

SketchNet: Sketch Classification with Web Images

Hua Zhang¹, Si Liu¹, Changqing Zhang², Wenqi Ren², Rui Wang^{1*}, Xiaochun Cao¹

¹State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering,
Chinese Academy of Sciences

²School of Computer Science and Technology, Tianjin University

{zhanghua, liusi, wangrui, caoxiaochun@iie.ac.cn, zhangchangqing@tju.edu.cn,
rwq.renwenqi@gmail.com }

Abstract

In this study, we present a weakly supervised approach that discovers the discriminative structures of sketch images, given pairs of sketch images and web images. In contrast to traditional approaches that use global appearance features or rely on keypoint features, our aim is to automatically learn the shared latent structures that exist between sketch images and real images, even when there are significant appearance differences across its relevant real images. To accomplish this, we propose a deep convolutional neural network, named **SketchNet**. We firstly develop a triplet composed of sketch, positive and negative real image as the input of our neural network. To discover the coherent visual structures between the sketch and its positive pairs, we introduce the softmax as the loss function. Then a ranking mechanism is introduced to make the positive pairs obtain a higher score comparing over negative ones to achieve robust representation. Finally, we formalize above-mentioned constrains into the unified objective function, and create an ensemble feature representation to describe the sketch images. Experiments on the TU-Berlin sketch benchmark demonstrate the effectiveness of our model and show that deep feature representation brings substantial improvements over other state-of-the-art methods on sketch classification.

1. Introduction

Sketch can be seen as the abstract representation of expressing some types of ideas. Moreover, sketch image can convey information that is hard to describe using text without requiring a tremendous amount of effort (e.g. Sketch2photo [7]). With the popularity of touch devices, sketch has been attracted more and more researches' attentions in the computer vision and graphic field. There exist

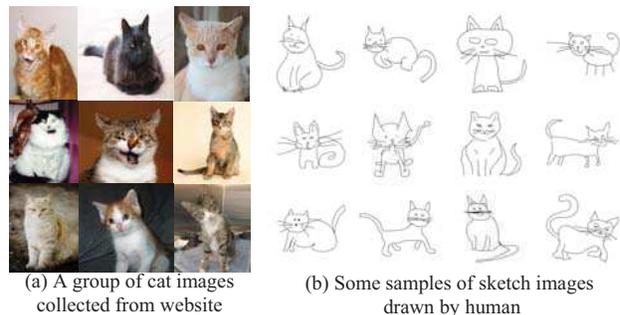


Figure 1. Motivation of our proposed method. Humans draw sketch images with their prior knowledges, which would generate a large variations for the same category (b). Sketch classification can be seen as the reverse processing of drawing sketch. We introduce the real images (a) as the references, which can bridge the representation gap between sketch images in the same category.

various interesting applications including sketch based image retrieval [14, 34], human-computer interaction [10], and other relevant works [31, 3, 15, 21, 22, 24, 28].

While most existing methods use global appearance [13] or interesting points [1] based feature representations to model sketches, recent work demonstrate that the effective way of using localized part based representation with embedding structure information [14, 2, 29]. Based on the extracted low-level hand-crafted local features [1, 2, 13], one way is following the process of image matching [30] to find the most-similar image. The other way is based on the paradigm of bag of features [17], and then train the classifiers [4] of categories using histogram representations.

One of the biggest limitations of traditional methods is the sketch image itself as shown in Fig.1 (b): 1) Without the texture and color informations, distinct categories appear to be similar e.g. tire and donut. 2) Different drawing styles can cause the variation of sketches, which would further enlarge the intra-similarities. 3) The existing localized sketch descriptors [9, 25, 1] are sensitive to the view perspectives and some appearance cues (the stripes of tigers) of drawn

*corresponding author

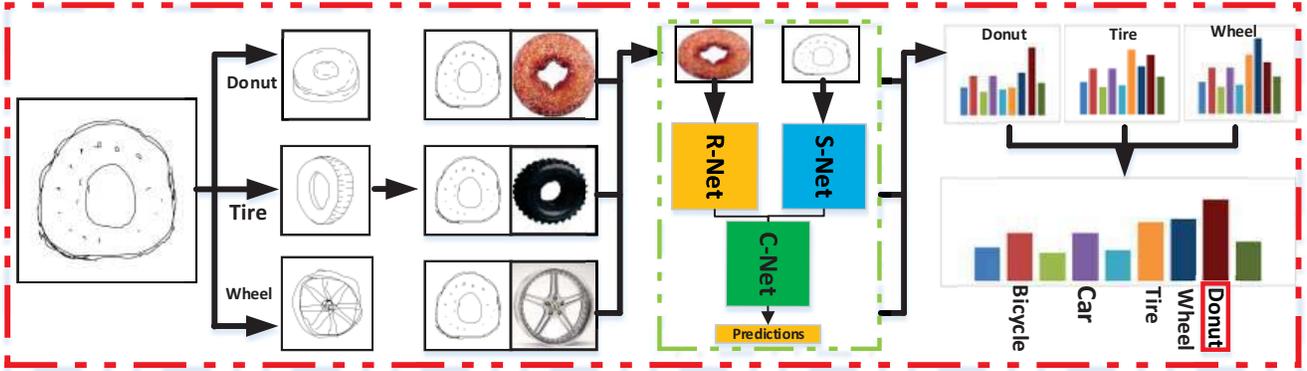


Figure 2. **Sketch classification with real image embedding using SketchNet.** Given a sketch image of an object, *e.g.*, donut, we find its initialized category predictions (Donut, Tire and Wheel) based on the pre-trained sketch model. Then the test pairs are constructed based on their visual similarity. Conditioned on these test pairs, we use our SketchNet model to achieve the prediction scores of each test image pair. Finally, all the predictions are merged to make the decision to identify the sketch category.

