Predicting ocean health One plankton at a time

Abhilash Kumar and Peeyush Agarwal

Department of Computer Science and Engineering, IIT Kanpur

Overview

Motivation

- Planktons are critically important to our ecosystem
- Their population levels are an ideal measure of the health of world's oceans

Objective

- To create an algorithm that given an image, assigns class probabilities for various plankton classes.

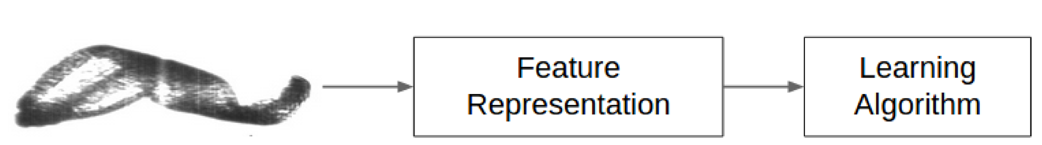
Dataset

- 121 plankton classes
- 30,000 training images and 130,000 test images

Challenges

- Many different species with varying size
- Image can have any orientation within 3-D space
- Ocean replete with detritus
- Presence of “unknown” classes

Methodology



Feature Extraction

- SIFT, GIST, HOG
- Feature Learning

Learning Algorithm

- SVM
- SoftMax

Random Forest

- Extract portion of image containing plankton
- Re size images to 25x25
- Extract Width to Height ratio
- Use pixels and extracted feature for training

