketch. Therefore, we summarize the application of deep learning on the edge detection task. In addition, in order to evaluate the effectiveness of our model for sketch generation, we transform images into corresponding sketches and perform SBIR. Therefore, we also summary the established works on SBIR.

2.1 *Sketch Generation*

Sketch generation from images has been much discussed in previous decades. Previous approaches can be mainly divided into two categories. Some models try to extract sketches from gradient intensity maps computed from natural images, which focuses on considering style abstraction and thus produces sketches that look similar to original photos.[18,21,22] The gradient intensity map is generally obtained by applying directional derivative operators to gray-scale images smoothed by a Gaussian kernel. On the contrary, some other works focus on sketch generation from multiple images.[10,23] They usually use deformable strokes or perceptual grouping of edges from multiple images to synthesis the final sketch. So their works focus on how to fuse the collection of sketches with similar poses.

For the task of sketch generation from a single image, Alelaez et al.[18] put forward a method using edge detection to obtain a hierarchical segmentation region, which could produce the corresponding edge as the final sketch. Zhu et al.[21] made full use of the principle that a smooth outline should have one good dimensional topology, which could produce a desirable sketch with the smooth contour. Guo et al.[24] combined two generative models learned from images. They use a sparse coding model and Markov random field to represent the geometry and texture information of an object. The model searches the most important outline to represent the sketch. Zhang et al.[25] used Markov absorption to detect visual saliency and Gabor filter to refine the salient region map. The descriptor of the Sobel operator results in discontinuous lines in the final result. Inspired by GAN,[26]

the related work is given by Kim et al.[27] They proposed generative adversarial networks (DiscoGAN), which learns to discover relations between images and sketches. The experimental results show the sketches which are generated from colored images of handbags. Although a set of natural color images are used to train the model, the generated sketches present many redundant edges. Because previous works have an assumption that object contours are complex and meaningful, they cannot be applied in our task.

For the task of sketch generation from multiply images, Qi et al.[10] proposed a perceptual grouping framework that generates sketches from organized image edges. In particular, they used RankSVM for the first time and combined multiple Gestalt principles as a cue for edge grouping. Following the idea of Ref. 10, Qi et al.[28] extracted information from a large of human sketches for sketch generation. Li et al.[23] presented a generative model by deformable stroke model (DSM) learning, which synthesizes a freehand sketch automatically by the collection of sketches with similar poses. This work focuses on obtaining the DSM by semantic parts with perceptual grouping. Once DSM is obtained, they could synthesize sketches for given images. The final result is full of short lines, which cannot match well with human visual impression. Marvaniya et al.[29] proposed to automatically extract a sketch from a set of weakly labeled images with the common object structure, which is hard to merge the repeatable contour fragments perfectly. Thus, it tends to generate sketches with discontinuous lines. Authors of Ref. 30 decomposed training image–sketch pairs into various patches and trained the model for path-level mapping. As a result, the strategies for the image are often sufficient to generate sketches. Moving onto hand-drawn-like sketches, Ref. 12 directly made use of strokes collected from a portrait sketch dataset which are drawn by professional artists and trained the model to reflect style and abstraction of different artists.

Although previous works laid a foundation for hand-drawn-like sketches, they presented two major challenges. On the one hand, the majority of sketch generation methods look for a repetitive and smooth sketch to represent the contour of the same object. However, as we know, edges and contours are not equivalent to sketches. On the other hand, previous result for sketch generation contains a large amount of redundant information.

2.2 *Edge Detection*

Inspired by psychological discovery, extracting edges have long been regarded as the key to solving vision problems, such as object detection[31] and object proposal generation.[32,33] As a result, these applications promote development of different methodologies, from simple gradient-driven Canny edges[34] to more sophisticated methods[35,36] that exploit multiple features. The result they obtained evidently demonstrates that deep learning-based features[37–39] have stronger capability for high-level image representation compared with handcrafted visual descriptors. Recently, many well-known CNN-based approaches push the field of recent studies on deep features, e.g., DeepEdge,[20] CSCNN,[40] and DeepContour.[41] Such promising results clearly demonstrate the ability that deep learning techniques can be extended and used as a powerful methodology for sketch generation. Although many interesting approaches have been proposed in recent years, there is also a huge gap between edges and sketches. The line of edges is discontinuous in depth and surface direction, and we extract low-level features to generate edge-maps. We extract high-level features and the sketches describe images by several continuous lines.

2.3 *Sketch-Based Image Retrieval*

Query by visual example has become a popular problem in recent years.[42,43] With the prevalence of touch-screen devices in daily life, sketch can be easily obtained and SBIR has already become a mainstream trend. The background story can be described as follows: (i) one sketch is worth 1000 words, which makes SBIR more popular and accurate than the traditional text-based image retrieval;[2] (ii) because the prevalence of touch-screen phones, tablets, and personal computers is increasing, SBIR is changing the way of searching for interesting things; and (iii) words are not always accurate to describe an object compared with sketches, especially if objects possess fine-grained details.

Most previous works of SBIR[1–6] require the following steps. First, they extract edges from images to obtain an approximate sketch. Second, a feature descriptor (e.g., shape contexts,[44] SURF,[45] SIFT,[46] and HOG[47]) is implemented on the generated edge-maps. Third, features of query sketches and edge-maps generated from natural images are matched by a distance calculation module, among of which k-nearest neighbors classification has been regarded as one of the top 10 data-mining algorithms,[48] due to its simplicity and efficiency. Most important of all, we rarely study the key role played by sketch generation in the previous work, to a great extent, which can bridge the semantic gap between sketches and images. In this paper, we give a detailed discussion of sketch generation on the impact of SBIR.

There are many strategies for sketch generation and we consider the strategy of feature fusion from different layers. The idea is similar to Ref. 49, which develops a network named holistically nested edge detection and is designed for edge detection. As we all know, sketches are multiscale in nature and face a problem of unknown scale. Using a local filter to detect the edge pixel, no matter what the size of the filter, will generate responses, either stronger or weaker. Therefore, we consider multiscale detection responses, which are used for the input image. As demonstrated in the experiment, multiscale detection is able to improve the performance of edge detection. On the other hand, the structural details of sketches become more obscure in deeper layer and a lot of useful information is missed.

Based on the above discussion, we try to address three important issues through SG-Net: (1) we train the model on sketches and the corresponding images using a per-pixel labeling cost; (2) we incorporate multiscale and multilevel learning of deep image features[49] via auxiliary cost function at each convolutional layer; and (3) we refine the rough sketch and generate hand-drawn-like sketches.[50] Per pixel cost functions make it possible that RoughSketchNet or fully convolutional networks[51] is effectively trained using image pairs. To remain the basic information and lower the noise, a specific mathematical morphology method is proposed to refine the former result and finally we get

hand-drawn-like sketches. As a subsequent step, TPS is further used to improve SBIR performance.

