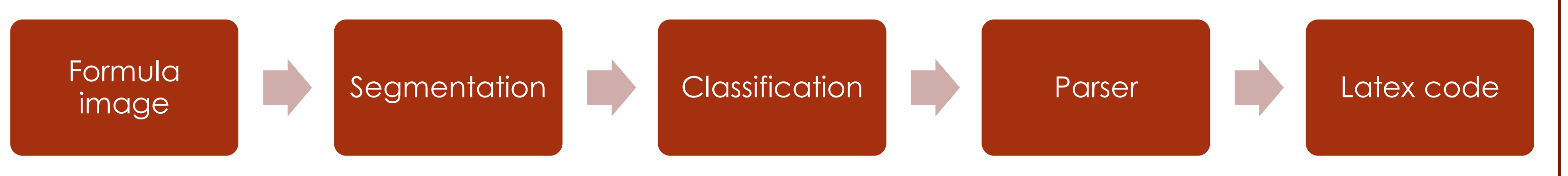
Offline Mathematical Expression Recognizer

Shawn Peng: yisupeng@iu.edu
Yang Zhang: zhang505@Indiana.edu

2016 Spring B657 Computer Vision
Prof. David Crandall

System Architecture



Formula image → Segmentation → Classification → Parser → Latex code

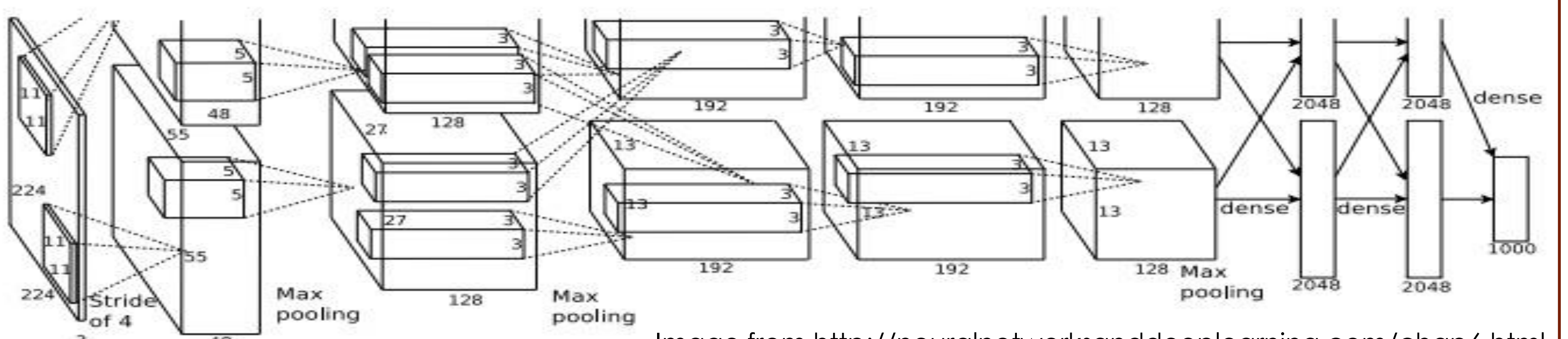
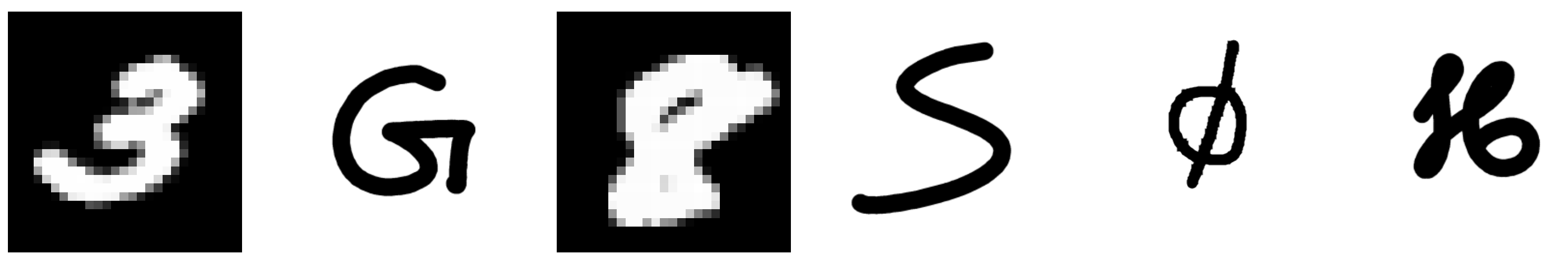