sketches. Furthermore, the ability of learning algorithms to train the classification models are also influenced by the handcrafted features and the capacity of classifiers (SVM [4]) to memorize feature information.

So, how can we achieve robust visual representation for the sketch image, and develop the classifiers to memorize more feature information? To this end, we need to automatically identify the discriminative visual patterns in each sketch image whose appearance may be significant differences. This is a challenging task considering the above-mentioned factors and the limited useful information of sketch images. Fortunately, there exists large amounts of web image data online with weakly supervised labels as shown in Fig.1 (a), which would be helpful to identify the latent discriminative structure of sketch images via discovering the coherent appearance between the sketch and real images of the same category. This assumption is supported by the process of human recognizing sketches: we can classify sketch images even without seeing any sketch images. Given the task, deep Convolutional Neural Network (CNN) [20] is proved to be a valuable model choice since it allows for large scale feature representation learning.

Specifically, in this paper, we propose a novel deep convolutional neural network, named **SketchNet** for sketch classification. To construct the auxiliary repository, the real images are collected from the web which covers all the sketch categories in the TU-Berlin sketch dataset [12]. To extract the real reference images for each training sketch, we first train an preliminary model based on Alexnet [20] following the fine-tuning process. Afterwards, the top K predicted category labels of each training sketch is extracted based on the pre-trained Alexnet model. For each training sketch, we find the most visual similar real images from the image sets of top predicted categories to construct the training pairs. Thus, the sketch with the real images which are in the same class is used to generate the positive im-

age pair while the sketch with the real images which are in distinct classes are defined as negative image pair. Next, a triplet is constructed based on the positive and negative pairs. Our proposed SketchNet contains three subnetworks: **R-Net** is used to extract features from the real images. **S-Net** is applied on the sketch images. And the **C-Net** is proposed to discover the common structures between real images and sketches. To guarantee the positive pair achieves a higher prediction score on the corresponding category than the negative pair, we customize a novel loss function based on the extracted features. In the test time as shown in Fig. 2, we feed the test pairs composing of sketches and their similar real images into SketchNet to set the category predictions. Finally, the predictions are merged together to achieve the final results. Extensive experiments are conducted on the TU-Berlin sketch dataset [12] and the experimental results show that our SketchNet can significantly improve the performance of sketch classification by introducing the real images as the references.

The contributions of our proposed method can be summarized as: (1) We propose a novel sketch feature representation learning method named SketchNet based on the deep convolutional neural network to address the problem of sketch classification. (2) To learn the discriminative feature representation, we propose to construct the image pairs to discover the shared structure between sketch and real images. (3) Our method boosts the benchmark of sketch classification, achieving the state-of-the-art performance in terms of classification evaluation metrics. (4) We have collected a new real image dataset which includes the corresponding real images of TU-Berlin sketch benchmark [12]. This dataset will be released at the author’s websites.

2. Related work

In this section, we review the most related work on sketch classification. There has been a large body of re-

search on analyzing sketch classification. Most of the works [13, 12, 29] firstly extract the low-level features from the sketch image, and then model the sketch representation by using bag of feature model. These methods are limited to a specific class of sketch with small variations, with similar shape appearances. Moreover, in [21] the authors proposed an ensemble matching method for sketch recognition. This method is firstly construct the star-graph with bag of features representation, then employ the SVM to train the category classifiers. However, this method is still sensitive to the variations of the sketches. On sketch retrieval, Eitz et al. [14] propose to extract several kinds of descriptors to construct the bag of features model, and then evaluate the performance of these descriptors on a large scale dataset. While Cao et al. [2] propose a flip invariant feature representation named FISH for sketch retrieval by embedding the symmetry structure of the sketch. But all the existed methods are based on the traditional low-level features which are still sensitive to the sketch variations.

With the development of deep learning, we have witnessed the successful of applying the deep learning framework on image recognition and retrieval. The generative and discriminative power of deep features have been used to build deep generative model shapes [16, 18]. In [18], the authors propose to use DBN to generate the hand-written digits, which has achieved notable performance. While Es-lami et al. [16] develop an object shape model method by using the boltzmann machine. These methods are able to efficiently handle the intra-class variations. Moreover, Wu et al. [35] propose a novel shape representation named 3D shapeNets, which focuses on modeling the 3D shapes. In this method [35], the authors propose to recognize the object from the depth images and then construct a 3D shapenets to capture the structure of 3D shapes. Different from the existing works based on deep learning on modeling the object shape, our method is focused on the sketch image which has large intra-class variations and large inter-class similarity. Another related work is Siamese network [8] which has been widely used in text classification [33] and speech feature classification [6]. This network contains two identical sub-convolutional networks shared the weighting parameters. The goal of this network is to make the output vectors similar when the input pairs have the same label, and dissimilar for the input pairs that are labeled dissimilar.

