IIOAB JOURNAL

ARTICLE

# HAND DRAWN SKETCH CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

**Habibollah Agh Atabay***

*Department of Computer, Gonbad Kavous University, Gonbad Kavous, IRAN*

## ABSTRACT

*Sketch-based object recognition and classification has become an important research topic in vision applications. In the recent years, Convolutional Neural Networks (CNNs), have emerged as a powerful framework for feature representation and recognition for variety of applications in image analysis. But there exists few works that utilized CNNs in sketch classification domain. In this paper, a few deep CNNs are trained to improve the accuracy of sketch image classification in comparison with current state of the art. The size of inputs in the architectures of CNNs that currently have used for sketch recognition, has considered to be greater than 200×200 pixels which has limited the accuracy of classification. In this paper, tiny images are used as inputs, thus the architecture of CNNs are simplified and can be trained in a reasonable time, in CPU mode. These simplified CNN architectures are used to recognize sketch categories of TU-Berlin sketch dataset. The results show that the proposed method can outperforms state of the art performance of sketch recognition in addition to increasing the speed of training.*

**KEY WORDS**
*Sketch, Classification, Recognition, Convolutional Neural Network*

Published: 16 October 2016

***Corresponding Author**
Email:
atabay@gonbad.ac.ir
Tel.: +98-9353038148

## INTRODUCTION

Sketches are very intuitive to humans and have long been used as an effective communicative tool. Sketch can be seen as abstract representation of variety of ideas. Moreover, sketch images can convey information that is hard to describe using text, without requiring a tremendous amount of effort. With the proliferation of touch devices, sketch has been attracted more and more researches attentions in recent years. There exist a wide range of interesting applications including sketch recognition [1-3], sketch-based image retrieval [4, 5] and sketch-based shape [6, 7] and 3D model retrieval [7, 8].

Recognizing free-hand sketches is an extremely challenging task. This is due to a number of reasons [3]: (1) sketches are highly iconic and abstract; (2) due to the free-hand nature, the same object can be drawn with hugely varied levels of detail/abstraction; (3) sketches lack visual cues, such as color and texture information.

Traditional methods on sketch recognition generally follows the conventional image classification paradigm that is, extracting hand-crafted features from sketch images followed by feeding them to a classifier. Most hand-crafted features traditionally used for photos such as HOG [9], SIFT [10] and shape context [11], have been employed, which are often coupled with bag of visual words to yield a final feature representations that can then be classified. But these features are sensitive to the view perspectives and some appearance cues of drawn sketches. Furthermore, the ability of learning algorithms to train the classification models are also influenced by the handcrafted features and the capacity of classifiers like SVM to memorize feature information.

Learning robust and invariant representation has been a long-standing goal in computer vision. In comparison to hand-crafted visual features, features learned by deep neural networks (DNNs) have recently been shown more capable of capturing abstract concepts invariant to various phenomenon in visual world [12, 13]. In the recent years, Convolutional Neural Networks (CNNs), a specific type of DNNs, have emerged as a powerful framework for feature representation and recognition for a variety of image domains. A deep CNN is able to learn basic filters automatically and combine them hierarchically to enable the description of latent concepts for pattern recognition, and have successfully been applied to large image datasets such as ImageNet [12], MNIST [14], CIFAR-10/100 [15] for image classification. However, the domain of sketch images has been rarely explored. With emerge of large-scale sketch datasets such as TU-Berlin (Eitz, et al, 2012), using CNNs in sketch classification has attracted more attention [3, 16-18].

