Training Convolutional Neural Network for Sketch Recognition on Large-Scale Dataset

Wen Zhou[1] and Jinyuan Jia[2]
[1]School of Computer and Information, Anhui Normal University, China
[2]School of Software Engineering, Tongji University, China

Abstract: *With the rapid development of computer vision technology, increasingly more focus has been put on image recognition. More specifically, a sketch is an important hand-drawn image that is garnering increased attention. Moreover, as handheld devices such as tablets, smartphones, etc. have become more popular, it has become increasingly more convenient for people to hand-draw sketches using this equipment. Hence, sketch recognition is a necessary task to improve the performance of intelligent equipment. In this paper, a sketch recognition learning approach is proposed that is based on the Visual Geometry Group16 Convolutional Neural Network (VGG16 CNN). In particular, in order to diminish the effect of the number of sketches on the learning method, we adopt a strategy of increasing the quantity to improve the diversity and scale of sketches. Initially, sketch features are extracted via the pretrained VGG16 CNN. Additionally, we obtain contextual features based on the traverse stroke scheme. Then, the VGG16 CNN is trained using a joint Bayesian method to update the related network parameters. Moreover, this network has been applied to predict the labels of input sketches in order to automatically recognize the label of a sketch. Last but not least, related experiments are conducted, and the comparison of our method with the state-of-the-art methods is performed, which shows that our approach is superior and feasible.*

Keywords: *Sketch recognition, VGG16 convolutional neural network, contextual features, strokes traverse, joint Bayesian.*

Received September 5,2017;Accepted April 28, 2019
https://doi.org/10.34028/iajit/17/1/10

1. Introduction

With the increasing popularity of mobile devices and tablets, it is becoming increasingly more convenient for people to hand draw sketches using this equipment. Hence, a great deal of sketch data exists on the internet. In addition, sketch-based 3D shape retrieval has become a research hot topic that has attracted many researchers to this area. Furthermore, as a special type of image data, sketches always appear vague and are difficult to understand because sketches are often arbitrarily drawn and abstract. Therefore, it is not always very easy for people to thoroughly understand an abstract sketch. In some cases, we can view inscriptions on bones as a sketch. As a special type of image, sketches may be dull, lack sufficient information, and may only be comprised of simple white-and-black lines. Furthermore, the quality of a sketch mainly depends on the skill of the draftsman, which leads to huge differences, even if the same object was drawn. In other words, sketch recognition is a very arduous task; moreover, human recognition of sketches often contains many errors. This situation shows that sketch recognition is not easy. This is due to many factors:

1. Sketches are highly iconic and abstract, e.g., human Figures can be described as stickmen.
2. The same object can be drawn with hugely different details, which is the nature of hand-drawn sketches, e.g., a human figure sketch can be either a stickman or a portrait with details depending on the drawer.
3. Sketches lack visual cues, i.e., they are comprised of black and white lines instead of colored pixels.

Nevertheless, Convolutional Neural Networks (CNNs) have achieved tremendous success recently in replacing hand-crafting feature representation with representation learning for a variety of vision problems. However, existing CNNs are primarily designed for images, and if we are directly employing them for sketch-based research, there would be little improvement over hand-crafted features. In fact, CNNs typically require large amounts of training data to avoid over fitting, given the millions of model parameters. However, the existing hand-drawn sketch datasets, including the largest TU-Berlin dataset, are far smaller than the image datasets that are typically used for training CNNs.

Therefore, in order to better train our samples, in this paper we present an increasing quantity strategy, i.e., some operations on sketches are completed to increase the diversity of existing sketches. Our definitive goal is to enlarge the scale of sketches, which will allow us to avoid the over fitting of the training network to some extent.

Our mainly contribution can be concluded as follows.

1. We proposed two different kinds of methods to increase the scale of sketches. To some extent, we

can better train our network.

2. We presented a contextual feature that is extracted via traversing strokes, which is helpful in training our network. In particular, it can achieve the goal of multi-feature learning.
3. We utilized a joint Bayesian metric to measure the similarity and update the parameters of the network. In the testing stage, it can improve the accuracy of the predicted result.

The remainder of this paper is organized as follows, in the section 2, the related works will be presented, in the section 3, the overview of proposed framework is presented, the detail of our proposed method will be shown in section 4. In the section 5, we perform related experiments to validate our methods, besides, the comparison between ours and the-state-of-the-arts approaches also is conducted in this section. Finally, in the section 6, the conclusion is drawn and the future works also is proposed.