3. SketchNet

In this section we introduce our proposed SketchNet on the task of sketch classification. The training sketch image is represented as $\mathbf{S} = \{s_1, \dots, s_N\} \in \mathbb{R}^{1 \times N}$ where N denotes the number of training sketches. The corresponding real images dataset is denoted as $\mathbf{R} = \{r_1, \dots, r_M\} \in \mathbb{R}^{1 \times M}$ where M is the number of real images. The category label of these real images are $\mathbf{T} = \{t_1, \dots, t_N\} \in \mathbb{R}^{C \times M}$

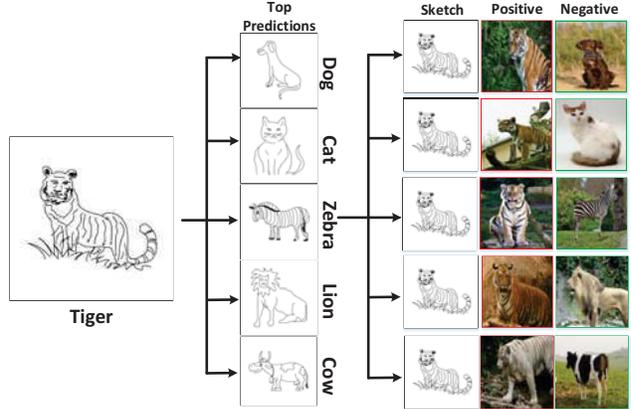


Figure 3. Triplet generation for each training sketch. Given the sketch image, top 5 category predictions (dog, cat, zebra, lion and cow) are extracted based on the pre-trained sketch model. For each sketch image, we find one positive real image from the corresponding category real image set and one negative image from the predicted category set.

where C represent the number of categories.

3.1. Triplet construction by embedding real images

Our goal is to discover the shared structures between sketch images and real images, and then based on the shared feature representation learn a model to predict the label for any new sketch images. This is a challenge problem: the real images are not provided any localized information of objects so we must automatically discover the relevant regions in each images that correspond to the sketch image. Moreover, the appearance of these real images may change drastically caused by views, occlusion and scales. To address above-mentioned problems, we exploit the fact that for each sketch image, the nearest real images which are in the same category would have the coherent appearance or changing smoothly. To this end, each sketch image is matched to its most similar real images from the auxiliary repository to construct the image pairs. After that, we develop a triplet from these image pairs which is composed of the sketch image, the same category real image (positive image) and different category real image (negative image). However, considering the ratio between relevant images and irrelevant ones, it would generate a large number of negative pairs which makes the training data be computation inefficient and unbalanced.

Considering that problem, Alexnet [20] is firstly trained by mixing the sketches and real images as the training data. We follow the fine-tune process to train this sketch model which initialize the Alexnet with the pre-trained model on ImageNet [11]. Specifically, we set the number of categories to be 250 and the softmax loss function is employed. The learning rate is initialized as 0.001. When the sketch model is obtained defined as \mathcal{F} , we extract the prediction

Table 1. Configurations of the SketchNet with input images of size 256×256

Net	Type	Filter Size/Stride	Output Size
S-Net	S_{cov1}	$11 \times 11 / 4$	$96 \times 54 \times 54$
	S_{Pool1}	$3 \times 3 / 2$	$96 \times 27 \times 27$
	S_{cov2}	$5 \times 5 / 1$	$256 \times 27 \times 27$
	S_{Pool2}	$3 \times 3 / 2$	$256 \times 13 \times 13$
	S_{cov3}	$3 \times 3 / 1$	$384 \times 13 \times 13$
R-Net	R_{cov1}	$11 \times 11 / 4$	$96 \times 54 \times 54$
	R_{Pool1}	$3 \times 3 / 2$	$96 \times 27 \times 27$
	R_{cov2}	$5 \times 5 / 1$	$256 \times 27 \times 27$
	R_{Pool2}	$3 \times 3 / 2$	$256 \times 13 \times 13$
	R_{cov3}	$3 \times 3 / 1$	$384 \times 13 \times 13$
	R_{cov4}	$3 \times 3 / 1$	$384 \times 13 \times 13$
C-Net	C_{cov1}	$3 \times 3 / 1$	$384 \times 13 \times 13$
	C_{cov2}	$3 \times 3 / 1$	$384 \times 13 \times 13$
	C_{cov3}	$3 \times 3 / 1$	$256 \times 13 \times 13$
	C_{Pool1}	$3 \times 3 / 2$	$256 \times 6 \times 6$
	f_{c_a}	$1 \times 1 / 1$	$4096 \times 1 \times 1$
	f_{c_b}	$1 \times 1 / 1$	$4096 \times 1 \times 1$

score of each sketch denoting as $\mathbf{P}_i \in \mathbb{R}^{1 \times C}$. We sort the prediction scores and extract the top K incorrelated categories \mathbf{P}_i^{top} to construct the negative image pairs. In all our experiments, we set the $K = 5$ by considering the computation efficiently and the performance of classification model.