In this paper, a few deep CNNs are trained to improve the accuracy of sketch image classification, in comparison with current state of the art. The contribution of the paper is as follows: (1) Working with tiny input images: in the conventional methods, rather large scale images were used to extract feature vectors to feed them to a classifier as a descriptor of the original images to be categorized. But mining these raw images, in large numbers, using DNNs is a difficult and practically unefficient task. Therefore, in the case of DNNs, using smaller scales of images has been preferred, because the amount of information in the down-scaled image is often not changed in comparison with the original image. Currently a commonly used image scale as the input size of CNNs is 32×32 pixels. But in the current related works in sketch classification (that summarized in the next section), this input scale have not been considered. Thus in this paper, the input size is fixed as 32×32 to experiment more possibilities. (2) Fewer number of layers and neurons: the number of layers and neurons in DNNs, due to cover different levels of information, is very important. But excessive levels, slow down the learning process. On the other hand, the amount of information in sketch images are much lower that colorful, texturized natural images, and it seems that

CNNs can process such images without need of very deep architectures like AlexNet [12]. Therefore this paper tries to use simple but powerful networks that can be trained in CPU mode (i.e. not to use GPU hardware to accelerate training process) in an affordable amount of time. (3) Accuracy of classification: it is found that these simple architectures outperforms state-of-the-art performance for sketch recognition, without the need for large scale input images, complicated augmentations, specific activation functions and response normalization. The recognition frameworks are evaluated on TU-Berlin [1], the publicly available benchmark dataset, containing thousands of freehand sketches.

The rest of the paper is organized as follows: first the related literature is reviewed in the second section. Then the trained CNN architectures, used in this work, are presented. The experiments and results are showed in fourth section and the paper is concluded in the final section.

## RELATED WORKS

In the first attempt to utilize CNNs to recognize hand-drawn sketches and object categories, [16], two popular CNNs – AlexNet CNN [12] and a modified version of LeNet CNN [14] were used for experiments, and results showed minor improvements over the conventional state-of-the-art.

The major work on utilizing Deep CNNs for free-hand sketch recognition was Sketch-a-Net which firstly was introduced in [17] and further revised in [3]. Sketch-a-Net aimed to exploit the unique characteristics of sketches, including multiple levels of abstraction and being sequential in nature. In the work, ensemble fusion and pre-training strategies were applied to boost the recognition performance. Compared to the earlier version of Sketch-a-Net, a number of modifications were applied in the latest network. In the second version, the authors used stroke timing and geometry information to define a data augmentation strategy that synthesizes sketches at varying abstraction levels, and deformed them to achieve a richer training set and to alleviate the problem of over-fitting to scarce sketch data. They achieved 77.95% classification accuracy on TU-Berlin sketch dataset [1].

The very recent attempt in this area is [18], where another deep convolutional neural network, similarly named SketchNet, was proposed for sketch classification. But the main purpose of the paper is to automatically learn the shared structures that exist between sketch images and real images. In order to do this, a triplet was composed of sketch, positive and negative real image that was developed as the input of the neural network. The authors used SoftMax as lost function and ranked the results to make the positive pairs to obtain a higher score comparing over negative ones to achieve robust representation. To construct the auxiliary repository, the real images were collected from the web which covered all the sketch categories in the TU-Berlin sketch dataset. To extract the real reference images for each training sketch, first a preliminary model was trained based on AlexNet [12] following the fine-tuning process. Afterwards, the top K predicted category labels of each training sketch were extracted based on the pre-trained AlexNet model. For each training sketch, the most visual similar real images were found from the image sets of top predicted categories to construct the training pairs. Thus, the sketch with the real images which is in the same class was used to generate the positive image pair while the sketch with the real images which is in distinct classes was defined as negative image pair. SketchNet contained three subnets: R-Net was used to extract features from the real images. S-Net was applied on the sketch images. And C-Net was proposed to discover the common structures between real images and sketches. Finally, the predictions were merged together to achieve the final results. The best classification accuracy, achieved in the paper on TU-Berlin sketch benchmark, was 80.42%.

Another related paper that used CNNs in sketch recognition is [19], in which a CNN based classifier for 3D shapes using 2D image renderings of those shapes was proposed. In particular, a CNN was trained on a fixed set of rendered views of a 3D shape and was provided with a single view at test time. The authors used their method as a data augmentation method and applied in the context of sketch recognition on SketchClean dataset [2]. SketchClean is a simpler version of TU-Berlin sketch dataset [1] on which humans can achieve 93% recognition accuracy compared to 73% on original dataset. They achieved 87.2% sketch classification accuracy on this simplified dataset.