2. Related Works

2.1. Sketch Recognition

Lu *et al.* [10] Jabal *et al.* [6] and Zitnick *et al.* [21] performed deeply research workson sketch recognition with professional CAD or artistic drawings as input. Recently, Eitz *et al.* [4] proposed a new framework to learning sketch recognition. Besides, an open source sketch dataset is released to help others to training own samples. Sequentially, Schneider *et al.* [15] adopt Support Vector machine as the classifier and differ only in what hand-drawn features borrowed from images are used as representation. Yi *et al.* [18] illustrated that fusing different local features using multiple kernel learning helps to improve the performance of sketch recognition. They also examined the performance of many features individually such as, Speeded Up Robust Features (SURF) [1], and found the fact that Histogram of Oriented Gradients (HOG) features generally are believed as the best feature descriptor for the sketch. On the other hand, CNN had recently achieved impressive performance for many recognition tasks across many different disciplines. In particular, Convolutional Neural Network (CNN) have dominated top benchmark results on visual recognition challenges. An important advantage of CNN, compared with conventional classifier such as SVMs, lies with the closely coupled nature of presentation learning and classification, which makes the learned feature representation maximally discriminative. Hence, Simonyan and Zisserman [16] proposed a deeper network with smaller filters are preferable for images recognition. Despite these advances, most existing image recognition CNNs are optimized for images, not for sketch. Ultimately make them conduct sub-optimally on sketches. However, Yu *et al.* [19, 20] propose new methods using CNNs to obtain better recognition performance, even beats the human. It makes a huge breakthrough for sketch recognition. Wang *et al.* [17] design a framework for cross-domain matching, it uses a variant of Siamese network where the shape view images branch and sketch branch have the same architecture without any special treatment of the unique sketch.

In this paper, we show that directly using successful sketch-oriented CNNs to sketches leads to little improvement over hand-drawn feature-based methods. Our changes mainly are not only depending on geometry feature extracted from sketches, but also relying on the contextual information in each sketch. Although the training time will become more longer, the result also obtain advancement. Besides, multi-features fusion approaches have been adopted to solve the problem of computer vision, such as face recognition [14].

2.2. Sketch Datasets

To explore how human draw sketches and human sketch recognition, Eitz *et al.* [4] collected human-drawn sketches and named it as TU-Berlin sketch dataset, which is the largest and now the most commonly used human sketch dataset. It contains 250 categories with 80 sketches per category. It was collected on Amazon Mechanical Turk from 1,350 participants, thus providing a diversity of both categories and sketching styles within each category. It is exhaustive in terms of the number of object categories. More specially, it avoids the bias issue since they collect the same number of sketches for every class and the number of sketches for one class is also adequate for a large-scale retrieval benchmark.

Besides, Li *et al.* [7] aim to building the sketch-based 3D model retrieval benchmark, the SHREC12STB datasets are presented, it is based on Princeton Shape Benchmark (PSB) datasets, which has found 1258 relevant models for 90 of the total 250 classes from the PSB benchmark.

2.3. Deep Learning

CNNs trained on the large datasets such as Image Net have been shown to learn general purpose image descriptors for a number of vision tasks such as object detection, scene recognition, texture recognition and fine-grained classification. Donahue *et al.* [3] proposed a very deeper CNN to learn the feature, and then to classify the data. This method gains a very good classification result and the related network structure is used many other's researcher. Besides, Girshick *et al.* [5] also use CNN to perform object detection and acquire very good result.

Razavian *et al.* [13] extract the relative features to conduct object recognition, the result shows that the approach of using CNNs is obviously superior to others. Visual Geometry Group (VGG) 16 CNN [16] had been widely used to conduct classification task in many different domain, in this paper, this learning model is used to obtain related sketch features, in this way, we can collect correct representation for sketch.

3. The Overview of Proposed Approach

In this section, we present our proposed approach. Our method can mainly be divided into two parts. One is the feature extraction pipeline based on the pretrained CNN. To increase the scale of samples, we adopt two different methods, including sketch deformation and stroke removal. The other part is the extraction of the contextual features. According to the key point of sketch strokes, we build a graph structure and then traverse it according to the fixed number of steps. Traversing the length of the fixed number of relationships is used as a contextual feature. Finally, the joint Bayesian model is used to assess the relationship of multiple features. In fact, the joint Bayesian metric was proposed by Chen *et al*. [2] to validate the similarity in facial recognition. In this paper, we also adopt this method to measure the relation between many different feature vectors. The overview of our proposed approach is shown in Figure 1.