As discussed aforementioned, a triplet is composed of the sketch images, positive and negative real images. To obtain the positive real images for each sketch, we use the nearest neighborhood methods as Eq. 1 to extract the top 5 similarity real images from the positive training image set. In the same method, we also extract the top 5 similarity real images from each negative category training set. In total, there are 5 positive real images and 25 negative real images which would generate about $C_5^1 \times C_{25}^1 = 125$ triplets for each training sketch images. In the following all the experiments, we apply this method to generate the training data.

$$\{r_1^i, \dots, r_j^i\} = \underset{\mathbf{t}_j \in \mathbf{P}_i^{top}}{\operatorname{argmin}} \|\mathcal{F}_{f_{c_7}}(s_i) - \mathcal{F}_{f_{c_7}}(r_j)\|_2, \quad (1)$$

where $\mathcal{F}_{f_{c_7}}(\cdot)$ denotes the features of the layer f_{c_7} in Alexnet. $\|\cdot\|_2$ represents the Euclidean distance. $\{r_1^i, \dots, r_j^i\}$ represents the related training images for s_i .

3.2. SketchNet architecture

There exists significant appearance gap between sketch images and real images, thus the traditional neural network can not be directly transferred to our problem. To handle those differences, we customize the AlexNet [20] to develop a novel neural network.

Specifically, our proposed SketchNet is composed of **S-Net**, **R-Net** and **C-net**. **S-Net** is defined to extract the fea-

tures from the sketch images. With the consideration of limited information on the sketch images, we set three convolution layers and two pooling layers as shown in Fig. 4 in the green bounding box. Similarly, **R-Net** is used to extract the features from positive and negative real images. To achieve the relative similarities, R-Net should be able to extract different features from positive and negative images. Furthermore, one more convolution layers are added on the R-Net comparing with S-Net. There are four convolution layers and two pooling layers as shown in Fig. 4 in the blue bounding box. The reason is that in the step of loss back propagation, sketch images should be more sensitive to the loss. **C-Net** is developed to merge the feature maps between sketch and real images. We concatenate the layer of " R_{cov4} " and " S_{cov3} " as the input of C-Net. Moreover, two full connected layers are added in this net. Thus, three convolution layers and one pooling layers constitute the C-Net as shown in Fig. 4 in the red bounding box. We also make the C-Net in the structure of Siamese architecture [8] to let positive pair have a higher classification score comparing with the negative ones. The overall architecture between layers are shown in Fig. 4. Furthermore, the specific parameters of each layer are displayed in Table 1.

The reasons to choose three nets to form the SketchNet are: firstly, the corresponding real images of each sketch include more appearance and color information than sketch images, we need to use different number of convolution layers to extract their features to narrow the gap between sketches and real images. Moreover, from the loss back propagation perspective, sketch images should have more impact on the loss than real images to determine the common latent structures. Secondly, considering that positive and negative real images should have distinct structures with the sketch image, we design their nets sharing the same parameters to make sure that positive image pair could generate the higher scores in the prediction. Last but not the least, to adopt our task of sketch classification, we design the novel loss function to discover the discriminative features of different training pairs. Specifically, we need the learned features to be able to get smaller intra-class variances and bigger inter-class differences.

3.3. Loss function

We propose to use different loss functions to adopt to different tasks using our SketchNet. For sketch classification, we develop a novel loss function which is composed of two components. The first component is the traditional loss function softmax which has demonstrated its strong ability on image classification. In our neural network, we use this loss function to achieve the prediction score of the input images. Specially, we can use this loss function to obtain the initialization prediction results on positive pair and negative

