CIS 519

Project

Group 14

Sketch Classification using Deep Convolutional Neural Network for Interactive Drawing Interface

Pedro Rizo

Byeongchan Lee

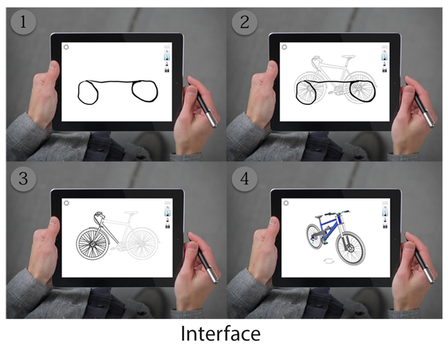
Nikhil Jamdade

ABSTRACT

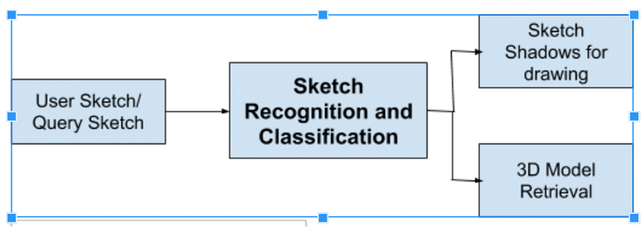
An interactive interface is designed to improve drawing skills and retrieve 3D shapes based on sketch classification and sketch matching. The system involves input, output and computational steps to recognise and classify free-hand sketch using deep learning neural network approach. This paper explores the classifier built using a convolutional neural network (CNN) based on the Sketch-a-Net's CNN architecture. We not only replicated the architecture but further experimented with the model by applying it to unseen data ("real world data") to understand zero-shot learning. Moreover, we have experimented with the architecture with multi-channel generalisation that encodes sequential ordering in the sketching process, and a multi-scale network ensemble with joint Bayesian fusion that accounts for the different levels of abstraction exhibited in free-hand sketches. The primary objective of this project is to develop and experiment with set of algorithms and classifiers in order to accurately classify drawing / human drawn sketches into correct categories.

INTRODUCTION

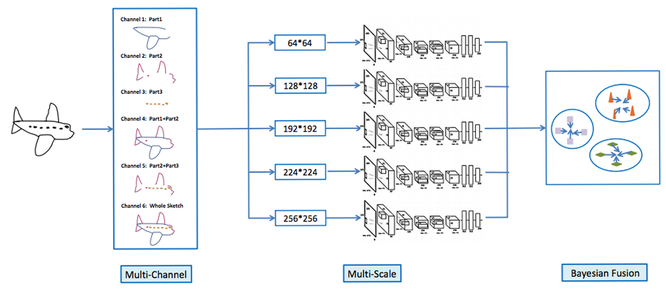
Sketches are very intuitive to humans and have long been used as an effective communicative tool. With the proliferation of touchscreens, sketching has become a much easier undertaking for many – we can sketch on phones, tablets and even watches. The recognition of free hand drawn sketches/drawing is extremely challenging task unlike image recognition. This is particularly because of sketches are highly iconic and abstract with highly varied different level of abstractions/details. Sketches lack visual cues as they consist of black and white lines instead of colors. Initially sketch and image recognition generally followed conventional methods such as feature extraction using HOG or SIFT etc and feeding them to the classifier like SVM. However these methods does not count the unique abstract and sparse nature of sketch, moreover they ignore sequential stroke ordering in sketch. The project follows deep neural network (DNN) approach from Sketch-a-Net. Recently DNN especially deep convolutional neural network have achieved tremendous success in replacing representation hand crafting with representation learning for variety of vision and Machine Learning problems. An important advantage of DNNs, particularly CNNs compared with conventional classifiers such as SVM's les with closely coupled nature of presentation learning and classification i.e