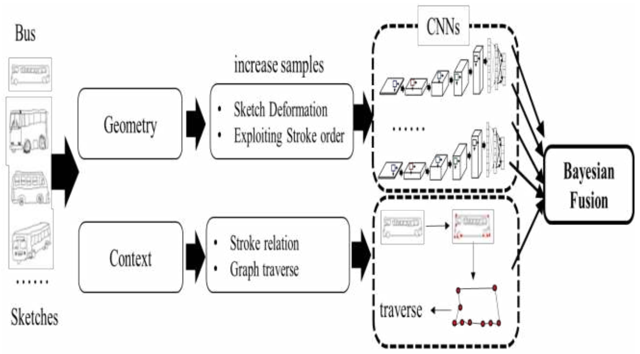

Figure 1. The overview of our proposed methods.

4. The Description of Proposed Framework

4.1. Exploring Stroke Order

Our aim is to increase the scale of sketches by removing unimportant strokes. More specially, this method is precisely the clever use of people to draw sketches of habits. People always draw long strokes and draw short strokes [4] In a certain sense, you can calculate the length of the strokes to determine the order of sketches. In general, a sketch is an ordered list of strokes, some of which convey broad aspects of the sketch, and others convey fine details. In order to obtain the strokes of sketch.

Recently Ma *et al*. [11] and Liang *et al*. [8] proposed key point-based approaches, respectively, such as, Difference of Gaussian (DoG) [9], Hessian operator [18] and Harris-Laplace detector [12] are fit for line drawing feature extraction. Harris corner detector is employed to obtain strokes. Given a sketch$\mathcal{s}_x$, we compute the Harris key points of it at first. A set of stroke S_x s are built as shown in Figure 2.

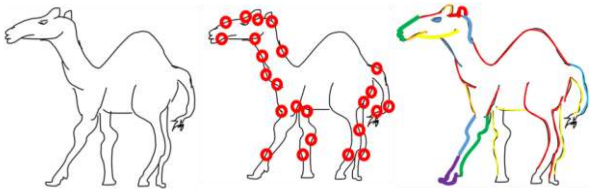

Figure 2. The process of a set of strokes are built.

In order to obtain the strokes of a sketches, we uniformly sampling N points (points set P) in unregular curves of a sketch. The angular of every two sampling points is computed, we assume $\forall p_i \in P, p_j \in P$,their corner cuts can be denoted as the term θ_i, θ_j respectively.Two connecting lines can be obtained between $\theta_i, \theta_j$and the center point of a sketch, then, the included angle $\theta_{ij}$of two connecting lines can be easily acquired. We can ceaselessly merge these sampling points until they meet following Equation (1).

$$\phi(\theta_i, \theta_j) = \cos(\theta_i - \theta_{ij}) \cos(\theta_j - \theta_{ij}) < \varepsilon \qquad (1)$$

In Equation (1), the threshold εis an experiments value, in this paper, we set $\varepsilon = 0.5$. Finally, we can obtain strokes set $\mathcal{S}_x$of a sketch after these sampling points.

In detail, given a sketch consisting of a set of N order strokes $\mathcal{S}_x = \{i \in N | \mathcal{s}_i\}$, the order of the stroke and its length are used together to compute the probability of removing the i^{th} strokes as Equation (2):

$$\Pr(\mathcal{s}_i) = \frac{e^{\alpha * count(h_i) - \beta * l_{h_i}}}{\sum_i e^{\alpha * count(h_i) - \beta * l_{h_i}}} \qquad (2)$$

Here, the term$h_i$represents the set of Harris key points in the i^{th} strokes, the term count (.) is to compute the number of Harris key points. In addition, the terms α, β are the experimental value, They mainly assure that the numerator or denominator is meaningful (i.e.,$e^{\alpha * count(h_i) - \beta * l_{h_i}} > 0$), in this paper, $\alpha = 1, \beta = 100$. Moreover, the term l_{h_i} is the length of thei^{th} stroke. Therefore, through Equation (2), we are able to remove strokes with the largest probabilities, resulting in new sketches, which would greatly increase the number of our sketch samples for us to raise more learning samples. The process result will be shown in Figure 3.

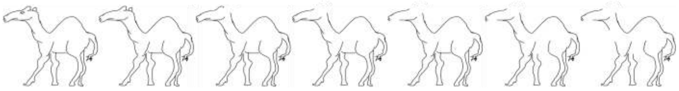

Figure 3. The process of new sketch generated by means of removing strokes.