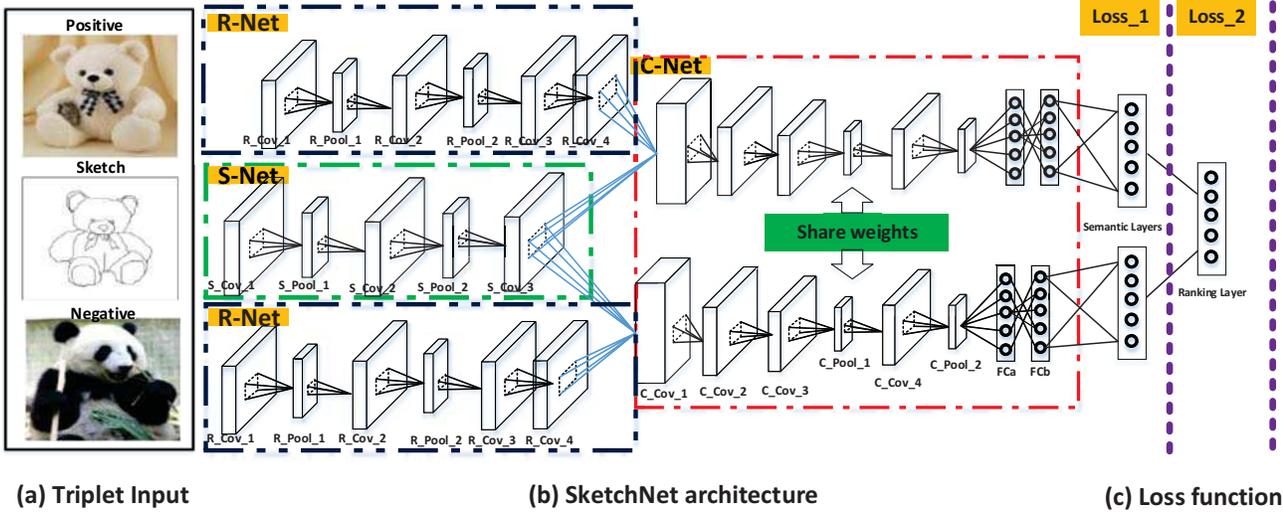


Figure 4. **Architecture of our SketchNet.** As described in Section 3.2, there are three components **R-Net** (in the blue box), **S-Net**(in the green box) and **C-Net** (in the red box) to construct our neural network. Given the training triplet, we apply the sketch into the S-Net and real images into the R-net. In particular, positive and negative images share the same R-net. Then, the positive and negative sketch pairs are fed into the C-Net. Finally, we use our defined loss function to back propagation.

pairs. The definition of softmax is following:

$$L_c(x^i, y^i, W_c) = -\log P(y^i = k | x^i, W_c) \quad (2)$$

$$= -\log \frac{e^{-f^k(x^i, W_c)}}{\sum_{l=1}^C e^{-f^l(x^i, W_c)}}$$

where x^i represent the input image data and y^i denotes the label of the input image x^i , and C is the number of sketch categories. k is the category label of current input image and W_c denotes the weights in the fully connected layers which is used to map the extract deep features to the labels.

Based on the softmax loss function we can separately extract the predictions of positive and negative pair images. However, there do not exist evidence that the score of positive pair could be higher than the negative ones. To this end, we introduce the second loss function which is based on the prediction scores of the positive and negative image pairs. In this loss function, we constrain the score of correct category label should be higher than all the other labels, which could be seen as a ranking loss. We set \mathbf{p}_+ as the prediction score of positive image pair and \mathbf{p}_- as the prediction score of negative pair. Then the loss function is defined as:

$$L_r(\mathbf{p}_+, \mathbf{p}_-, y^i) = \max(0, 1 - (\mathbf{p}_- - \mathbf{p}_+)) \quad (3)$$

This loss function is the variation of ranking loss which is the convex.

Finally, the loss function of our proposed SketchNet for sketch classification is:

$$L_{SketchNet} = L_r + \lambda * L_c \quad (4)$$

where λ is the weights parameters to balance the values of two components in the objective function. Moreover, the

loss function is convex function which can be easily computed its gradients.

3.4. Testing of SketchNet

In the test time, we have obtained the SketchNet model to extract the sketch feature representations. However, the category of test sketch images are unknown, we can not develop the triplet as the training time. Different from the training process as shown in Fig. 4, in the test time, our SketchNet shrink to contain One R-Net, S-Net and One C-Net as shown in Fig. 2. So, a sketch image is firstly fed into the pre-trained Alexnet model to achieve the top 5 prediction labels. After that, we extract 5 similar real images for each prediction label from auxiliary repository to develop the sketch pairs. In total, each test sketch image would generate 25 image pairs. One of the test pairs on sketch classification are shown in Fig. 5. Afterwards, we can adaptive the pre-trained sketchnet model to extract augmented feature representation of test sketch images. The feature maps of each layer could be used as the feature representation of test images.

The straight way to get the prediction results of the test sketch is to sum the feature maps of the last layer of test image pairs. However, this would generate the biased results since in the training time we only use a small of image pairs comparing to the possible combinations. We need to learn a metric that could make the positive pairs achieve a higher prediction scores on the correct label than the negative ones. Specifically, A validation image set composed of sketch images and all the auxiliary real images is extracted from the training part. Similarly, we generate the test pairs which

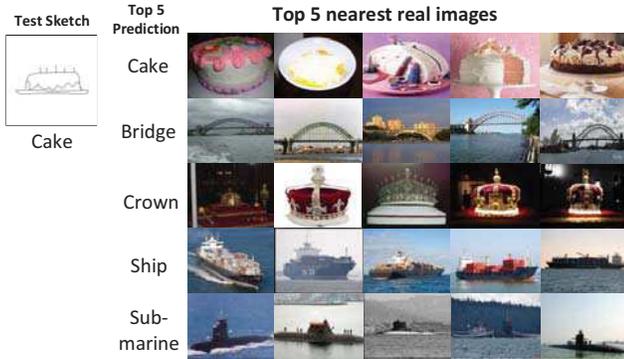


Figure 5. Examples of generating test pairs. The first column image shows the test sketch image “cake”. The top 5 prediction results are displayed in the second columns and the rest columns of images correspond to the related real images which are used to construct the test pairs.

contains the sketch and its nearest real images. Then, the feature map is constructed based on the prediction scores of each test pair. After that we apply the metric SVM [26] based on the feature maps of full connected layers (FC_b) for each image pairs to compute the metric weighting. Finally, the prediction score combining with the learned metric parameters are used to propagate the label of test images.