Image from http://neuralnetworksanddeeplearning.com/chap6.html

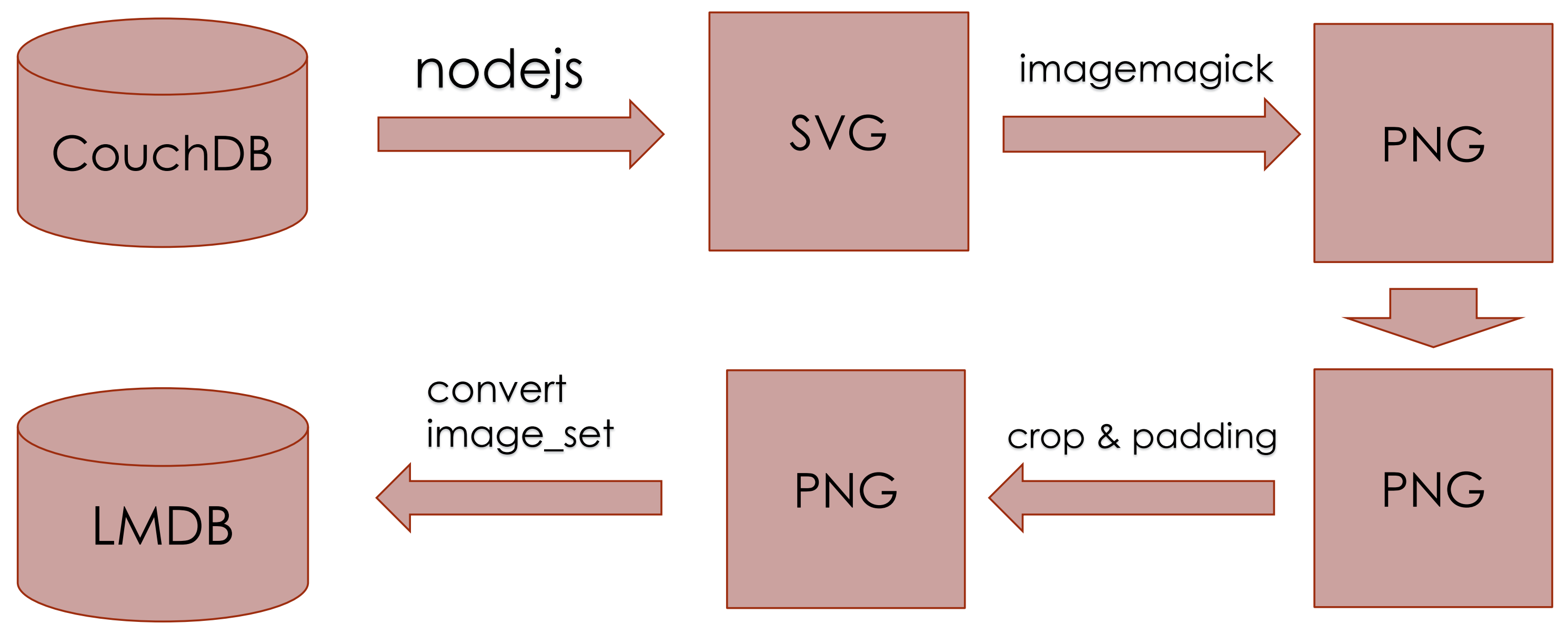
Data Set

7	Digits	MNIST
K	English	Chars74K, Write-Math
Ψ	Greek	Detexify
∫	Math Symbols	Write-Math



Since we use different data sources, and they use different image formats, we need to preprocess those data. Some of them is preprocessed, like mnist and Chars74K dataset, but Chars74k, Detexify and write-math dataset use whte background.