4.2. Sketch Deformation

Sketch deformation is another important method to increase the number of sketches. The deformation process mainly consists of two approaches: local deformation and global deformation. More specifically, local deformation mainly moves the positions of the key points or controls the positions and then moves the total spline curve. In this case, the stroke lines can be regarded as cubic spline curves. Therefore, we can obtain new local strokes to increase the number of sketches. Global deformation is common in the process of image transformation, including sketch rotation. In this paper, we mainly apply sketch rotation to acquire the global sketch deformation

Initially, we collect the key points on stroke lines that are based on the Harris corner points. Here, let the term $\mathcal{K}_x = \{i \in x|k_i\}$ represent the key points set of sketch strokes, and then the cubic spline curves can be represented as follows.

$$\mathcal{S}_i(x) = (1-x)^3k_0 + 3(1-x)^2xk_1 + 3(1-x)x^2k_2 + x^3k_3 \; 0 \leq x \leq 1 \tag{3}$$

Where, the term k_0 and k_3 are the two endpoints of each spline curve, respectively. In order to obtain new spline curve of stroke spline, we will move the key point in each spline. Assumed the term$\mathcal{K}_i$ is the key points set of the i^{th} strokes, and then for every key points$\vec{\mathrm{k}} \in \mathcal{K}_i, \overrightarrow{\mathrm{k}_x} \rightarrow \vec{\mathrm{k}}_x + \mathcal{I}\frac{1}{2\pi}e^{-\frac{x}{2\sigma^2}}$. Here, the term $\mathcal{I}$ is an identity matrix, the term σ set to 0.1 by experimentally, and the term $\overrightarrow{\mathrm{k}_x}$ represents the x-axis coordinate vector. The process of local deformation will be shown in Figure 4.

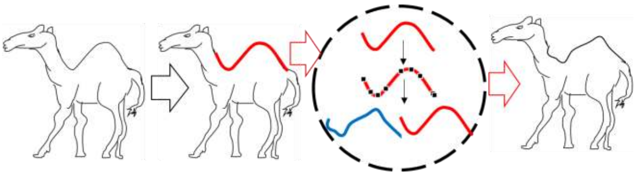

Figure 4. The process of local sketch strokes deformation.

On the other hand, the global deformation is sketch transformation, i.e., sketch rotation. First, we obtain the pivot position$\vec{\mathcal{P}}_i$ of the entire i^{th} sketch. Then $\forall \vec{\mathrm{p}} \in \mathrm{S}_i$, after rotation operation, the new point $\vec{p}_{new} = \begin{bmatrix} cos\omega & -sin\omega \\ sin\omega & cos\omega \end{bmatrix} * \vec{\mathcal{P}}_i + \vec{\mathrm{p}}$.

Where, we set the term$-\frac{\pi}{6} \leq \omega < \frac{\pi}{6}$ to guarantee the main structure can be identified. The process result will be seen in Figure 5.

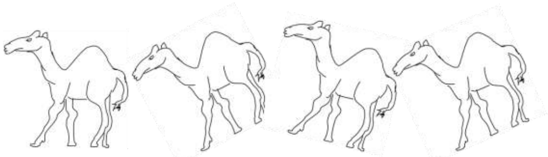

Figure 5. The process of global sketch strokes deformation.

4.3. VGG16 CNN

In this section, we mainly present VGG16CNN. Though above operations, the train sketches samples will scale up to huge. In the TU-Berlin sketch dataset [4], there only are 80 sketches in each category. A small number of sketches can lead to the network over fitting. Moreover, the parameters of CNN will arrive to over several million. It is difficult to calculate the correct fitting parameters completely using fewer samples. Fortunately, VGG16 CNN with pre-trained model can be used to extract the sketch features. In order to better representation for sketch, we need more kinds of different sketch, in this way, for a sketch, we can improve the robustness and correctness of features. In addition, an important reason is such, using CNN to learn a sketch is often worse than an image. By the above operations, we can increase the count of sketches in every category to 1000. Furthermore, we construct the three same structure CNNs, which train three kinds of different sketches, which are the sketch of removing part strokes, the sketch of local deformation, and global deformation, respectively.

However, VGG16 with pre-trained mainly is used to learning the images object, not sketches. Therefore, in order to better obtain the predicting result in test stage, we need to increasing update the parameters of network with pre-trained.

Before we train our samples, we firstly scaled the sketches from 1111×1111 to 256×256, the aim is to decrease the amount of calculation, because the sketch consisted of stroke lines rather than pixels.

The first two convolutional layers followed by a 2×2 pooling generate 64 response maps, each pooled to a size of 2×2, the 4096 features generated by the final fully connected layer are linearly transformed to 500×1 feature vector in the output layer. The Soft Max function is used in output layers; besides, the rectified linear units are used in all layers. The overview of our CNNs will be shown in Figure 6 and the detail of VGG16 structure is shown in Figure 7.

