

Sketch Recognition Classification

Wayne Lu
Stanford University
waynelu@stanford.edu

Elizabeth Tran
Stanford University
eliztran@stanford.edu

Abstract

Automatic recognition of sketches differs from other areas of image classification because sketches of the same object can vary based on artistic style and drawing ability. In addition, sketches are less detailed and thus harder to distinguish than photographs. Using a publicly available dataset of 20,000 sketches across 250 classes from Eitz et al. [cite], we are applying ConvNets in order to improve performance over traditional multi-class support vector classifiers (SVMs) using by experimenting with different architecture methods, such as quadrant pooling and fine tuning, and hyperparameters to asses the overall performance of our proposed model.

1. Introduction

Sketching is one of the primary methods people use to communicate visual information. Since the era of primitive cave paintings, humans have used simple illustrations to represent real-world objects and concepts. Sketches are often abstract and stylized, varying based on artistic ability and style. In addition, sketches emphasize defining characteristics of real-world objects and ignore features which are either less important or more difficult to draw. For example, texture is almost never rendered unless it is important for recognition, such as the spikes on a hedgehog. In this way, sketches can be interpreted as a distillation of human visual recognition schemas.

The sketch recognition problem differs from traditional photographic image classification. First, sketches are less visually complex than photographs. Whereas color photographs have three color channels per pixel, sketches are encoded as either black-and-white or grayscale. Photographs contain visual information throughout the image whereas sketches consist primarily of blank space. Second, sketches and photographs have different sources of intra-class variation. Whereas photographic image classification faces obstacles such as camera angle, occlusion, and illumination, photographs are still grounded in reality. On the other hand, sketches differ based on artistic style, which is

unique to every artist. While people can agree on what an object looks like, how they ultimately render the object can vary significantly.

In this paper, we explore various convolutional neural network architectures for sketch recognition, including classic models like AlexNet [2] and more recent deep architectures such as ResNet.

2. Related Work

Since SketchPad [6], sketch recognition has introduced sketching as a means of human computer interaction. Computer vision has since tried different approaches to achieve better results in multiple application areas. Eitz et al. [1] were able to demonstrate classification rates can be achieved for computational sketch recognition by using local feature vectors, bag of features sketch representation and SVMs to classify sketches. Schneider et al. [3] then modified the benchmark proposed by Eitz et al [1] by making it more focused on how the image should look, rather than the original drawing intention, and they also used SIFT, GMM based on Fisher vector encoding, and SVMs to achieve sketch recognition.

3. Dataset

Eitz et al. [1] defined a taxonomy of 250 object categories gathered from 20,000 human sketches (TU-Berlin sketch benchmark). The dataset has a total of 250 classes, and it is split into three categories: exhaustive, recognizable, and specific. Images are available as 1111x1111 px PNG files, which we have resized to 128x128 px grayscale images.

4. Our Approach

Our approach is based on the use of partial sketches of the TU-Berlin sketch benchmark. Instead of the original sizes of the images, of TU-Berlin sketch benchmark 256 x 256, we resized our images to 128 x 128.



Figure 1. Example sketch from TU Berlin dataset.

Method	mAP
SIFT-variant + BoF + SVM [1]	56
IDM + SVM [8]	71.30
ConvoNet [4]	75.42
ConvoNet [6]	77.69

Table 1. Performance compairison of our models versus other methods.

5. Experiments

The first model that we train is a single sketch CNN. We would pass an individual sketch through the CNN. The CNN has 3 convolutional layers consisting of 128×128 , 64×64 , and 32×32 filters respectively. Each convolutional layer is followed by a batch normalization, ReLU (Rectified Linear Unit) activation function, and a 2×2 max pooling. After the convolutional layer is a fully-connected layer, we use cross entropy as a cost function. With this model, we were able to achieve a test accuracy of 0.004.

In order to improve our network, we are going to try different parameters such as CNN with different levels of regularization, CNN with dropout, CNN with data augmentation, and CNN with dropout and data augmentation.

6. References

- [1] Eitz, Mathias, HAYS, James, Alexa, "Marc. How do humans sketch objects?", ACM Trans. Graph., vol. 31, no. 4, pp. 44, 2012.
- [2] Krizhevsky, Alex, Sutskever, Ilya, Hinton, E. Geoffrey, "Imagenet classification with deep convolutional neural networks", Advances in neural information processing systems, pp. 1097-1105, 2012.
- [3] Schneider, G. Roslia, Tuytellaars, Tinne, "Sketch Classification and Classification-driven Analysis using Fisher Vectors", ACM Transactions on Graphics (TOG), vol. 33, no. 6, pp. 174, 2014.
- [4] Seddati, Omar, Dupont, Stephane, et Mahmoudi, Said. Deepsketch: deep convolutional neural networks for sketch recognition and similarity search. In : Content-

Based Multimedia Indexing (CBMI), 2015 13th International Workshop on. IEEE, 2015. p. 1-6.

[5] Seddati, O., Dupont, S., & Mahmoudi, S. (2016, June). DeepSketch 2: Deep convolutional neural networks for partial sketch recognition. In Content-Based Multimedia Indexing (CBMI), 2016 14th International Workshop on (pp. 1-6). IEEE.

[6] Sutherland, E. Ivan, "Sketch pad a man-machine graphical communication system", Proceedings of the SHARE design automation workshop. ACM, pp. 6.329-6.346, 1964.

[7] Szegedy, Christian, LIU, Wei, JIA, Yangqing et al., "Going deeper with convolutions. arXiv preprint arXiv: 1409.4842", 2014.

[8] Yesilbek, K. T., Sen, C., Cakmak, S., et al. SVM-based sketch recognition: which hyperparameter interval to try?. In : Proceedings of the workshop on Sketch-Based Interfaces and Modeling. Eurographics Association, 2015. p. 117-121