Interface



Interactive Interface Block diagram



Framework

Index	Layer	Type	Filter Size	Filter Num	Stride	Pad	Output Size
0		Input	-	-	-	-	225 × 225
1	L1	Conv	15 × 15	64	3	0	71 × 71
2		ReLU	-	-	-	-	71 × 71
3		Maxpool	3 × 3	-	2	0	35 × 35
4	L2	Conv	5 × 5	128	1	0	31 × 31
5		ReLU	-	-	-	-	31 × 31
6		Maxpool	3 × 3	-	2	0	15 × 15
7	L3	Conv	3 × 3	256	1	1	15 × 15
8		ReLU	-	-	-	-	15 × 15
9	L4	Conv	3 × 3	256	1	1	15 × 15
10		ReLU	-	-	-	-	15 × 15
11	L5	Conv	3 × 3	256	1	1	15 × 15
12		ReLU	-	-	-	-	15 × 15
13		Maxpool	3 × 3	-	2	0	7 × 7
14	L6	Conv(=FC)	7 × 7	512	1	0	1 × 1
15		ReLU	-	-	-	-	1 × 1
16		Dropout (0.50)	-	-	-	-	1 × 1
17	L7	Conv(=FC)	1 × 1	512	1	0	1 × 1
18		ReLU	-	-	-	-	1 × 1
19		Dropout (0.50)	-	-	-	-	1 × 1
20	L8	Conv(=FC)	1 × 1	250	1	0	1 × 1

CNN Architecture

Methodology

1. **Data Augmentation**
2. **CNN for Sketch Recognition**

It uses five convolution layers each with rectifier (ReLU) units, while the first, second and fifth layers are followed by max pooling. The filter size of the sixth convolution layer is 7x7 which is the same as the output from the previous pooling layer which works like fully connected layer. The softmax loss is placed after the final layer. The unique aspects of the architecture for sketch modelling compare to other network are larger first layer filter, no local response normalisation, larger polling size, higher drop-out etc.

3. **Modelling Sketch Stroke Order with Multiple Channel**

A multi channel architecture is designed to model the sequential ordering of strokes in each sketch.The sequential ordering is strong cue in human sketch recognition.

4. **A Multi-scale Network Ensemble with Bayesian Fusion**

A multi-scale network ensemble to address the variability in abstraction and sparsity, followed by a joint Bayesian fusion scheme to exploit the complementarity of different scales. In this method each network in the ensemble is learned a model of varying coarseness by blurring training data to different degrees.

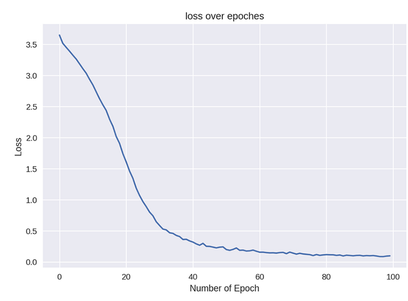
We have used pretrained model and pre ordered stroke data to experiment with the DNN architecture on dataset.

CNN Result

```
epoch:  90 loss:  0.1060623750090599 /-----/ Accuracy:  0.9673
epoch:  91 loss:  0.09658442437648773 /-----/ Accuracy:  0.9674
epoch:  92 loss:  0.101921446621418 /-----/ Accuracy:  0.9647
epoch:  93 loss:  0.09924624860286713 /-----/ Accuracy:  0.9697
epoch:  94 loss:  0.10237214714288712 /-----/ Accuracy:  0.9678
epoch:  95 loss:  0.09513237327337265 /-----/ Accuracy:  0.9686
epoch:  96 loss:  0.08573770523071289 /-----/ Accuracy:  0.9729
epoch:  97 loss:  0.08527219295501709 /-----/ Accuracy:  0.9728
epoch:  98 loss:  0.09366397559642792 /-----/ Accuracy:  0.969
epoch:  99 loss:  0.09763593971729279 /-----/ Accuracy:  0.9699
final training accuracy:  0.9699
```