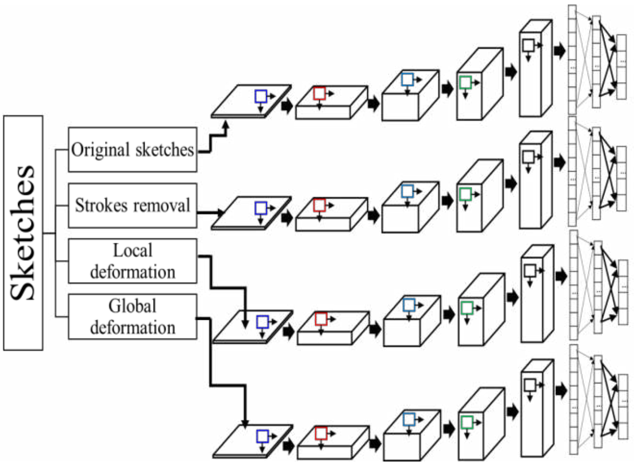

Figure 6. The overview of CNNs train sketches.

4.4. Contextual Features based on Strokes Traverse

In the above section, we have presented the method how to extract geometry features using CNNs. Besides, we enlarge the number of sketches by many different ways. However, sometimes, only geometry features can't obtain good result, above all, for the very abstract art sketches, it's hard to obtain better results. Therefore, the contextual features are extracted to better represent the contour of sketch. As the same as the above, we also collect the Harris corner key points $\mathcal{K}_i$ from the sketch S_i.

In order to take context message into account, we build a dual graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, for each sketch S_i. Every node in the term $\mathbb{V}$ represents a key point $k_i^s \in \mathcal{K}_i$. The two nodes are connected with an edge e ∈ $\mathbb{E}$, if their key point is spatially closest adjacent in the sketch.

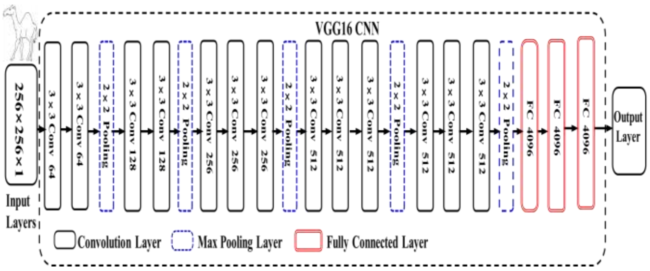

Figure 7. The structure of VGG16 CNN, it consists of 13 convolution layers, 4 max-pooling layers, 3fully connected layers.

Given the term Φ_s^n denotes a walk of length n starting at the node (i.e., a key point) k_i^s. Hence, we can define the similarity of two walks as follows.

$$d_{walk}^n(\Phi_s^n, \Phi_t^n) = \frac{1}{n+1}\sum_{m=1}^{n+1} e^{\frac{d(\,k_s^m, k_t^m)}{2\sigma^2}} \cos(\theta_s, \theta_t) \quad (4)$$

Here, the term d(·,·) represents the Euclidean distance between their normalized mean positions in the sketch (with the term σ=0.2) .Besides, the variable θ_x is the orientation of the strokes between the key point k_x and the key point K_{x+1}, the termk_s^m is the m^{th} node on the walk of the variableΦ_s^n.As a result, we can measure the context similarity of two nodes $k_i^s$and k_i^t by compare all the walks of them. This process can be formulated as follows.

$$d_{context}^n(k_i, k_j) = \frac{1}{|L_i^n|}\sum_{\{a|a \in L_i^n\}} \max_{\{b|b \in L_j^n\}} d_{walk}^n(a, b) \quad (5)$$

Here, the set$L_i^n$is a set of all walks of length n starting from node k_i^i and the value$|L_i^n|$ is the number of walks in the set L_i^n. For the setL_i^n is able to capture the context information of the node k_i^i. We call it as the context descriptor of the node k_i^i .Besides, by experimentally, we found that the length of walks should been between 3 and 5, which would provide good and stable result. Therefore, we set the length of walks n to 5. Moreover, though Equation (4), we can obtain the sketch context feature descriptor Fi of the sketch S_i.

$$\mathcal{F}_i = \begin{cases} d_{context}^5(k_i, k_j) if i \in N, j \in N and i \neq j \\ 0 \qquad\qquad if i \in N, j \in N and i = j \end{cases} \quad (6)$$

Here, the constant N is the number of key points in a sketch S_i.