4. Experiments

In this section, we conduct extensive evaluations of the proposed method on the task of sketch classification. We implement the proposed method based on the opensource Caffe[19]. We change sketch images from one channel to three by copying themselves three times. In all experiments, our networks are trained by stochastic gradient descent with 0.9 momentum. We initiate learning rate to be 0.0001 and decrease it by 0.1 after finishing about 30 epochs. The weight decay parameter is 0.0005. The training time is proportional to the total number of triplets and the number of epochs. Overall training it takes approximately 2 days based on a PC with 2.8GHz CPU and GTX TITAN Black GPU. The mini-batch size of images is 64. Considering the total number of triplets is very large, the training time is sensible.

The ImageNet ILSVRC-2012 dataset [27] is utilized to pre-train the CNN model by optimizing multinomial logistic regression objective function in the image classification task. This dataset contains about 1.2 million training images and 50,000 validation images, roughly 1,000 images in each of 1,000 categories. We use the pre-trained parameters of convolutional layers and fully-connected layers to initialize the CNN part of SketchNet in our method.

4.1. Dataset

TU-Berlin sketch benchmark¹ [12] is proposed for sketch classification and recognition. There are 250 object categories in this dataset, which cover mostly daily objects in the life, *e.g.* car, horse. For each category, 80 sketch images are collected without any common templates and each sketch was defined by the person drawing it. After the dataset was constructed, there was a phase where humans tried to recognize the sketches and the mean recognition accuracy over all the 250 categories is 73%.

Furthermore, we also propose a new sketch dataset which extends the TU-Berlin sketch benchmark by introducing the real images for each category. This extended dataset can be used for sketch classification and retrieval. To construct this dataset, for each sketch category we find its corresponding class in the ImageNet and directly collect these real images to extend. While for these categories which are not overlapping with ImageNet, we collect the real images by using Google image search based on their category labels. Finally, we collect 191,067 real images in total, and average 764 images for each category. To guarantee the wide variety of real images, we do not add any constraints on collecting the real images.

Evaluation criteria: In our experiment, we use the aforementioned dataset and measure the performance using the following criteria: we use the average precision (AP) to calculate the classification accuracy for each category. Then to compare with other existing work on sketch classification, Mean Average Precision (MAP) is computed.

4.2. Sketch classification

In this section, to validate the effectiveness of our SketchNet, we show the sketch classification results on the TU-Berlin sketch benchmark. We firstly split this dataset into training and testing parts. Following the related work [12], 9 kinds of training splits are collected by selecting different number of sketch images for each categories which are {8, 16, 24, 32, 40, 48, 56, 65, 72}. The training sketch images for each class are randomly chosen from the dataset. And the rest images of this dataset are used as the test. To train the metric parameters as discussed in Sec. 3.4, we randomly select 20% training data as the validation dataset.

Each training sketch image is firstly used to construct a triplet, and we would obtain 125 triplets for each sketch image. Similarly, the test sketch image is asked to construct the test pair by computing the similarity between sketch and real images. Specially, in the test time, we extract the top 5 prediction categories and for each category 5 related real images are extracted to construct 25 test pairs. All the parameters of SketchNet are defined as shown in Table 1. Furthermore, S-Net and R-Net is initialized based on the two

¹<http://cybertron.cg.tu-berlin.de/eitz/projects/classifysketch/>

Table 2. The comparison classification results on TU-Berlin sketch benchmark

Methods	8	16	24	32	40	48	56	64	72
SketchNet	58.04%	64.43%	67.89%	72.01%	73.54%	75.18%	76.08%	77.33%	80.42%
SketchNet(no metric)	55.69%	64.37%	66.20%	71.19%	69.57%	73.62%	73.43%	76.50%	77.41%
AlexNet(mixed real images)	51.96%	59.22%	63.80%	65.97%	68.58%	69.80%	70.46%	72.31%	73.25%
AlexNet [20]	54.8%	62.3%	67.6%	68.12%	69.86%	71.65%	72.62%	74.02%	75.02%
GoogLeNet [32]	52.01%	59.61%	62.45%	67.48%	69.19%	70.5%	71.5%	72.4%	75.25%
NIN [23]	51.4%	61.9%	65.50%	68.05%	70.61%	71.50%	72.02%	73.82%	74.40%
VGGNet [5]	53.85%	60.65%	63.05%	65.54%	67.34%	69.54%	73.83%	75.17%	76.53%
FisherVector size 24 (SP) [29]	43%	52%	56%	59%	62%	65%	66%	67%	68%
FisherVector size 24 [29]	41%	50%	53%	56%	60%	62%	64%	64%	65%
FisherVector size 16 (SP) [29]	44%	50%	55%	57%	60%	63%	64%	65%	66%
FisherVector size 16 [29]	39%	45.5%	50%	53%	56%	59%	60%	61%	62%
Eitz et al. [12] (SVM soft)	33%	41%	44%	46%	50%	51%	54%	55%	55%
Eitz et al. [12] (SVM hard)	32%	37%	42%	45.5%	48%	49%	50.8%	53%	53%
Eitz et al. [12] (Knn soft)	26%	31%	34.8%	36%	39%	40.5%	42%	43%	44%
Eitz et al. [12] (knn hard)	22%	26%	28%	31%	33%	34.5%	35%	36%	37.5%

different pre-trained sketch models.