3 Methodology

Compared with the edge whose lines are discontinuous in the surface-normal direction, sketch is composed by continuous lines. In general, we summarize the difference between the edge and sketch as follows: (1) the sketch is more abstract and the line is continuous. Meanwhile, the representation contains both low-level and high-level features; (2) the shape of sketch varies very much and it does not need to satisfy the demand of pixel-to-pixel matching between the original image and the generated sketch.

For the first problem, we propose a model called RoughSketchNet, which aims to generate rough sketch. The exterior outline of sketch is more continuous, whereas the result is full of redundant information. Therefore, we then introduce a thinning algorithm and propose two kinds of template to obtain the human-drawing-like sketch. For the second problem, we exploit metamorphosis to generate sketches with different styles.

The details are as follows: Sec. 3.1 introduces the architecture of RoughSketchNet. Section 3.2 follows the notation of refinement in mathematical morphology and gives definitions for sketch refinement. We then introduce a new principle defined for sketches to solve the two-pixel width problem. In Sec. 3.3, we provide a detailed discussion of sketch deformation to generate different drawing style.

3.1 *Network Architecture of Sketch Generation*

The choice of hierarchy for our framework deserves some thought. We need the architecture: (1) to be deep, so as to efficiently generate perceptually multilevel features; and (2) to have multiple stages with different strides, which aims to capture the intrinsic scales of sketch.

Based on above discussion, we find that fine-tuning CNN based on the traditional image classification task is very helpful between low-level tasks and mid-level tasks, e.g., edge detection,[52] image classification,[53] object detection,[54,55] etc. According to our survey, we find that VGGNet[53] architecture design is a well-suited solution for sketch generation, since it is very deep (16 convolutional layers), dense (stride-1 convolutional kernels), and has multiple stages (five two-stride subsampling layers). Meanwhile, the conv layers are divided into five stages, in which a pooling layer is connected after each stage. We can capture the useful information by each conv layer with its receptive field size increasing. The use of this rich hierarchical information is hypothesized to help a lot. The starting point of our network design lies here.

According to the characteristic of sketch, we design crop-layer and multiscale VGGNet named of RoughSketchNet for sketch generation and the whole neural network is shown in Fig. 2. The improved details are as follows:

- We connect each side output layer to the last convolutional layer in each stage: conv1_1, conv2_2, conv3_3, and conv4_3. The architecture cannot be as deep as

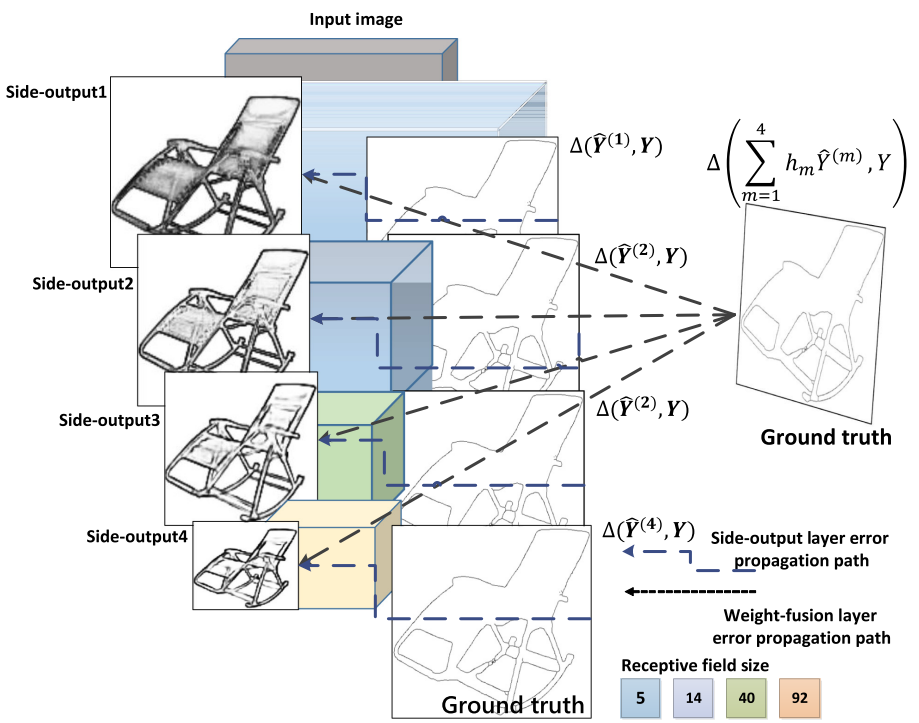

Fig. 2 Illustration of our network architecture for rough sketch generation, highlighting the error backpropagation paths. Side-output layers are inserted after convolutional layers. Deep supervision is imposed at each side-output layer, guiding the side-outputs toward sketch predictions with the characteristics we desire to get. The outputs of RoughSketchNet are multiscale and multilevel, as the size of side-output becomes smaller and the receptive field size becomes larger. In the setting of sketch generation, one weighted-fusion layer is added to automatically learn how to combine outputs from multiple scales. The whole network is trained with side-output layer error propagation path and weight-fusion layer error propagation path. They are marked by dashed lines.

previous versions. The main reason is that with the deeper network, the problem of edge smearing emerges and thus we get low quality sketch. However, it is difficult to learn such deep neural networks with multiple stages from scratch, so we use the pretrained network provided by Ref. 49 and use our specific training data to fine-tune the parameters.

- We use multiscale transform (0.8, 1.0, 1.2, 1.4, 1.6, and 2.0) of the input. As described in Ref. 56, the use of outer multiscale input improves the final sketch generation significantly. Thus, we run the single network on multiscale images to yield more accurate prediction. This strategy is applied on both training stage and testing stage such as "ensemble testing," which is particularly common in nondeep learning-based methods.[52]

In the training phase, the training data $S = \{(X_n, Y_n), n = 1, \ldots N\}$ are composed of original image X_n and the corresponding ground-truth Y_n. For each input, the M side-output layers are realized as classifiers and the corresponding weights are $w = [w^{(1)}, w^{(2)}, \ldots, w^{(M)}]$. For simplicity, we denote all standard network layer parameters as W. Hence, the objective function can be defined as follows:

$$L_{\text{side}}(W, w) = \sum_{m=1}^{M} \alpha_m l_{\text{side}}^{(m)}(W, w^m). \tag{1}$$

Here, α_m is a hyperparameter, which allocates each individual side-output layer with different loss weight. l_{side} denotes the image-level loss function for side-outputs, which is computed over all pixels between the image X and ground-truth Y.