4.5. Joint Bayesian Metric

In the above section, we have obtained geometry and contextual features, respectively. We denote these features as the term$\mathcal{F}_i^1, \mathcal{F}_i^2, \mathcal{F}_i^3, \mathcal{F}_i^4$, which are the i^{th} sketch S_i feature of strokes removing, local deformation, global deformation, and contextual feature, respectively. In particular, the similarity metric of different feature vectors is a very necessary step to improve the performance of sketch recognition. Chen *et al.* [2] proposed joint Bayesian method to test and verify the face features and reduce the separability between classes. Therefore, we adopt this method to find the inter-class relation.

$$\mathrm{r}(x_1, x_2) = \log\frac{P(x_1, x_2|H_I)}{P(x_1, x_2|H_E)} = x_1^T A x_1 + x_2^T A x_2 - 2x_1^T G x_2 \quad (7)$$

Where, the vector$x_1$represents geometry features, and the vectorx_2 is the contextual features. Let each x_1 represent the 4×500=2000D concatenated feature vector from our network ensemble. In order to better metric the features between geometry features and contextual features. The K-means methods are used to cluster the contextual features into 2000×1 feature vector, and then the size of vector$x_1$and vector x_2 is same.

The paramete A, G are negative semi-definite matrixes, whose value can be determined by learning from the data. Finally, we train the Joint Bayesian metric, thus learning a good metric that exploits intra-ensemble correlation. Note that in this way each feature dimension is fused together, implicitly giving more weight to more important features, as well as finding the optimal combination of different features of different models.

In particular, the joint Bayesian metric is used to measure the similarity relationship of multi-features, moreover, the joint Bayesian metric can update the parameters of VGG16 network, in this way, the back-forward scheme of CNN can improve the correctness of parameters, in this way, by training VGG16 network, the correct label name of input sketch can be predicted. In this way, we can achieve the task of sketch recognition.

5. Experiments

5.1. Environment Setting

The method presented in this paper is implemented mainly using Python programming language, and is executed on a PC under Windows 10, Intel core I7-7700HQ processor, and 8G memory size. The framework of CNN in this paper is Google Tensor Flow, which is an open-source, distributed, deep learning library on Python language. Moreover, in this section, TU-Berlin sketch dataset is employed to train and evaluate our proposed method.

In VGG16 CNN, the initial learning rate is set to 0.001, and mini-batch is 100. During training, each sketch is randomly cropped to a 256×562 sub-sketch. The Adam optimizer is adopted to optimize the network, in addition, the cross-entropy loss function is used to train the VGG16 network.

In the test stage, by training VGG16 CNN, we can obtain more correct network parameters, besides, the parameters of joint Bayesian also are increasing updated. Therefore, the network can correct predict the label of the input sketch, in this way, we can complete the task of sketch recognition.

5.2. Evaluation

We use the proposed method to enlarge sketches. In TU-Berlin dataset, there are 80 samples in each category, we increase the number to 1000 samples. We random collect 100 samples in each category as the test datasets. Next, we use Python Imaging Library (PIL) package in Python to scale the size of all sketches to 100×100.

Hereto, we have completed all the preparatory work. In order to better evaluate our proposed method, we compare our proposed method with others, such as Histogram of Oriented Gradients - Support Vector Machine (HOG-SVM) method [4], multi-kernel Support Vector Machine (SVM) [18] and Sketch-a-Net [19, 20]. The result is shown in Figure 8.

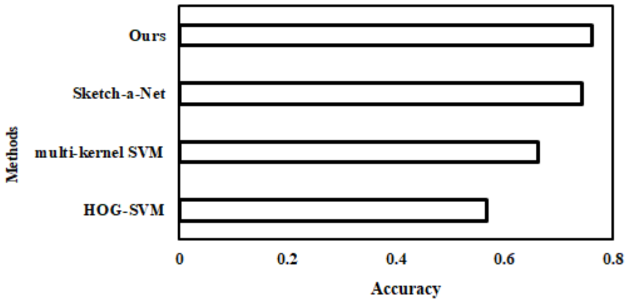

Figure 8. Comparative results on sketch recognition in different methods.

Next, we will perform the evaluation of context similarity to validate our proposed approach. Firstly, we randomly collect one thousand sketches in the test dataset and then extract their contextual features, and then predict the label of test samples by joint Bayesian model, and computed the accuracy as follows.

$$\text{Accuracy}\ (S_i) = \frac{1}{n}\sum_{k=1}^{n} f_i(actual(S_i), predicted(S_i)) \tag{8}$$

Where, actual (S_i) is the actual label name, and then *predicted* (S_i) is the predicted label name, the function $f_i(\cdot,\cdot)$ will compare whether the label names are equal, or 1 if they are equal, otherwise 0. The result will be shown in Figure 9.