Preprocessing



CouchDB → nodejs → SVG → imagemagick → PNG → PNG → crop & padding → PNG → convert image_set → LMDB

- We use nodejs to convert the couchdb dat to svg and make the stroke border.
- We use imagemagick to negate, resize the image and convert it to png
- We use python to crop the image and add our image
- We balance the training and testing data set by resampling
- The last step is to use caffe tool conver_image_set to convert the image to lmdb

Conclusion & Discussion

First, we found it is very useful to have the character to be properly bold before we send it to the convolutional neural network. For svg image this can be done very easily, for images one thing we can use is the dilution.

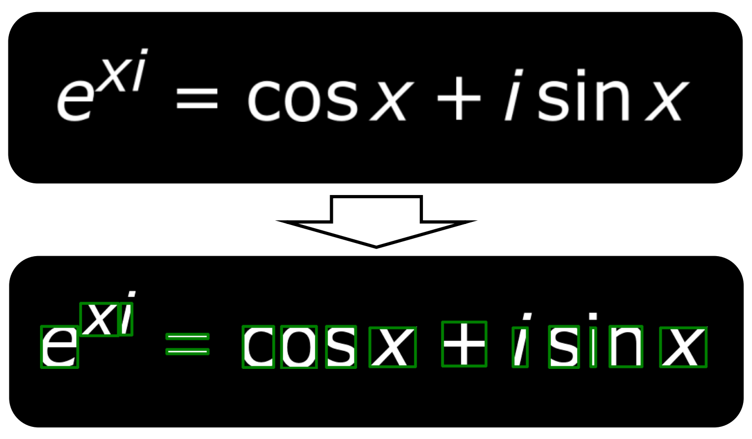
Second, in the training of our convolutional neural network (we use LeNet), we find it is very useful to perform the trim operation on images.

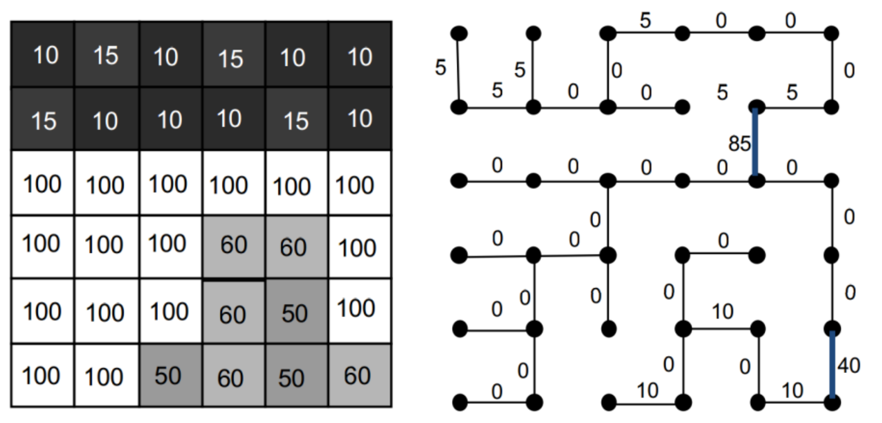
On the other hand, the problem is that we do not have enough data to train neural network. We have enough digit number and geek letter data, but for English letter we only get 50 samples. We believes that is the problem that we only get 81.5% accuracy.

We have not begin to do grammar parse because of the poor performance of classifications.

We thank Prof. David Crandall for really helpful comments and discussions. We also thank Dr. Daniel Kirsch and Martin Thoma to provide us the permission to the handwrite data.