Meanwhile, we add a "weight-fusion" layer in the network, which can be simultaneously learned during the training stage. The loss function of information fusion L_{fuse} is defined as

$$L_{\text{fuse}}(W, w, h) = Dist(Y, \hat{Y}_{\text{fuse}}), \tag{2}$$

where $\hat{Y}_{\text{fuse}} = \sigma(\sum_{m=1}^{M} h_m^{\hat{A}_{\text{side}}})$ with $h = (h_1, \ldots, h_M)$ represents the fusion result. $Dist(.., ..)$ is defined as distance measurement between ground truth labels and fused predictions.

The final objective function is minimized via stochastic gradient optimization algorithms, and we update the parameter by backpropagation method:

$$(W, w, h)^* = \arg \min[L_{\text{side}}(W, w) + L_{\text{fuse}}(W, w, h)]. \tag{3}$$

Some details of training and testing stage are referenced in Sec. 7.

3.2 *Sketch Refinement by Mathematical Morphology*

As shown in Fig. 1(c), there exists some redundant pixels (black regions with one-value pixels). As a result, sketches are composed of a few bold lines. In order to resolve this problem, we introduce mathematical morphology[57,58] to generate human-drawing-like sketches. The idea described here for the refinement comes from hit transform.[59,60]

Taken together, we give some basic notations in Sec. 3.2.1. In Sec. 3.2.2, we present an efficient two-pass algorithm. First, we use the template to process sketches and obtain two-pixel-width version. Second, we further refine the above results and obtain one-pixel-thick sketches. After the above processing, we ultimately get human-drawing-like sketches.

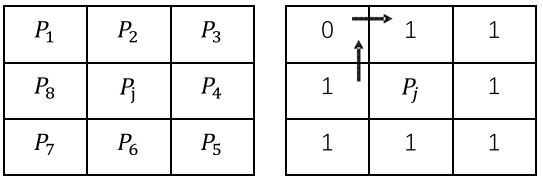

Fig. 3 (a) Points under consideration and their locations. (b) Pattern representation of one-to-zero and zero-to-one.

3.2.1 *Sketch refinement basic notation*

Sketches are a series of black and white lines of different widths. In the following text, we will give basic notation involved in the thinning algorithm.

Definition of eight-neighbors or four-neighbors: as shown in Fig. 3(a), P_1, P_2, P_3, P_4, P_5, P_6, P_7, and P_8 are the eight neighbors of P_j. P_1, P_3, P_5, and P_7 are the four neighbors of P_j. We define P_2, P_4, P_6, and P_8 representing the object and the pixel is represented as 1. P_1, P_3, P_5, and P_7 represent the background and the pixel is represented as 0.

Definition of the crossing number of a pixel: as shown in Fig. 3(b), the number of transition is defined as pixels changing from a white point to a black point and vice versa [when the points in $N(p)$ are traversed in a counterclockwise order]. Therefore, we define the number of eight-connectedness as follows:

$$A(p) = \sum_{i=1}^{8} |P_{i+1} - P_i|, \tag{4}$$

where $P_9 = P_1$ and $A(p)$ is equal to twice the number of black four-components in $N(p)$. As defined in Ref. 61, the crossing number $A(p)$ is the number of crosses from a white point to a black point.

3.2.2 *Sketch refinement algorithm and principle*

In this section, we adopt a two-pass refinement algorithm to determine whether black pixels should be removed or not. In fact, the whole process for sketch refinement could be seen as an iterative and parallel algorithm. In the parallel algorithm, the deletions of pixels in the n'th iteration will depend only on the result that remains after the $(n-1)$'th. Therefore, all pixels can be examined independently in a parallel manner in each iteration.

For the first step, we define two refinement rules in nearly all cases in Figs. 4 and 5. The proposed rules are derived by rotating each mask by one bit from left to right or right to left for one complete rotation to achieve the thinning rules. This process is repeated until no more changes occur in the template.

As shown in Fig. 6, there exists some two-pixel-width pixels in the cross-court position for results of the first process. The main reason is that rule-2 (Fig. 5) is not applicable for redundant pixels. Therefore, the result should be further refined to get a better sketch with one-pixel-width.

1	1	0		1	1	1		1	1	1		1	1	1		1	1	1
0	P_j	0		0	P_j	0		0	P_j	1		0	P_j	1		0	P_j	1
0	0	0		0	0	0		0	0	0		0	0	1		0	1	1
(1)				(2)				(3)				(4)				(5)		

Fig. 4 Masks used to derive refinement rules.

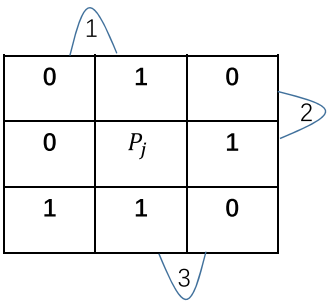

Fig. 5 Counting the number of 0-to-1 patterns in a clockwise direction.

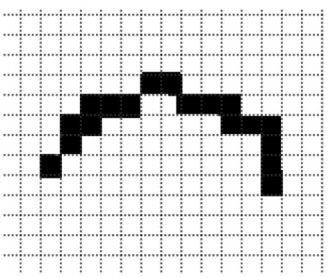

Fig. 6 The redundant slash after the process of pass-1.

*	1	*		*	1	*		0	0	*		*	0	0
1	P_j	0	OR	0	P_j	1	OR	0	P_j	1	OR	1	P_j	0
*	0	0		0	0	*		*	1	*		*	1	*
(1)				(2)				(3)				(4)							

Fig. 7 Elimination template for the process of pass-2.

For the second step, we propose four elimination templates to remove redundant pixels. As shown in Fig. 7, * represents any pixel value. We use elimination templates scanning on the sketch. Any pixels incompatible to the origin template will be edited out.

The result of the entire procedure is shown in Fig. 8. We can see that the outcome of RoughSketchNet is redundancy. For example, there are many redundancy lines in the shoelaces, and some discontinuous lines exist in the inner of the chair for the original VGGNet. These lines can be represented as a set of edges, but they are useless for sketches. These problems are improved in RoughSketchNet, where the lines are continuous shown in Fig. 8(c). But there are many redundant pixels and the lines are more than two-pixel-width. At last, we give the final result generated by SG-Net in Fig. 8(d), which is consistent with the view in human cognition that we draw an object by several lines and finally get one-pixel-thick sketch.

Fig. 8 (a) Original images. (b) The result of RCF[56] that is based on VGGNet.[53] (c) The result of RoughSketchNet. (d) The result of SG-Net.