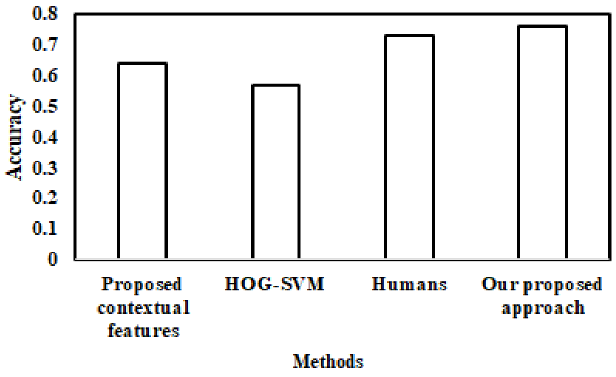

Figure 9. The evaluation result of contextual features method.

From the above Figure, it's not hard to find that proposed contextual feature is better method than others, apart from deep learning. The result is approach to the one of humans [19, 20]. Moreover, we conduct more experiments to find the relationship between the size of samples and the accuracy. The result will be shown in Figure 10.

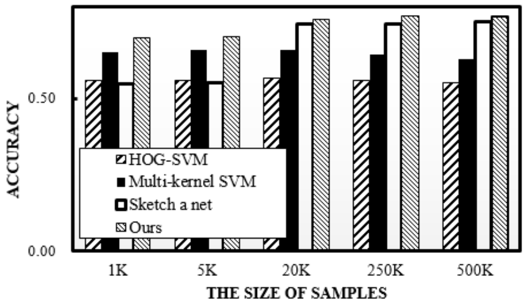

Figure 10. The compared result figure of different methods in five different samples size.

5.3. Discussion

In this subsection, we mainly discuss the result in Figure 10. From the above figure, it's not hard to find that our method can obtain good results both in small and large samples. In sketch a net [19, 20] approach, it shows very good performance when the samples are very large, whereas, if the samples aren't enough large, the result is not very good. The reason is that CNN need large scale learning samples, when the train samples isn't enough, the over fitting dilemma will haunt our Neural network. Therefore, it's very important that a large-scale sample are used to train, it is foundation of CNN acquiring good accuracy.

Moreover, we proposed method can obtain good performance when the size of samples is small. The reason is that our method consists of two kinds of features vectors, one is geometry features, and the other is contextual features. The joint Bayesian model always guarantee the better feature vectors can obtain bigger weights. Therefore, when the size of samples is very big, the feature vectors generated by CNN will obtain bigger weights, otherwise, the contextual feature will obtain bigger weights. In a word, our approach has a greater adaptability to the size of the samples space.

6. Conclusions

In this paper, we propose the multi-features learning framework for sketch recognition. Initially, through VGG16 CNN with pre-trained, geometry features are extracted, as well as we obtain the contextual relation via strokes traverse, which is named as contextual features. Hence, a complex four VGG16 CNNs are constructed on learning many different sketches. Besides, in order to widely enlarge the number of sketches in every category, we perform some tasks with exploring strokes order, local deformation and global deformation. Therefore, by these tasks, the sketches would scale up to huge. It can greatly improve the train samples spaces of our CNN and increase the diversity of sketches. Last but not least, we utilize the method of joint Bayesian to conduct different features metric. The final experimental result shows our method has arrived to high accuracy in test samples, as well as, it illustrates our method is totally feasible and superior.

However, our approach also exists some deficiencies, in train stage, it needs more execution time, above all, in extracting contextual features. In the future, we plan to adopt more approaches to decrease the computing time, such as, GPU acceleration. In fact, in test stage, we do not need to extract the contextual features, because we only depend on VGG16 network to obtain final predicting result, i.e., the sketch recognition label. In this way, the more execution time only affected the train procedure, it put less effects on sketch recognition. Additionally, there still exits some error prediction results, the reason mainly is that the diversity and scale of dataset is still not enough, therefore, in the future, we plan to obtain a greater number of sketch dataset, in order to improve the performance of ours. Furthermore, we consider to utilize more complex network to learn better representation, such as Google-Net.

References

[1] Bay H., Tuytelaars T., and Gool L., "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, 2006.

[2] Chen D., Cao X., Wang L., Wen F., and Sun J., "Bayesian Face Revisited: A Joint Formulation," *in Proceedings of European Conference on Computer Vision*, Florence, pp. 566-579, 2012.