Stroke Order + CNN+BJ Result

```
Starting the training...
Loading the dataset...
Dataset loaded!
dataset/model_with_order_info_224.mat
Pretrained Model Loaded!
Max Steps: 100
Epoch: 1
Batch Size: 135
Step 10001: loss = 0.23 (51.582 sec)
Step 10011: loss = 0.24 (51.718 sec)
Step 10021: loss = 0.21 (46.368 sec)
Step 10031: loss = 0.07 (48.082 sec)
Step 10041: loss = 0.23 (7377.978 sec)
Step 10051: loss = 0.24 (47.334 sec)
Step 10061: loss = 0.05 (45.179 sec)
Step 10071: loss = 0.33 (47.441 sec)
Step 10081: loss = 0.14 (47.151 sec)
Step 10091: loss = 0.11 (46.840 sec)
Running Evaluation on testset
  Num examples: 6500  Num correct: 4158  Precision @ 1: 0.6397 (3626.734 sec)
```



Dataset

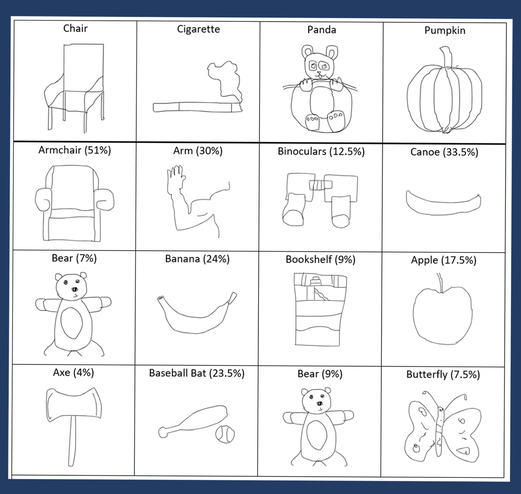
The data set contains a total of 20,000 sketches that were obtained by crowdsourcing and were made by non-experts/(non artists), which provides vital because we want to capture the "rough" or often "inaccurate" drawings that people uses to represent high level abstractions of daily things like a bird or a toothbrush. Furthermore, the 20,000 sketches are divided into a total of 250 categories that try to provide a representative sample of many of the daily words used in human interactions. Each of the categories contains a total of 80 sketches.

DISCUSSION AND CONCLUSION

- unseen data performance
- only CNN (without stroke order info): 0.42 (test), 0.943 (training)
- stroke order + CNN + JB : 0.64
- stroke order & multi-scale JB did matter in terms of performance

For testing unseen data with our implementation of only CNN-based classifier, same number of images from each 50 classes are used. If the model did not learn after training and guessed arbitrarily on test data, the result would have been near 0.02. Given this base line, we can conclude that our implementation of convolutional neural network for sketch recognition can effectively recognize and classify unseen data. However, the result is relatively low compared to the experiment data for the sketch-a-net implementation that incorporates stroke ordering and multiscale network ensemble with Bayesian fusion, which produces an unseen data accuracy of 0.64 after just two epochs. The result confirms that unique features found in sketches such as stroke orders can be the key to performance improvement for the sketch recognition system.

zero-shot learning - it demonstrates possibility zero-shot learning based on semantic information. The predictions for the labels, for which no instances is provided to the model based on our CNN implementation for training, exhibited an interesting result. Top 3 predictions for those labels with no training data - "chair", "cigarette", "panda", and "pumpkin" - shared some important characteristics with the actual label. They tend to have similar orientation, shape, or pattern. We saw this result as a promising sign for the possibility of zero-shot learning since this may indicate that the model / network is actually learning semantic information about the sketch, which, in turn, can be used to inform a sketch recognition system so that it can predict labels without any training data but with provided semantic information.

Experiment Results

	Top1	Top2	Top3
Chair	Armchair (102/ 200 = 51%)	Bear (14 / 200 = 7%)	Axe (8 / 200 = 4%)
Cigarette	Arm (60 / 200 = 30%)	Banana (48 / 200 = 24%)	Baseball Bat (47 / 200 = 23.5%)
Panda	Binoculars (25 / 200 = 12.5%)	Bookshelf (18 / 200 = 9%)	Bear (18 / 200 = 9%)
Pumpkin	Canoe (67 / 200 = 33.5%)	Apple (35 / 200 = 17.5%)	Butterfly (15 / 200 = 7.5%)