3.3 *Sketch Metamorphosis*

All images and the corresponding sketches are pixel-to-pixel as described above. Whereas sketch deformation has a number of applications, from animation to medical imaging.[62] As shown in Fig. 9, the freehand sketches form vary with different styles. Therefore, TPS is applied on sketch metamorphosis to generate different drawing styles.

As described in Ref. 63, TPS is useful for the computation of image interpolation and smoothing. When converting position of x-coordinate to another form and the same with y-coordinate, TPS is commonplace in the domain of biological shape. In the case, where x is two-dimensional, the fit aims to have a mapping function $f(x)$ between point-sets $\{x_i\}$ and $\{y_i\}$. Both of them minimize the following energy function:

$$E_{\text{tps}}(f) = \sum_{i=1}^{K} \|y_i - f(x_i)\|^2. \tag{5}$$

The parameter λ in the following construction is a so-called tuning parameter, which aims to control rigidity of an appropriate deformation. Finally, we achieve the effect of smoothing on the sketch. That is to say, the parameter is used to balance the aforementioned criterion. As a result, we have a minimizing energy function (the second section minimizes the nonnegative quantity) as follows:

$$E_{\text{tps,smooth}}(f) = \sum_{i=1}^{K} \|y_i - f(x_i)\|^2 + \lambda \iint \left[\left(\frac{\partial^2 f}{\partial x_1^2} \right)^2 + \left(\frac{\partial^2 f}{x_1 x_2} \right)^2 + \left(\frac{\partial^2 f}{\partial x_2^2} \right)^2 \right] \mathrm{d}_{x_1} \mathrm{d}_{x_2}. \tag{6}$$

Based on the aforementioned theory and algorithm, the final result of sketch deformation is shown in Fig. 10.

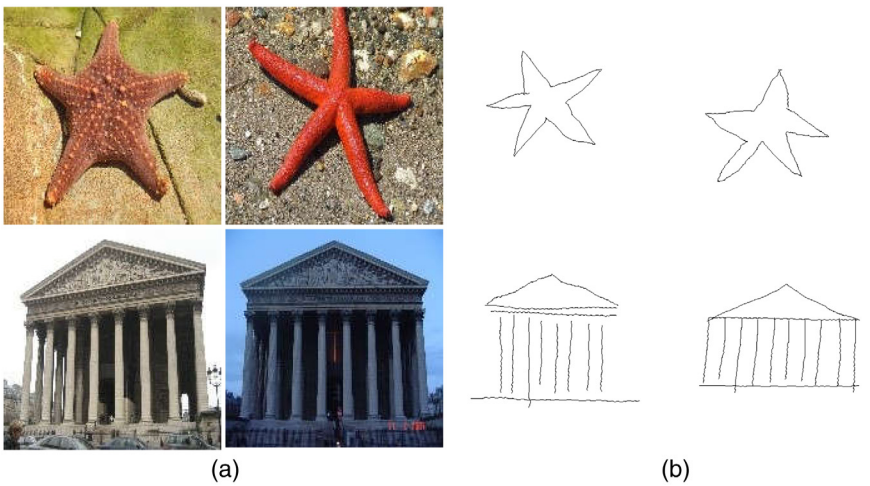

Fig. 9 Samples of image-sketch pairs. (a) The Flickr15k.[2] (b) The corresponding freehand sketch.

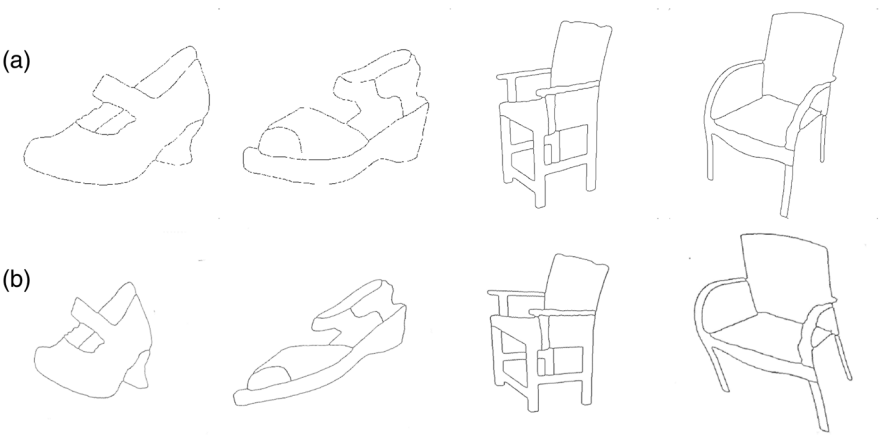

Fig. 10 (a) The original sketches. (b) The result after deformation.

4 Sketch Generation Experiments

This section evaluates the performance of the proposed SG-Net. First, we introduce the experimental dataset, evaluation metric, and the experimental settings. Then, we use different sketch generation methods to validate the model. To further explore the validity and accuracy of the model, six participants are required to rate the similarity degree of sample pairs.

4.1 *Dataset and Evaluation Metric*

4.1.1 *Dataset*

We use Flickr15k dataset[2] as the benchmark in the SBIR system to evaluate the SG-Net framework. (i) The dataset has ∼15k photographs sampled from Flickr and are manually labeled into 33 categories according to shape; and (ii) the dataset includes 330 freehand drawing sketches drawn by 10 nonexpert participants. In the experiment, we use natural images in Flickr15k as the candidates and 330 sketches without any processing as the query images.

4.1.2 *Evaluation metric*

We measure sketch using qualitative subjective evaluation, which aims to evaluate the quality of generated sketches by subjective measures of human.

4.2 *Experimental Settings*

We train the neural network by stochastic gradient descent with momentum of 0.9. We use a learning rate initiation of 0.0001 and decrease the rate by 0.1 every 30 epochs. The weight decay parameter is 0.0005. Overall training the model takes ∼1 day based on a PC with Intel Xeon E5-2630 v3 and GTX 1080 GPU. We set the minibatch size of images to 64. The SG-Net framework is implemented with Caffe.[64]

4.3 *Comparison with State-of-the-Art Methods*

We give the performance evaluation of different sketch generation methods on 300 images.[2] As shown in Fig. 11, (b) lacks bottom information and (c–e) have too much disorder and interferential information. On the other hand, (b) filters out some unimportant and unnecessary line information compared with (c–e). Finally, we find that our proposed algorithm can produce a clear line that can represent the most important information of natural images.