**Table 1:** The architecture of proposed CNNs

| Layers | MNIST-CNN | CIFAR-CNN | TS-CNN |
|---|---|---|---|
| Input | 32×32×1×20 | 32×32×1×64 | 32×32×1×96 |
| L1 | Conv - 5×5 - 1<br>Pool - 2×2 - 2 | Conv - 3×3 - 1<br>Pool - 3×3 - 3 | Conv - 5×5 - 1<br>Pool - 3×3 - 2 |
| Input | 14×14×20×40 | 10×10×64×128 | 14×14×96×192 |
| L2 | Conv - 5×5 - 1<br>Pool - 2×2 - 2 | Conv - 3×3 - 1<br>Pool - 2×2 - 2 | Conv - 5×5 - 1<br>Pool - 3×3 - 2 |
| Input | 5×5×40×150 | 4×4×128×256 | 4×4×192×192 |
| L3 | Conv - 3×3 - 1<br>Pool - 2×2 - 2 | Conv - 3×3 - 1<br>Pool - 2×2 - 2 | Conv - 3×3 - 1 |
| Input | 1×1×150×250 | 1×1×256×128 | 2×2×192×192 |
| L4 | Conv - 1×1 - 1 | Conv - 1×1 - 1 | Conv - 1×1 - 1 |
| Input | 1×1×150×250 | 1×1×128×250 | 2×2×192×250 |
| L4 | ---- | Conv - 1×1 - 1 | Conv - 1×1 - 1 |
| Loss | SoftMax | | |

In all above mentioned works, input images have considered greater than 200×200 pixels and extensive endeavors have been applied to achieve the results. But at the point of view of this paper, such a great size (for deep CNNs to work on) is not necessary, specifically on sketch datasets. It is possible to work with smaller sizes and achieve even better results with modification of current state-of-the-art CNN architectures.

## TRAINED CNN ARCHITECTURES

Designing an architecture for CNN is an important step to achieve impressive results on vision problems. Architectures may vary with type of images and especially when input image sizes are different. In this paper, the size of input images is considered to be 32×32 pixels thus the CNN architectures, proposed in previous works, cannot be applied without modifications on these images (for example applying 15×15 filters in the first layer of the CNN, in the case of Sketch-a-Net, on input size of 32×32 pixels is not rational). Therefor the architectures that were used before (Sato et al, 2015) to classify images of size 32×32 pixels include MNIST-CNN and CIFAR-CNN and an architecture that inspired from [20] (for ease of reference, is named TS-CNN, stands for Tiny Sketch CNN) were trained to sketch classification task. These architectures are described in [Table 1].

In all networks, the ReLU activation function [12] was used for each convolutional layer and MaxPooling for each pooling layer. In the architectures, randomness to the networks, such as Dropout [21] or DropConnect [22] was not imposed during training. In [Table 1], records named “Input” specifies the map size and the number of inputs and output maps of the below and the above layer; the first and second elements are the width and height of the input (output) map of the layer below (above), and the third and fourth elements are the number of input (output) maps of the below (above) layer and the number of output (input) maps of the below (above) layer, respectively. Also, the definition of layers in [Table 1] consists of three parts; type of layer, filter size and stride length, respectively. The fully connected layers are defined as convolutional layers with the filter size of 1×1 as it is conventional in MatConvNet [23]. The final layer has 250 output units corresponding to 250 categories of the dataset, upon which a SoftMax loss is placed.