[3] Donahue J., Jia Y., Vinyals O., Hoffman J., Zhang N., Tzeng E., and Darrell T., "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," *in Proceedings of International Conference on Machine Learning*, pp. 647-655, 2014.

[4] Eitz M., Hays J., and Alexa M., "How Do Humans Sketch Objects?," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1-10, 2012.

[5] Girshick R., Donahue J., Darrell T., Malik J., and Berkeley U., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *in Proceedings of Computer Vision and Pattern Recognition*, Columbus, pp. 580-587, 2014.

[6] Jabal M., Rahim M., Othman N., and Jupri Z., "A Comparative Study on Extraction and Recognition Method of CAD Data from CAD Drawings," *in Proceedings of International Conference on Information Management and Engineering. IEEE Computer Society*, Kuala Lumpur, pp. 709-713, 2009.

[7] Li B., Lu Y., Godil A., and Schreck T., "SHREC'13 Track: Large Scale Sketch-Based 3D Shape Retrieval," *in Proceedings of Eurographics Workshop on 3D Object Retrieval. Eurographics Association*, Girona, pp. 89-96, 2013.

[8] Liang S., Zhao L., Wei Y., and Jia J., "Sketch-Based Retrieval Using Content-Aware Hashing," *in Proceedings of Pacific Rim Conference on Multimedia*, Kuching, pp. 133-142, 2014.

[9] Lowe D., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[10] Lu T., Tai C., Su F., and Cai S., "A New Recognition Model for Electronic Architectural Drawings," *Computer-Aided Design*, vol. 37, no. 10, pp. 1053-1069, 2005.

[11] Ma C., Yang X., Zhang C., Ruan X., and Yang M., "Sketch Retrieval Via Dense Stroke Features," *Image and Vision Computing*,vol. 46, no. 2, pp. 64-73, 2016.

[12] Mikolajczyk K. and Schmid C., "Scale and Affine Invariant Interest Point Detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63-86, 2004.

[13] Razavian A., Azizpour H., Sullivan J., and Carlsson S., "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," *in Proceedings of Computer Vision and Pattern Recognition*, Columbus, pp. 806-813, 2014.

[14] Reddy N., Rao M., and Satyanarayana C., "Novel Face Recognition System by the Combination of Multiple Feature Descriptors" *The International Arab Journal of Information Technology*, vol. 16, no. 4, pp. 669-676, 2019.

[15] Schneider R. and Tuytelaars T., "Sketch Classification and Classification-Driven Analysis Using Fisher Vectors," *ACM Transactions on Graphics*, vol. 33, no. 6, pp. 174-184, 2014.

[16] Simonyan K. and Zisserman A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," *in Proceedings of International Conference on Learning Representations*, San Diego, pp. 1-14, 2015.

[17] Wang F., Kang L., and Li Y., "Sketch-based 3D shape Retrieval Using Convolutional Neural Networks," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, pp. 1875-1883, 2015.

[18] Yi L., Timothy M., Song Y., and Gong S., "Free-Hand Sketch Recognition By Multi-Kernel Feature Learning," *Computer Vision and Image Understanding*, vol. 137, pp. 1-11, 2015.

[19] Yu Q., Yang Y., Liu F., and Song Y., "Sketch-a-Net: A Deep Neural Network that Beats Humans," *International Journal of Computer Vision*, vol. 122, no. 3, pp. 411-425, 2016.

[20] Yu Q., Yang Y., Song Y., and Xiang T.,"Sketch-a-Net that Beats Humans," *in Proceedings of British Machine Vision Conference*, Swansea, pp. 101-112, 2015.

[21] Zitnick C. and Parikh D., "Bringing Semantics into Focus Using Visual Abstraction," *IEEE Conference on Computer Vision and Pattern Recognition*, Portland, pp. 3009-3016, 2013.

Wen Zhou received the Ph.D. degree from School of Software Engineering, Tongji University in 2018. Since2018, he has been in the School of Computer and Information, Anhui Normal University, China, where he is currently a lecturer, IEEE Member, Member of Chinese Computer Federation (CCF). His research interests include Virtual Reality, Sketch-based Retrieval and Machine Learning etc.

Jinyuan Jia received the Ph.D. degree from TheHong Kong University of Science and Technologyin 2004. Since 2007, he has been with Tongji University, Shanghai, China, where he is currently a Professor.His research interests include computer graphics, Web3D, mobile VR,etc. He is an ACM Member, and a Senior Member of the Chinese Computer Federation.