We select two kinds of baselines to compare with our method. We firstly compare our method with these existing neural network which only use the sketch images as the training images. We also compare the propose SketchNet with other methods on sketch classification based on hand-crafted features [12, 29]. Specially, we select Alexnet [20], GoogLeNet [32], Network in Network [23] and VGGnet [5] as the baselines which use the deep features. And for these methods using shadow features, we select the methods proposed in [29] which is used the fisher vector as the sketch representation. Moreover, four kinds of feature representations are selected. We also select the method described in [12] which uses the soft mapping and hard mapping. Since there do not exist the open source code for these hand-crafted methods, we copy their reported classification results in the corresponding papers [29]. While for the deep features, we implement their corresponding neural network by treating the sketch images as the training data with softmax as the loss function. Considering the limited number of training images, we adopt the fine-tune strategy to train the neural network based on the ImageNet model using Caffe. We also show the classification result without using metric learning (SketchNet no metric). We first sum the predictions of test pairs, and then find the maximal response label as the test image label.

The comparison results are shown in Table 2. Several observations can be drawn from the results. First of all, for all the cases, the proposed SketchNet consistently outperforms the baselines reaching **80.42%**. This can demonstrate that real images as the reference plays an important role on judging the category of sketches. Moreover, our proposed label propagation method can help making decisions. Secondly, we could find that these very deep methods e.g. GoogLenet

[32] and VGGnet [5] do not achieve much better classification accuracies comparing with Alexnet [20]. For example, when the number of training images for each category is 32, Alexnet obtains 68.12% comparing with GoogLenet up to 67.48% and VGGnet 65.54%. While the number grows to 72, they all get higher results than AlexNet. The main reason may be the limited number of training images on each category which may cause over-fitting. Thirdly, the classification performance is improving with the number of training images growing. Obviously, our SkechtNet can gain better performance even with smaller amount of training images as shown in Table 2. Furthermore, when the extra real image are added into the neural network, the classification accuracy is significantly improved. For example, for comparison the impact caused by introducing the real images as the context information, we can find that SketchNet obtain a higher accuracy comparing with AlexNet. Comparing with these baselines without discovering the correlation between sketch and real images, our proposed method based on triplets is able to effectively improve the classification accuracy. Comparing with Alexnet(mixed real images) whose training data are mixed of sketch and real images, our SketchNet achieves about **7%** improvement. It is interesting to observe that the overall performance of hand-crafted features on this dataset is worse than the results based on the deep features. Last but not the least, the performance of using deep features is beyond human with enough training images. For example, for our SketchNet, we achieve 73.54% when the number of training images for each category is 40. While for VGGnet the number of training images is 56. This can further demonstrate the representation ability of learned deep features.

We believe three possible reasons may explain such observations: (i) Similar with the human to recognize the con-

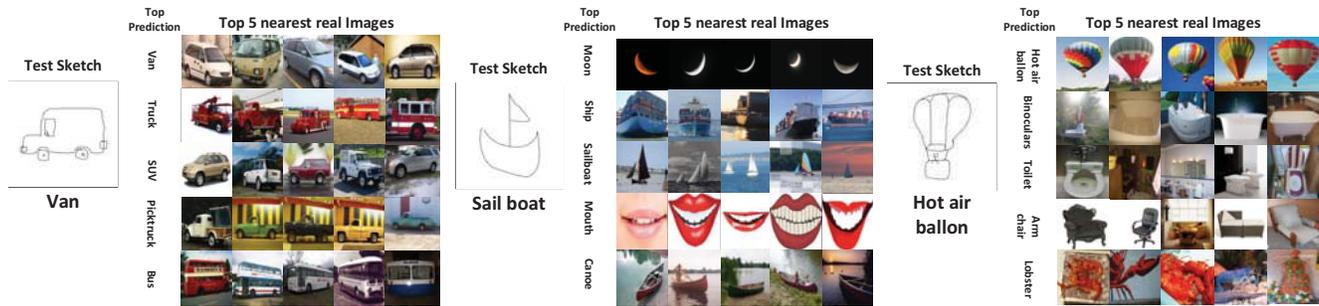


Figure 6. Examples of correctly classified sketches on test case with the real images as the references. The first column shows the test sketch, and the top 5 category predictions are displayed from the top row to the bottom in sequence. Specifically, each row of real image represents the nearest visual similarity images in their category.

tent of sketch images, our proposed SketchNet can imitate this process by introducing the most similarity real images. (ii) With our designed loss function, our SketchNet can not only discover the latent structure between sketches and real images, but also obtain the discriminative feature representation. The second component of our proposed loss function can be seen as ranking regularization to serve as feature selection. (iii) As the experiment results show, deep features have a significantly discriminative ability than the traditional hand-crafted feature representation.