Segmentation



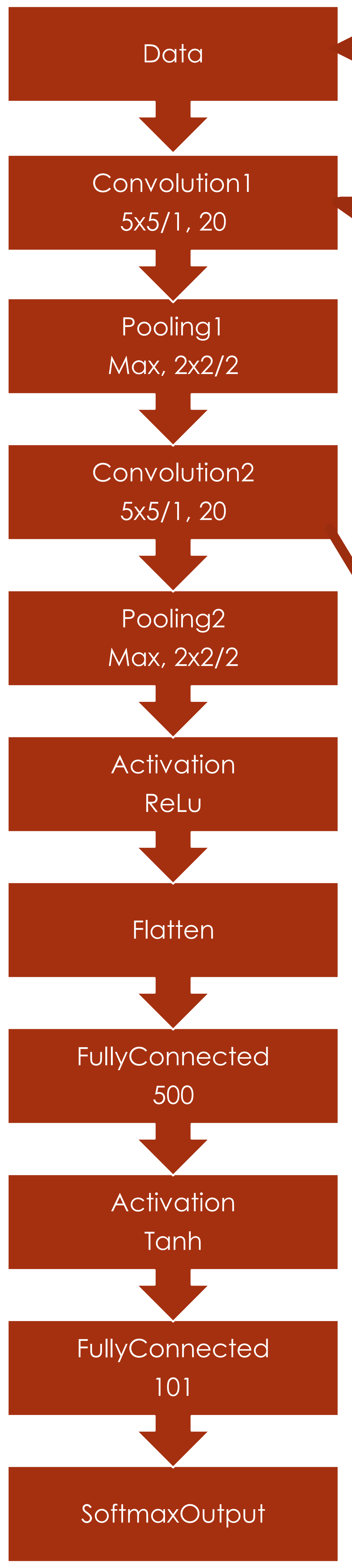


10	15	10	15	10	10
15	10	10	10	15	10
100	100	100	100	100	100
100	100	100	60	60	100
100	100	100	60	50	100
100	100	50	60	50	60