4.4 *Qualitative Results*

The result is evaluated by subjective measures of human. Six volunteers evaluate the images and the corresponding sketches through participant observation. We assume that

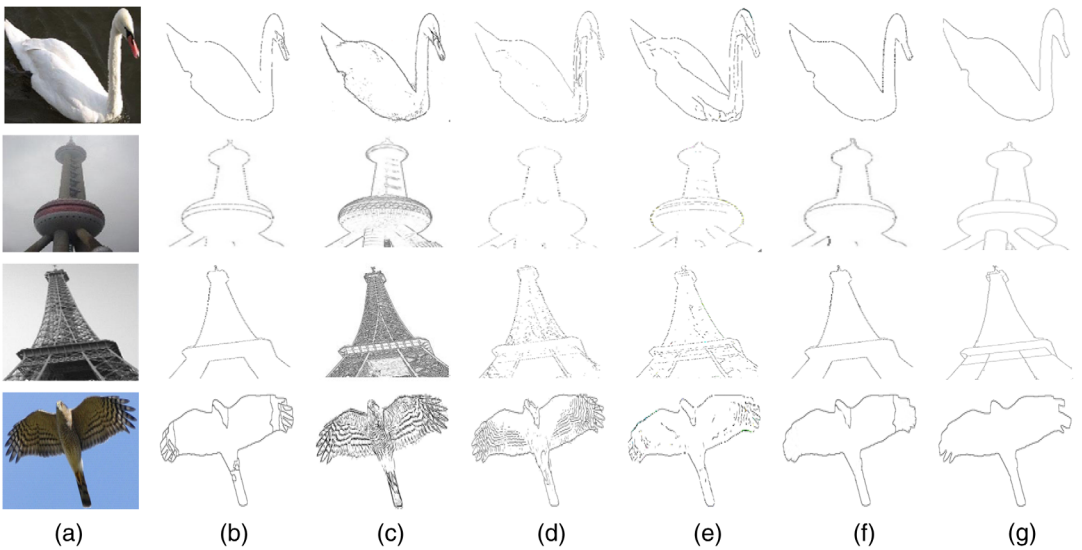

Fig. 11 Examples of different sketch generation methods are given. (a) Original image, (b) results of Pb,[21] (c) CLD,[65] (d) PS,[24] (e) contour cut,[66] (f) PerceptualEdge,[10] and (g) our proposed method. We can observe that sketches generated from SG-Net are at a similar level of humans.

Table 1 The result is obtained by a set of observers assessing similarity of sample pairs.

	Observer						
Method	1	2	3	4	5	6	Average
SG-Net	81.0%	85.0%	81.2%	88.0%	90.0%	84.7%	**84.9%**
PS[24]	79.3%	83.3%	79.7%	86.0%	87.0%	82.3%	82.9%
Pb[21]	76.7%	81.3%	77.3%	83.3%	85.0%	80.0%	80.6%

participants are not familiar with the entire process and only use their past experience to estimate the quality of sketch generation, which ensures the fairness for all methods. By using this result, the average accuracy can be easily obtained, which aims to avoid visual mistakes.

As shown in Table 1, the sketches are generated by SG-Net, PS,[24] and Pb.[21] Each of the participants judge the similarity of 300 sample pairs. Here, 1 indicates the similarity to each other. If the observers think that the sample is similar, the scores are added up. Finally, the above score divided by 300 is the proportional similarity. The result shows that the predicted results of our model accord well with the observed data and the positive prediction value of SG-Net is higher than two additional methods.

5 SBIR Experiments

The evaluation of sketch quality is an important topic. We follow the idea[1–6] and use the result in Fig. 11 to validate our model. SBIR is a challenging task due to the cross-domain gap between sketches and natural images. Therefore, sketch generation or converting natural images into edge-maps is necessary to improve retrieval performance. In the next experiment, we use the same dataset as described in Sec. 4.1.

5.1 *Evaluation Metric*

In our experiment, we measure sketch quality using the criteria of Average Precision (mAP). During the test phase of SBIR system, a database with a large number of images is searched to find top N images meeting the specification of the input sketch. If giving a high-quality sketch, we can obtain more accurate images and the mAP is higher. The definition is as follows:

$$\mathrm{mAP} = \frac{1}{Q_R} \sum_{p \in Q_R} \mathrm{AP}(p),$$

where Q_R is the number of queries, AP is the average precision scores for each query, and mAP for a set of queries is the mean of the average precision scores for each query.

5.2 *Evaluation for Different Sketch Generation Methods*

In this section, the extracted feature remains the same and the sketch we evaluate is generated from different methods (shown in Fig. 11). As shown in Fig. 12, the mAP score of SG-Net is higher than most of the methods. We can observe that our method achieves competitive performance with PerceptualEdge and Im2Sketch. Note that perceptual grouping organizes image edges into meaningful structures and combines multiple Gestalt principles as cues for edge grouping from multiple images, whereas SG-Net generates a sketch only from a single image. Although we only use one image to generate the sketch, our proposed method still achieves comparable results with PerceptualEdge. In addition, Im2Sketch trains the SBIR model directly from human sketch datasets. While our SG-Net trains SBIR model from edge-maps extracted from natural images, which is the same protocol as PerceptualGrouping. These results suggest that using SG-Net to generate sketches from images could boost SBIR retrieval performance and be a practical solution due to lack of large-scale sketch dataset.

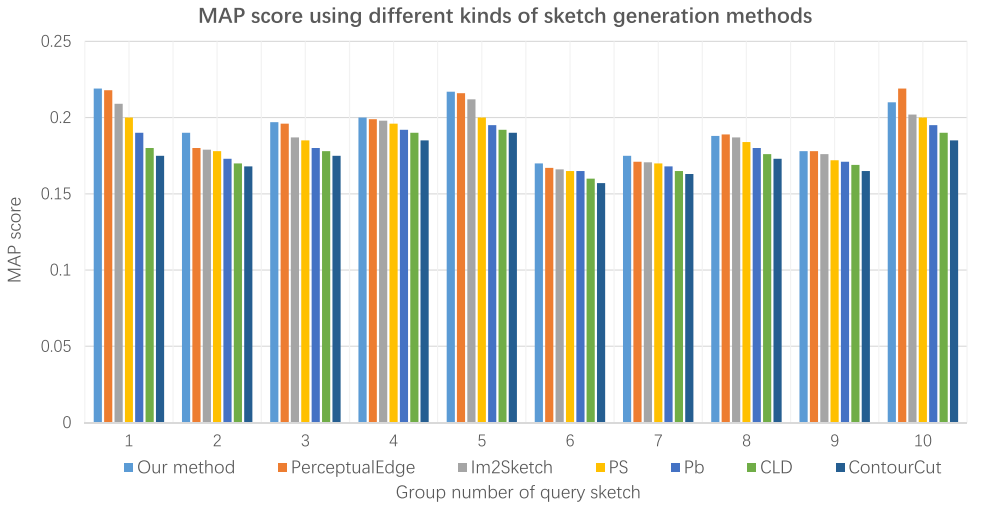

Fig. 12 Comparison of MAP for different methods, such as SG-Net, PerceptualEdge, Im2Sketch, PS, Pb, CLD, and ContourCut.