## EXPERIMENTAL RESULTS

In this section, the results of the proposed methods on the task of sketch classification are presented. The

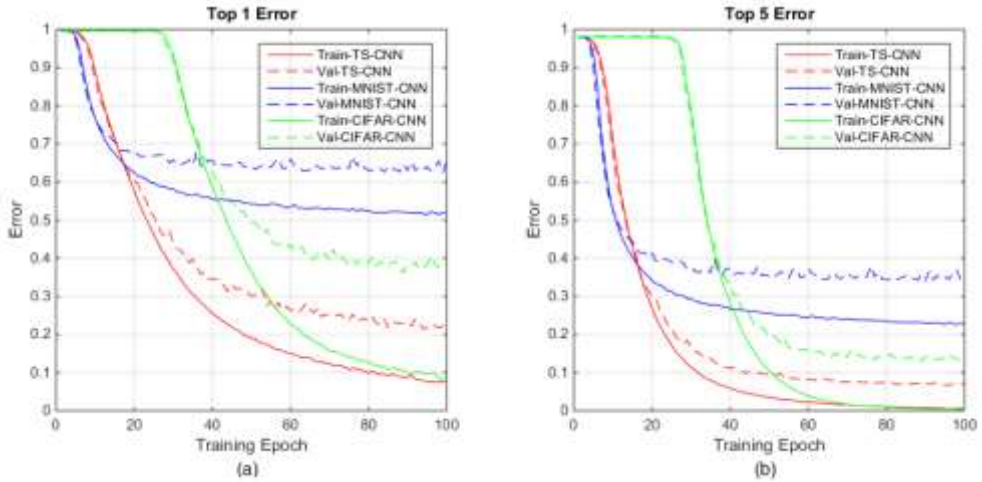


**Fig. 1:** Comparison of training and validation curves of the proposed networks.

**Table 2:** The classification results of the CNNs for sketch recognition on TU-Berlin sketch benchmark

| Method | Top 5 Accuracy | Top 1 Accuracy |
|---|---|---|
| TSCNN | **98.97** | **89.17** |
| Sketch-a-Net | --- | 77.95% |
| CIFAR-CNN | 86.42 | 62.70 |
| SketchNet | 80.42% | --- |
| MNIST-CNN | 63.93 | 35.01 |

networks are implemented using MatConvNet [23] on the TU-Berlin sketch dataset [1]. All images are down-scaled to 32×32 pixels and each pixel's value is converted to range [0, 1]. The conventional data augmentation methods are applied by replicating the sketches with a number of transformations. Specifically, for each input sketch, horizontal reflection and horizontal and vertical shifts (3 pixels) are done. These data augmentation strategies increase the training data size by a factor of 10. These 200000 images are randomly split into 70% train, 10% validation and 20% test sets.

In all experiments, the networks are trained by stochastic gradient descent with 0.9 momentum. The learning rate is set to be 0.001. The weight decay parameter is 0.0005. Initial weights are selected randomly from standard normal distribution, multiplied by 0.01. The mini-batch size is set to 100. All networks were trained up to 100 epochs. Overall training takes less than one day based on a PC with 3.30 GHz CPU and 8 GB RAM, using CPU mode. The training and validation curves of all three networks are depicted in [Fig. 1].

In [Fig. 1] the training process of CNNs can be compared. This figure demonstrates that TSCNN converges faster than the other CNNs. This quick convergence also lead to the better results after 100 epochs.

The sketch recognition results of the proposed CNNs, compared to the related works, are reported in [Table 2]. The results of SketchNet and Sketch-a-Net are stated as they were reported in corresponding works, (Zhang, et al, 2016) and (Yu, et al, 2016), respectively. In the case of SketchNet, only top 1 accuracy was mentioned in the original paper and for Sketch-a-Net the authors does not specified the type of accuracy thus it is considered as the top 1 accuracy. This table shows that TS-CNN significantly outperforms other methods, especially in comparison with SketchNet and Sketch-a-Net which specifically designed for sketch classification.

The results show that working with small size of input images, not only eliminates the need for additional layers but also increases the accuracy of learning. This phenomenon is reasonable because the sketches are so powerful in conveying information, even in small size and for CNNs working with small sized input is more convenient.

It must be noted that the architectures examined in this work were inspired from other related works and most of the hyper-parameters were set to the best common values. Thus, the results can be even better by applying recent modifications and discoveries on deep neural networks.