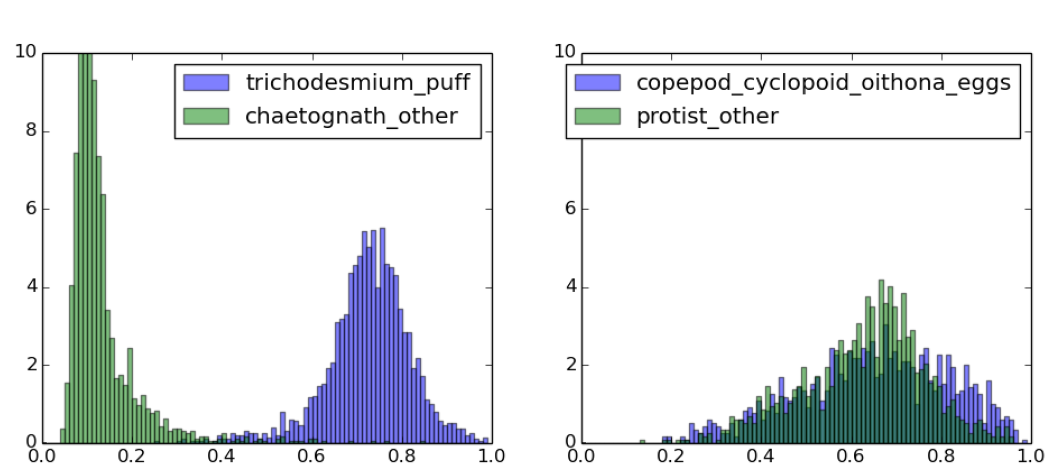


Figure 1: Class separation based on width to height ratio

Convolutional Neural Network

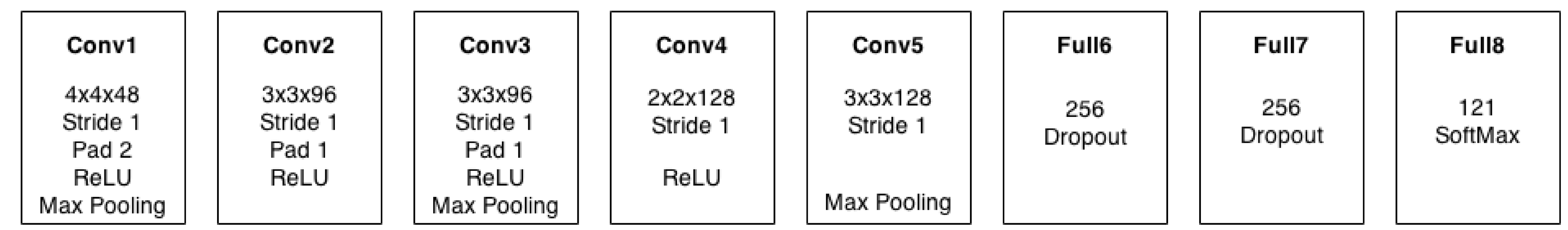


Figure 2: CNN Architecture

Network Description

- Similar to Hinton's ImageNet architecture [1]
- 8 weight layers (5 conv. and 3 fully connected)
- Dropout in 2 layers to prevent overfitting
- ReLU activation function

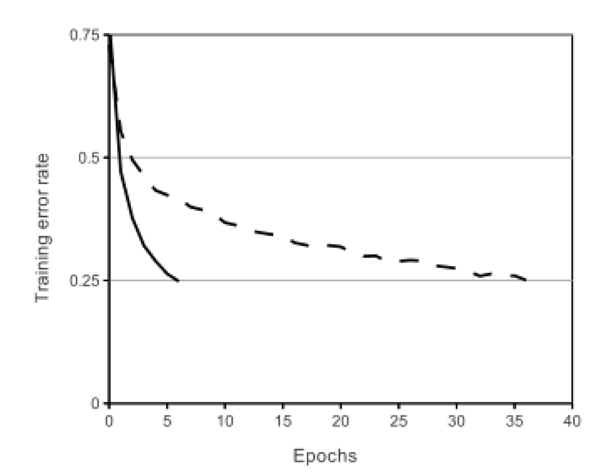


Figure 3: Training error rate with ReLU(solid line) and tanh(dashed line) neurons [1]

Training

- Input Size : 48x48
- Offline Data Augmentation
- 20 minutes training time on AWS GPU
- Run for 45 epochs

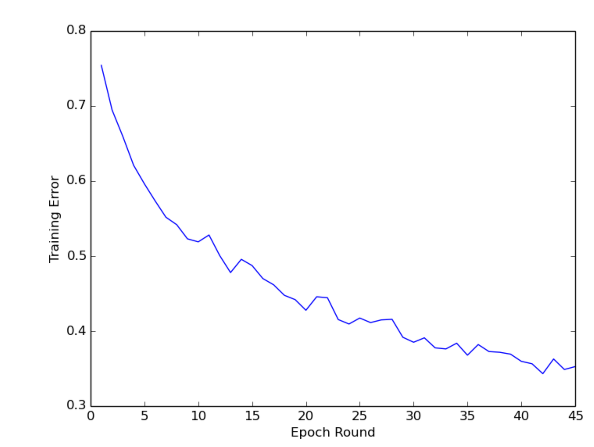


Figure 4: Training error v/s number of epochs

Maxout Network

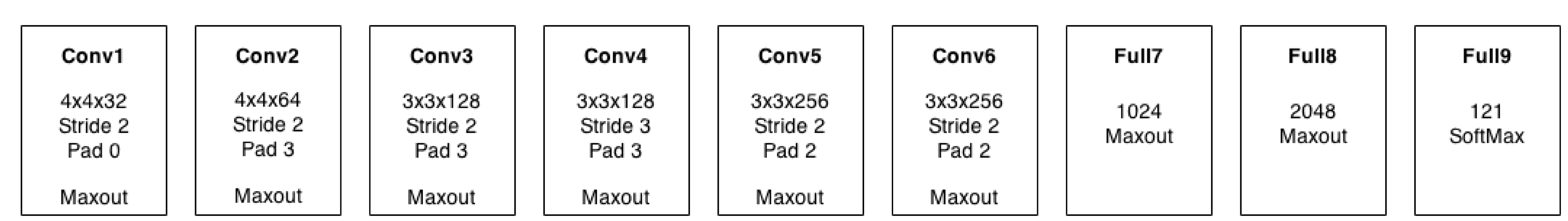


Figure 5: Maxout Network Architecture

Network Description

- Network similar to Bengio's Dropout Network [2]
- 9 weight layers (6 conv. and 3 fully connected)
- Maxout activation function