5.3 *Further Analysis*

In this section, we detail the impact of network structure and sketch deformation on the Flickr15k dataset.

5.3.1 *Deep structure*

We give a detailed study to explore network design of deeper and multiscale in Fig. 13 and Table 2, respectively. We give three chairs as the input and the result is shown in Fig. 13. When RoughSketchNet goes deeper, the details of sketch information become more and more obscure, and much helpful information is missing. More especially, the result of conv5_3 loses too much information in the outer region. Therefore, our model prevents the loss of the details and the rupture of lines, which ensures to generate high-quality sketches.

Table 2 reports MAP on Flickr15k dataset. From this result, we see that the model with multiscale input achieves better performance compared with the methods based on single-scale. The main reason for performance improvement is as follows: sketches using a multiscale approach make full use the edge information and thus the performance is better than before.

5.3.2 *Ablation study*

We perform an ablation study to illustrate the effect of sketch deformation. We conduct an assessment on the Flickr15k dataset with no deformation. The result is shown in Table 3 and Fig. 14, respectively. From this ablation study, we can draw the following conclusions:

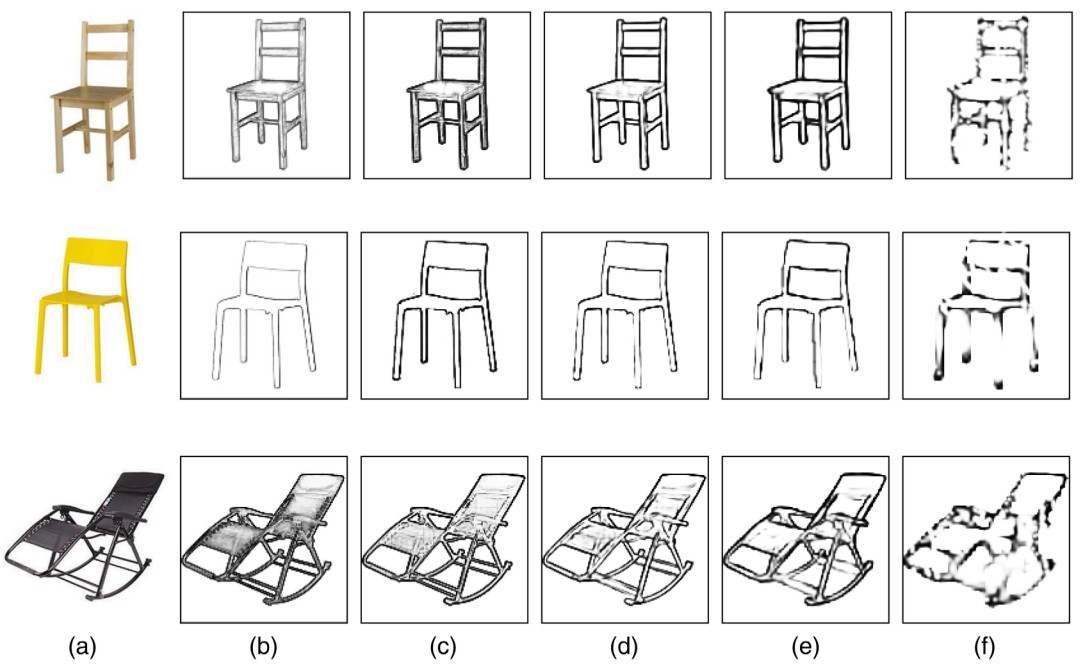

Fig. 13 (a) Original image, (b) detection results of conv1_1, (c) conv2_2, (d) conv3_3, (e) conv4_3, and (f) conv5_3.

Table 2 The value of MAP for SBIR with and without multiscale transform.

Descriptor methods	MAP
SG-Net (w/o multiscale)	0.1662
SG-Net (w/ multiscale)	0.1718

Table 3 SBIR comparison using MAP to evaluate indicator for sketch without deformation and with deformation.

Descriptor methods	MAP
SG-Net (w/o deformation)	0.1718
SG-Net (w/ deformation)	0.1838

- As shown in Table 3, MAP of SG-Net with deformation achieves a fraction of one percent improvement on Flickr15k.
- Figure 14 presents several sketch queries and their retrieval results over the Flickr15k dataset. From the result, we can see that sketches with style variety have an important influence on the retrieval performance and the returned top ranking images correspond closely to the query sketches shape. Although the bike and duck return an error in the fourth and fifth columns, the majority of results is relevant.

In a word, the experiments show that the algorithm could promote both the efficiency and accuracy comparing with the traditional methods. The principle reason focuses on the two following points. The first is that we introduce RoughSketchNet, which makes full use of deep features with multiscale and crop-layer and thus obtains all contextual information. Whereas previous methods use traditional feature descriptors and the generated sketch contains much redundant information. The second is the use of TPS. Because of a person with poor drawing skills or limited time, the final result may show different styles for the same object. In these cases, we have to enlarge spatial coverage on the style of creation and, to a large extent, it can bridge the gap between the queries and targets. But previous theories are mainly based on the hypothesis that images and sketches possess one-to-one mapping.

6 Conclusion

In this paper, we propose a unified approach named SG-Net to generate freehand sketches. We demonstrate the application on SBIR using the same descriptor with different sketch generation methods. In the experiment, the results indicate the stability and reliability of the proposed model for SBIR. To further generalize our method, we adopt sketch metamorphosis to improve the retrieval performance.

7 Appendix A: Training and Testing of RoughSketchNet

7.1 *Training Stage*

Because there is a heavy bias toward nonlabeled pixels for ground-truth datasets, a new item that automatically balances the loss between positive and negative classes is introduced via a per-pixel class-balancing weight β. By adding this parameter, the loss function offsets imbalances between edge and nonedge samples. In particular, we use a class-balanced cross-entropy loss function in Eq. (7) with j iterating over image spatial dimensions:

$$l_{\text{side}}^{(m)}[W, w^{(m)}] = -\beta \sum_{\text{j}\in Y_+} \log \Pr[y_j = 1 | X; W, w^{(m)}] - (1-\beta) \sum_{j \in Y_-} \log \Pr[y_j = 0 | X; W, w^{(m)}]. \quad (7)$$