## CONCLUSIONS

In this paper, a few different CNN architectures inspired by latest achievements on training DNNs were applied to sketch classification task with tiny input sketches. It is observed that small input size, dramatically improves the current classification accuracies on TU-Berlin sketch benchmark and simplifies the CNN architectures which also lead to increase in speed of training. Most of the hyper parameters in this work are set to common best fixed values, thus it is not hard to believe that even better results can be achieved with further investigations on architectures and hyper parameters and applying recent modifications and discoveries on deep neural networks.


## CONFLICT OF INTEREST

There is no conflict of interest.

## ACKNOWLEDGEMENTS

None

## FINANCIAL DISCLOSURE

None


## REFERENCES


[1] Eitz M, Hays J, Alex, M. [2012] How do humans sketch objects? ACM Trans. Graph, 31, 44:1-:10.

[2] Schneider, R.G, Tuytelaars, T. [2014] Sketch classification and classification-driven analysis using fisher vectors. ACM Transactions on Graphics (TOG). 33:174.

[3] Yu Q, Yang Y, Liu F, Song Y.-Z, Xiang T, Hospedales TM. [2016] Sketch-a-Net: A Deep Neural Network that Beats Humans. International Journal of Computer Vision. 1-15.

[4] Eitz, M, Hildebrand, K, Boubekeur, T, Alexa, M. [2011] Sketch-based image retrieval: Benchmark and bag-of-features descriptors. IEEE Transactions on Visualization and Computer Graphics. 17: 1624-1636.

[5] Hu R, Collomosse, J. [2013] A performance evaluation of gradient field hog descriptor for sketch based image retrieval. Computer Vision and Image Understanding. 117: 790-806.

[6] Wang F, Kang L, Li Y. Sketch-based 3d shape retrieval using convolutional neural networks. In: Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition, [2015], pp. 1875-1883.
[7] Zou C, Huang Z, Lau RW, Liu J, Fu H. [2015] Sketch-based Shape Retrieval using Pyramid-of-Parts. arXiv preprint arXiv:1502.04232.
[8] Liu Y.-J, Luo X, Joneja, A, Ma, C.-X, Fu, X.-L, Song, D. [2013] User-adaptive sketch-based 3-D CAD model retrieval. IEEE Transactions on Automation Science and Engineering. 10:783-795.
[9] Dalal N, Triggs B. [2005] Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), IEEE, 1:886-93.
[10] Lowe, D.G. [2004] Distinctive image features from scale-invariant keypoints. International journal of computer vision. 60, 91-110.
[11] Belongie, S, Malik, J, Puzicha, J. [2002] Shape matching and object recognition using shape contexts. IEEE transactions on pattern analysis and machine intelligence. 24: 509-522.
[12] Krizhevsky A, Sutskever I, Hinton GE. [2012] Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097-1105.
[13] Simonyan K, Zisserman A. [2014] Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[14] LeCun Y, Bottou L, Bengio Y, Haffner P. [1998] Gradient-based learning applied to document recognition. Proceedings of the IEEE. 86:2278-324.
[15] Krizhevsky, A, Hinton, G. [2009] Learning multiple layers of features from tiny images.
[16] Sarvadevabhatla, R.K, Babu, R.V. [2015] Freehand Sketch Recognition Using Deep Features. arXiv preprint arXiv:1502.00254.
[17] Yu Q, Yang Y, Song Y.-Z, Xiang T, Hospedales T. [2015] Sketch-a-net that beats humans. arXiv preprint arXiv:1501.07873.
[18] Zhang, H, Liu S, Zhang C, Ren W, Wang, R, Cao X. [2016], SketchNet: Sketch Classification With Web Images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1105-13.
[19] Su, H, Maji, S, Kalogerakis, E, Learned-Miller, E. Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE International Conference on Computer Vision, [2015], pp. 945-53.
[20] Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M. [2014] Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806.
[21] Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. [2012] Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.
[22] Wan L, Zeiler M, Zhang S, Cun YL, Fergus R. [2013] Regularization of neural networks using dropconnect. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 1058-1066.
[23] Vedaldi A, Lenc K. [2015] Matconvnet: Convolutional neural networks for matlab. In: Proceedings of the 23rd ACM international conference on Multimedia, ACM, pp. 689-692.