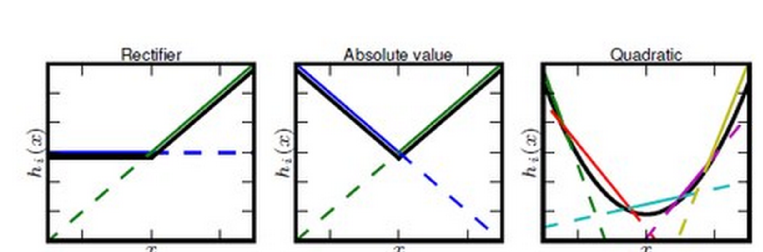


Figure 6: Graphical depiction of how maxout activation function can implement ReLU, absolute value rectifier, and approximate the quadratic activation function [2]

Training

- Input Size : 48x48
- Real time Data Augmentation
- 9 hours training time on AWS GPU

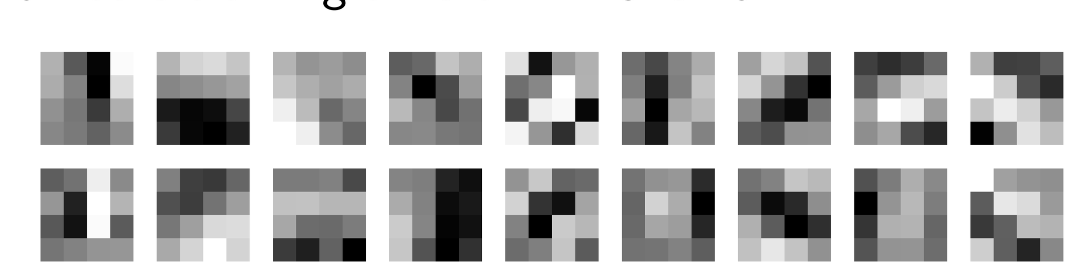


Figure 7: Visualization of weights learned in Convolutional layer

The network learns concepts like edge detectors, corner detectors etc. in addition to plankton specific filters.

Data Augmentation

- Offline data augmentation
- Real time data augmentation

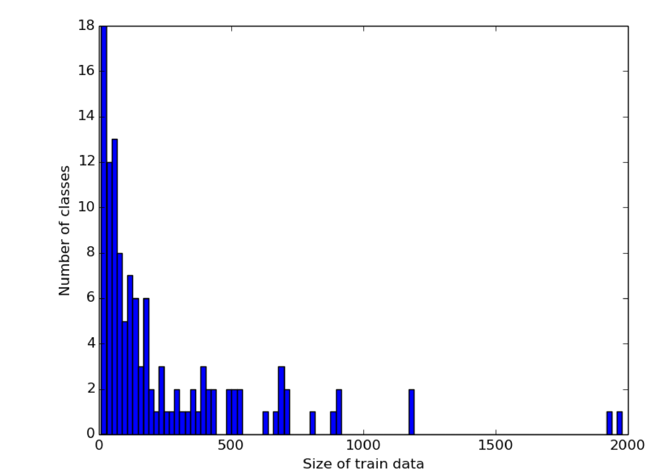


Figure 8: Histogram of size of training data for different classes

Results

Method	Accuracy
Random Forest	44%
CNN	61%
CNN + Dropout	65%
Maxout	71%

Table 1: Accuracy obtained for different methods

All evidences seem to suggest that training a deeper network with more data could improve our accuracy even further. The winning team in this contest obtained a classification accuracy of 81.52%. However, their winning model took 70 hours to train on an approx. 3 times more powerful GPU. Training on such large scales is currently infeasible for us.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[2] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

Acknowledgements

We acknowledge the authors of CXXNET, Theano and pylearn2 libraries. We also thank kaggle for providing dataset and random forest benchmark code. We also acknowledge Prof. Mukerjee for his guidance.