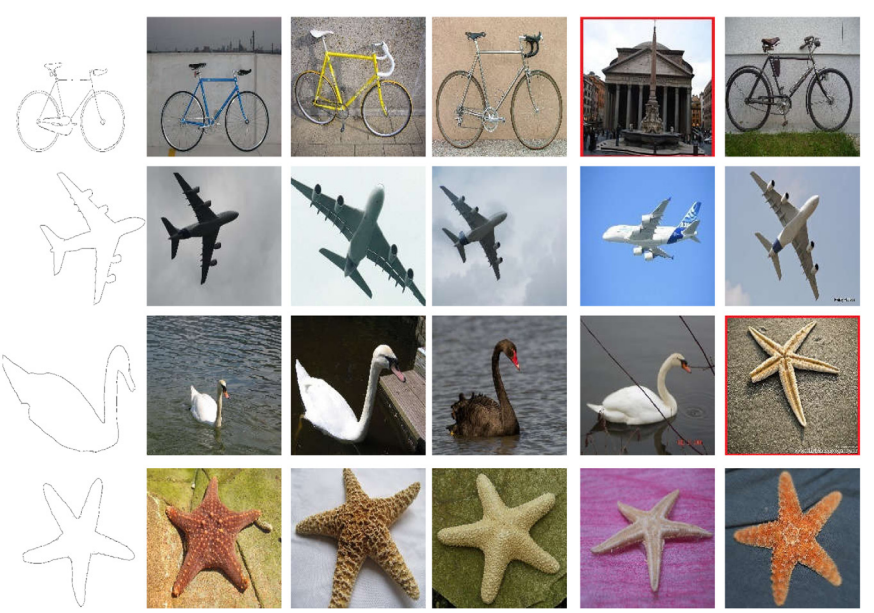

Fig. 14 Example query sketch and the top-5 ranking results (from left to right) with sketch deformation. Dashed red box shows false positives.

Here, we denote $|Y|$, $|Y_+|$, and $|Y_-|$ as all the pixels in the sketch S, edge (positive) pixels, and nonedge (negative) pixels in the image I, respectively. β is simply denoted as $|Y_-|/|Y|$ and $1-\beta = |Y_+|/|Y|$. The class probability $\Pr[y_j = 1|X; W, w^{(m)}] = \sigma[a_j^{(m)}] \in [0,1]$ is computed on the activation value at each pixel j using the sigmoid function $\sigma(.)$. Next, sketch map prediction $\hat{Y}(m)_{\text{side}} = \sigma[\hat{A}(m)_{\text{side}}]$ can be obtained by each side-output layer, where $\hat{A}(m)_{\text{side}} \equiv \{a_j^{(m)}, j = 1, \ldots, |Y|\}$ is the activation for the side-output layer m.

7.2 *Testing Stage*

Given an image X, we obtain predictions from side output layers and the weighted-fusion layer as follows:

$$(\hat{Y}^I_{\text{fuse}}, \hat{Y}^{I_1}_{\text{side}}, \ldots, \hat{Y}^{I_4}_{\text{side}}) = P[X, (W, w, h)]. \tag{8}$$

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61273364, 61473031, and 61472029), the Fundamental Research Funds for the Central Universities (2018YJS035).

References

1. M. Eitz et al., "Sketch-based image retrieval: benchmark and bag-of-features descriptors," *IEEE Trans. Visual Comput. Graphics* **17**(11), 1624–1636 (2011).
2. R. Hu and J. Collomosse, "A performance evaluation of gradient field HOG descriptor for sketch based image retrieval," *Comput. Vision Image Understanding* **117**(7), 790–806 (2013).
3. M. Eitz et al., "A descriptor for large scale image retrieval based on sketched feature lines," in *Eurographics Symp. on Sketch-Based Interfaces and Modeling*, ACM, pp. 29–36 (2009).
4. R. Hu, M. Barnard, and J. Collomosse, "Gradient field descriptor for sketch based retrieval and localization," in *IEEE Int. Conf. on Image Processing*, IEEE, pp. 1025–1028 (2010).
5. R. Hu, T. Wang, and J. Collomosse, "A bag-of-regions approach to sketch-based image retrieval," in *IEEE Int. Conf. on Image Processing*, IEEE, pp. 3661–3664 (2011).
6. C. Zou et al., "Sketch-based shape retrieval using pyramid-of-parts," CoRR abs/1502.04232 (2015).
7. M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Trans. Graphics* **31**, 44 (2012).
8. Y. Li, T. M. Hospedales, and Y.-Z. Song, "Free-hand sketch recognition by multi-kernel feature learning," *Comput. Vision Image Understanding* **137**, 1–11 (2015).
9. K. Li et al., "Universal perceptual grouping," in *European Conf. on Computer Vision*, Springer (2018).
10. Y. Qi et al., "Making better use of edges via perceptual grouping," in *Conf. on Computer Vision and Pattern Recognition*, IEEE, pp. 1856–1865 (2015).
11. M. Zhang et al., "Recognition of facial sketch styles," *Neurocomputing* **149**(1), 1188–1197 (2015).
12. I. Berger et al., "Style and abstraction in portrait sketching," *ACM Trans. Graphics* **32**(4), 55 (2013).
13. U. R. Muhammad et al., "Learning deep sketch abstraction," CoRR abs/1804.04804 (2018).
14. T. Tuytelaars, "Sketch classification and classification-driven analysis using Fisher vectors," *ACM Trans. Graphics* **33**, 174 (2014).
15. Q. Yu et al., "Sketch-a-net that beats humans," BMVC (2015).
16. J. Zhang, "The PASCAL visual object classes challenge," in *Int. Conf. on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, Springer-Verlag, pp. 117–176 (2005).
17. P. Young et al., "From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions," *Trans. Assoc. Comput. Ling.* **2**, 67–78 (2014).
18. P. Srinivasan, Q. Zhu, and J. Shi, "Many-to-one contour matching for describing and discriminating object shape," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, IEEE, pp. 1673–1680 (2010).
19. P. Sermanet et al., "OverFeat: integrated recognition, localization and detection using convolutional networks," CoRR abs/1312.6229 (2013).
20. G. Bertasius, J. Shi, and L. Torresani, "DeepEdge: a multi-scale bifurcated deep network for top-down contour detection," in *Conf. on Computer Vision and Pattern Recognition*, IEEE, pp. 4380–4389 (2015).
21. P. Arbeláez et al., "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 898–916 (2010).
22. B. Gooch, "Human facial illustrations: creation and psychophysical evaluation," *ACM Trans. Graphics* **23**(1), 27–44 (2004).
23. Y. Li et al., "Free-hand sketch synthesis with deformable stroke models," *Int. J. Comput. Vision* **122**(1), 169–190 (2017).
24. C. E. Guo, S. C. Zhu, and Y. N. Wu, "Primal sketch: integrating structure and texture," *Comput. Vision Image Understanding* **106**, 5–19 (2007).
25. X. Zhang et al., "Photo-to-sketch transformation in a complex background," *IEEE Access* **5**, 8727–8735 (2017).
26. I. J. Goodfellow et al., "Generative adversarial nets," in *Int. Conf. on Neural Information Processing Systems*, MIT Press, pp. 2672–2680 (2014).
27. T. Kim et al., "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. of the 34th Int. Conf. on Machine Learning* (2017).
28. Y. Qi et al., "Im2Sketch: sketch generation by unconflicted perceptual grouping," *Neurocomputing* **165**, 338–349 (2015).
29. S. Marvaniya et al., "Drawing an automatic sketch of deformable objects using only a few images," *Lect. Notes Comput. Sci.* **7583**, 63–72 (7583).
30. Y. Liang, "Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis," in *Conf. on Computer Vision and Pattern Recognition*, IEEE, pp. 2216–2223 (2012).
31. K. Schindler and D. Suter, "Object detection by global contour shape," *Pattern Recognit.* **41**(12), 3736–3748 (2008).
32. C. L. Zitnick and P. Dollár, "Edge boxes: locating object proposals from edges," *Lect. Notes Comput. Sci.* **8693**, 391–405 (2014).
33. B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(11), 2189–2202 (2012).
34. J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 679–698 (1986).
35. M. Leordeanu, R. Sukthankar, and C. Sminchisescu, "Generalized boundaries from multiple image interpretations," *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(7), 1312–1324 (2014).
36. J. J. Lim, C. L. Zitnick, and P. Dollar, "Sketch tokens: a learned mid-level representation for contour and object detection," in *Conf. on Computer Vision and Pattern Recognition*, IEEE, pp. 3158–3165 (2013).
37. H. R. Roth et al., "Improving computer-aided detection using convolutional neural networks and random view aggregation," *IEEE Trans. Med. Imaging* **35**(5), 1170–1181 (2016).
38. Y. Zhang et al., "Improving object detection with deep convolutional networks via Bayesian optimization and structured prediction," in *IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE Computer Society, pp. 249–258 (2015).
39. Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* **521**(7553), 436–444 (2015).
40. J. J. Hwang and T. L. Liu, "Pixel-wise deep learning for contour detection," CoRR abs/1504.01989 (2015).
41. W. Shen et al., "DeepContour: a deep convolutional feature learned by positive-sharing loss for contour detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*," IEEE Computer Society, pp. 3982–3991 (2015).
42. X. Qian et al., "Enhancing sketch-based image retrieval by re-ranking and relevance feedback," *IEEE Trans. Image Process.* **25**(1), 195–208 (2016).
43. S. Deniziak and T. Michno, "Query by shape for image retrieval from multimedia databases," in *Int. Conf. on Beyond Databases, Architectures and Structures*, Springer, Cham, pp. 377–386 (2015).
44. S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," in *IEEE Int. Conf. on Computer Science and Information Technology*, IEEE pp. 471–474 (2010).
45. H. Bay et al., "Speeded-up robust features (SURF)," *Comput. Vision Image Understanding* **110**(3), 346–359 (2008).
46. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision* **60**(2), 91–110 (2004).
47. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, IEEE, pp. 886–893 (2005).
48. X. Wu et al., "Top 10 algorithms in data mining," *Knowl. Inf. Syst.* **14**(1), 1–37 (2008).
49. S. Xie and Z. Tu, "Holistically-nested edge detection," in *IEEE Int. Conf. on Computer Vision*, IEEE, pp. 1395–1403 (2016).
50. T. Y. Zhang, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM* **27**(3), 236–239 (1984).
51. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4), 640–651 (2017).