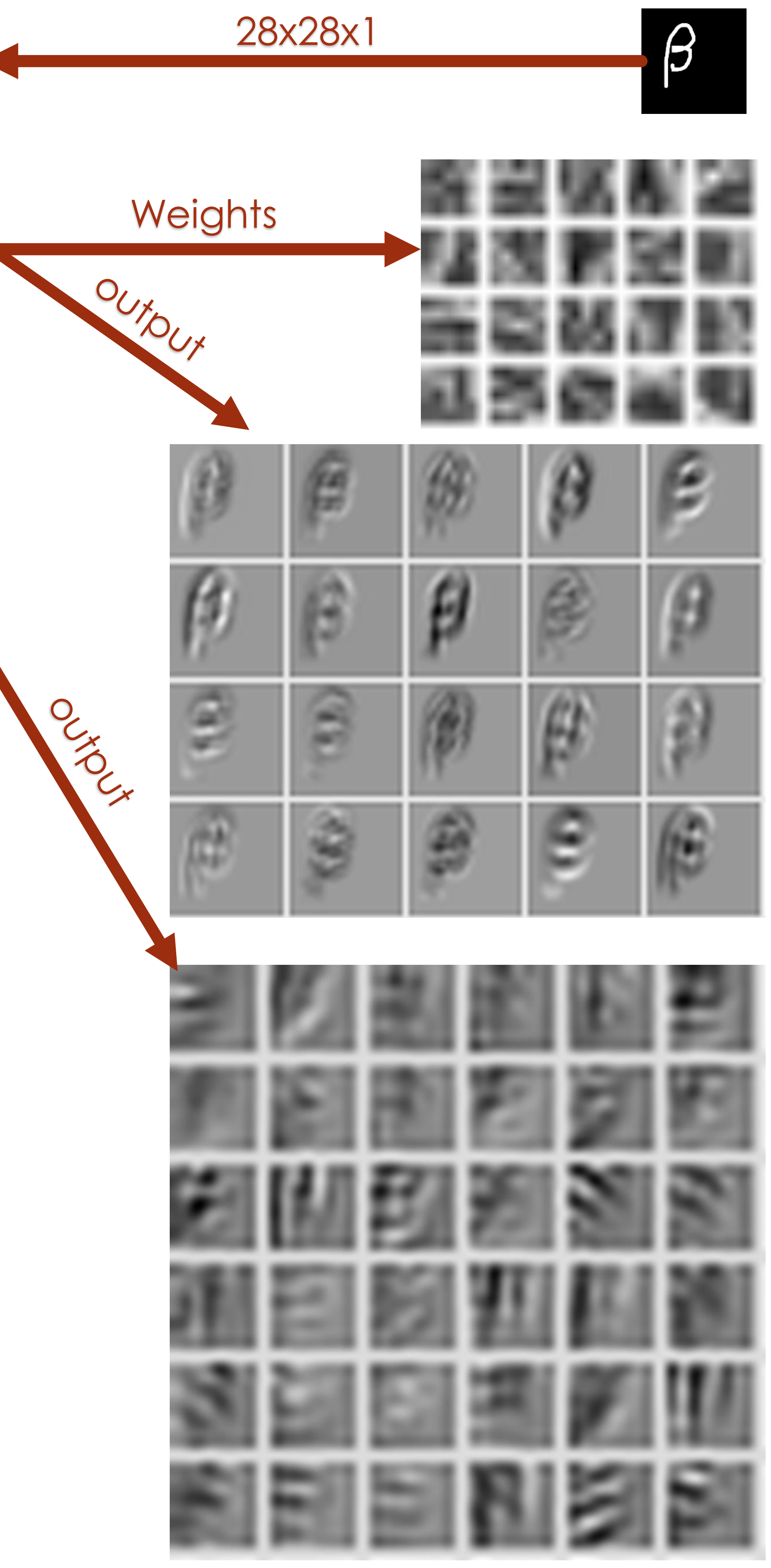
From: Proj. Crandall's B657 course ppt

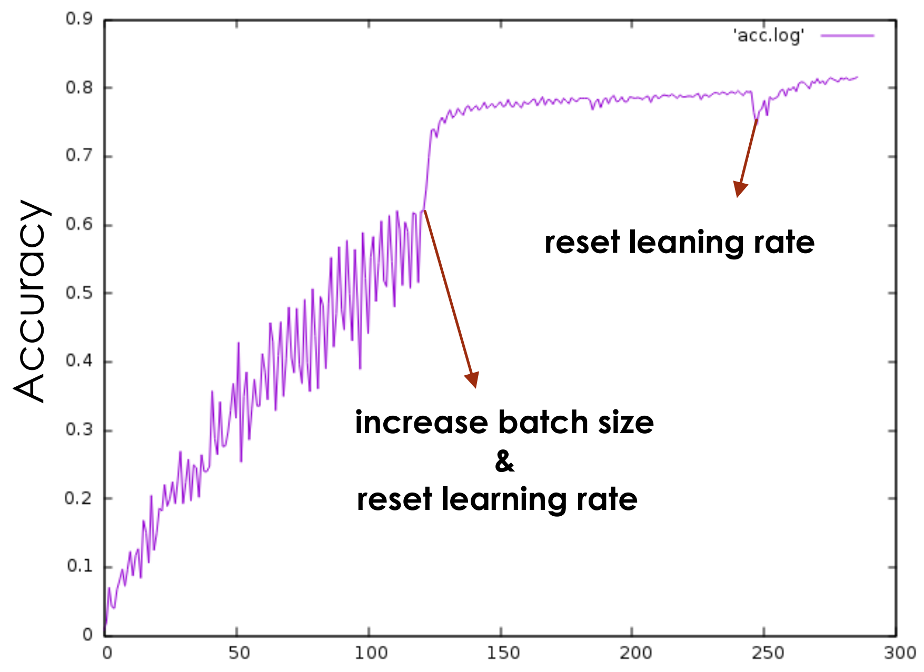
We use the spanning tree algorithm, which treat the image as a graph, and assign edge to every two adjacent pixels, according to their difference in color. After we construct the graph, we find the minimum spanning tree for this graph, and cut some edges with weights over a threshold to get a forest.

Convolutional Neural Network



Data → Convolution1 5x5/1, 20 → Pooling1 Max, 2x2/2 → Convolution2 5x5/1, 20 → Pooling2 Max, 2x2/2 → Activation ReLu → Flatten → FullyConnected 500 → Activation Tanh → FullyConnected 101 → SoftmaxOutput





Tech we used

Caffe