4.3. Discussion

Although the overall sketch classification performance can be boosted by introducing the real image as the context, there are still existing some limitations influencing the performance of specific categories. Some results are displayed in Fig. 6 and Fig. 7. As shown in Fig. 7, we can observe that because of the drawing perspective, the sketch “cake” is more similar with “submarine” than a cake. The second factor is the real images in the repository, which need cover the object in multi-views. In this case, we can not find the same view perspective real images from the cake image set. This demonstrates that the repository should include real images of different views. Last but not the least, the aspect ratio of sketch images also contain important categorical information. However in our work we ask the images show be fixed to 256×256 which lead to some unexpected situations. As shown in Fig. 7, the top five prediction includes trombone which is dissimilar with the query sketch. Furthermore, from the classification results we can observe that Human can recognize almost sketches without sketch training data. While for computer, with enough training sketches it can achieve much better classification accuracy even beyond human on sketch classification.

5. Conclusion

In this paper, we have proposed a novel method to develop sketch feature representation, named SketchNet based



Figure 7. Examples of incorrect classification case based on our SketchNet. The first column shows the test sketch image “cake”. The top 5 prediction results are displayed in the second columns and the rest columns of images are shown the related real images which are used to construct the test pairs.

on the deep convolutional neural network. We use the real web images as the reference to discover the latent discriminative structures of sketch images. To that end, a triplet is constructed as the input of our SketchNet. Moreover, we customize the novel neural network with our defined loss function to extract the discriminative feature representation of sketch images. In the step of testing, we introduce a metric learning method to merge the scores of test pairs. Empirical evaluations on sketch classification show that the proposed method has achieved superior performance gains over state-of-the-arts. Nevertheless, there exist many other factors e.g. view perspectives which influence the performance of classification. This can be solved by introducing the 3D shape models which would be our future work.

6. Acknowledgement

This work was supported by National Natural Science Foundation of China (No.61422213 and U1536203) and “Strategic Priority Research Program” of the Chinese Academy of Sciences (XDA06010701).

References

- [1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE TPAMI*, 24(4):509–522, 2002.
- [2] X. Cao, H. Zhang, S. Liu, X. Guo, and L. Lin. Sym-fish: A symmetry-aware flip invariant sketch histogram shape descriptor. In *ICCV*, 2013.
- [3] A. Chalechale, G. Naghdy, and A. Mertins. Sketch-based image matching using angular partitioning. *IEEE TSMC, Part A*, 35(1):28–41, 2005.
- [4] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.
- [5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.
- [6] K. Chen and A. Salman. Extracting speaker-specific information with a regularized siamese deep network. In *NIPS*, 2011.
- [7] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: internet image montage. *ACM TOG*, 28(5):124, 2009.
- [8] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.
- [9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
- [10] D. Dixon, M. Prasad, and T. Hammond. icandraw: using sketch recognition and corrective feedback to assist a user in drawing human faces. In *SIGCHI*, pages 897–906, 2010.
- [11] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.
- [12] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? *ACM Trans. Graph.*, 31(4):44–52, 2012.
- [13] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. A descriptor for large scale image retrieval based on sketched feature lines. In *Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pages 29–36, 2009.
- [14] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE TVCG*, 17(11):1624–1636, 2011.
- [15] M. Eitz, R. Richter, K. Hildebrand, T. Boubekeur, and M. Alexa. Photosketcher: interactive sketch-based image synthesis. *IEEE CGA*, 31(6):56–66, 2011.
- [16] S. A. Eslami, N. Heess, C. K. Williams, and J. Winn. The shape boltzmann machine: a strong model of object shape. *IJCV*, 107(2):155–176, 2014.
- [17] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.
- [18] G. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.
- [19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [21] Y. Li, Y.-Z. Song, and S. Gong. Sketch recognition by ensemble matching of structured features. In *BMVC*, 2013.
- [22] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, J. Dong, L. Lin, and S. Yan. Deep human parsing with active template regression. *IEEE TPAMI*, 37(12):2402–2414, 2015.
- [23] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1409.1556*, 2014.
- [24] S. Liu, X. Liang, L. Liu, X. Shen, J. Yang, C. Xu, L. Lin, X. Cao, and S. Yan. Matching-cnn meets knn: quasi-parametric human parsing. In *CVPR*, pages 1419–1427, 2015.
- [25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [26] B. McFee and G. R. Lanckriet. Metric learning to rank. In *ICML*, pages 775–782, 2010.
- [27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.
- [28] R. Schmidt, B. Wyvill, M. C. Sousa, and J. A. Jorge. Shapeshop: Sketch-based solid modeling with blobtrees. In *ACM SIGGRAPH courses*, number 43, 2007.
- [29] R. G. Schneider and T. Tuytelaars. Sketch classification and classification-driven analysis using fisher vectors. *ACM TOG*, 33(6):174:1–174:9, 2014.
- [30] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. 30(6):154, 2011.
- [31] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, pages 945–953, 2015.
- [32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.
- [33] W. tau Yih, K. Toutanova, J. C. Platt, and C. Meek. Learning discriminative projections for text similarity measures. In *CCNLL*, 2011.
- [34] F. Wang, L. Kang, and Y. Li. Sketch-based 3d shape retrieval using convolutional neural networks. In *ICCV*, pages 1875–1883, 2015.
- [35] Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao. 3d shapenets for 2.5 d object recognition and next-best-view prediction. *arXiv preprint arXiv:1406.5670*, 2014.