52. P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(8), 1558–1570 (2015).
53. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," CoRR abs/1409.1556 (2014).
54. R. Girshick, "Fast R-CNN," Computer Science arXiv:1504.08083 (2015).
55. N. Silberman, D. Sontag, and R. Fergus, "Instance segmentation of indoor scenes using a coverage loss," *Lect. Notes Comput. Sci.* **8689**, 616–631 (2014).
56. Y. Liu et al., "Richer convolutional features for edge detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 5872–5881 (2017).
57. R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-9**(4), 532–550 (2009).
58. J. Angulo et al., "Mathematical morphology and its applications to image and signal processing," *Comput. Imaging Vision* **6671**(4), 384 (2011).
59. B. Naegel, N. Passat, and C. Ronse, "Grey-level hit-or-miss transforms—Part I: unified theory," *Pattern Recognit.* **40**(2), 635–647 (2007).
60. A. K. Jain, *Fundamentals of Digital Image Processing*, Wiley-Blackwell, Hoboken, Jersey (2011).
61. C. Hilditch, "Linear skeleton from square cupboards," in *Machine Intelligence*, B. Meltzer and D. Michie, Eds., Vol. **6**, pp. 403–420, Elsevier, New York (1969).
62. T. Ju et al., "A geometric database for gene expression data," in *Proc. of the Eurographics/ACM SIGGRAPH Symp. on Geometry Processing*, pp. 166–176 (2003).
63. W. J. Schempp and K. Zeller, *Constructive Theory of Functions of Several Variables*, Springer-Verlag, Berlin (1977).
64. Y. Jia et al., "Caffe: convolutional architecture for fast feature embedding," in *22nd Proc. of the ACM Int. Conf. on Multimedia*, pp. 675–678 (2014).
65. H. Kang, S. Lee, and C. K. Chui, "Coherent line drawing," in *Int. Symp. on Non-Photorealistic Animation and Rendering*, ACM, pp. 43–50 (2007).
66. R. Kennedy, J. Gallier, and J. Shi, "Contour cut: identifying salient contours in images by solving a Hermitian eigenvalue problem," in *IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE Computer Society, pp. 2065–2072 (2011).

Xingyuan Zhang received his bachelor's degree from Tangshan College in 2013. Currently, he is a PhD candidate at Beijing Jiaotong University, China. His current research interests include deep learning, computer vision, and image processing.

Yaping Huang received her BS, MS, and PhD degrees from Beijing Jiaotong University, China, in 1995, 1998, and 2004, respectively. Currently, she is a professor at Beijing Jiaotong University. Her research interests include pattern recognition, computer vision, computer vision, image processing, and biological inspired object recognition.

Qi Zou received her BS, and PhD degrees from Beijing Jiaotong University, China, in 2001 and 2006, respectively. Currently, she is a professor at Beijing Jiaotong University. Her research interests include pattern recognition, computer vision, computer vision, and target tracking.

Qingji Guan is a PhD candidate in the School of Computer and Information Technology at Beijing Jiaotong University, China. Her research interests include deep learning, computer vision, and digital image processing.

Junbo Liu is a PhD candidate in the School of Computer and Information Technology at Beijing Jiaotong University, China. His research interests include deep learning, computer vision, and digital image processing.