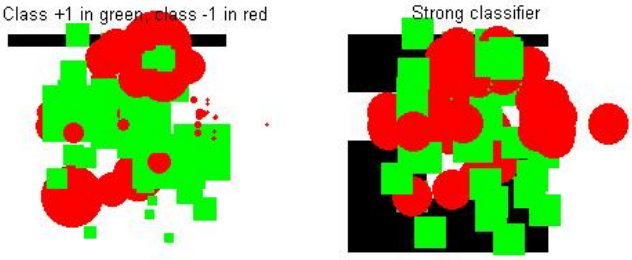
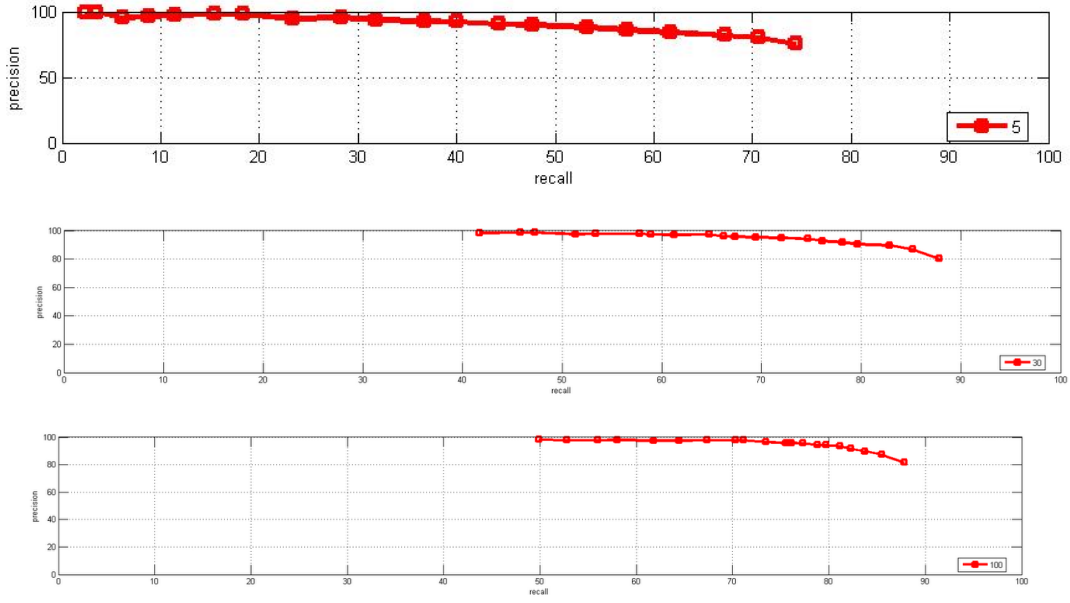
Adam Krebs
Computer Vision HW #4
5/9/12
Professor Fergus

Part 1:

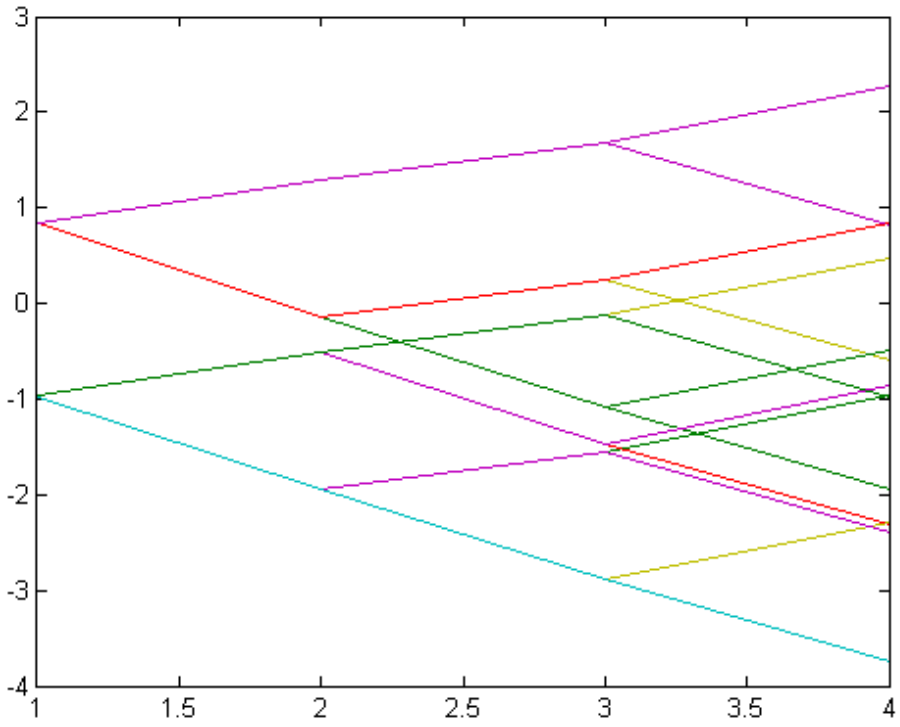
1.



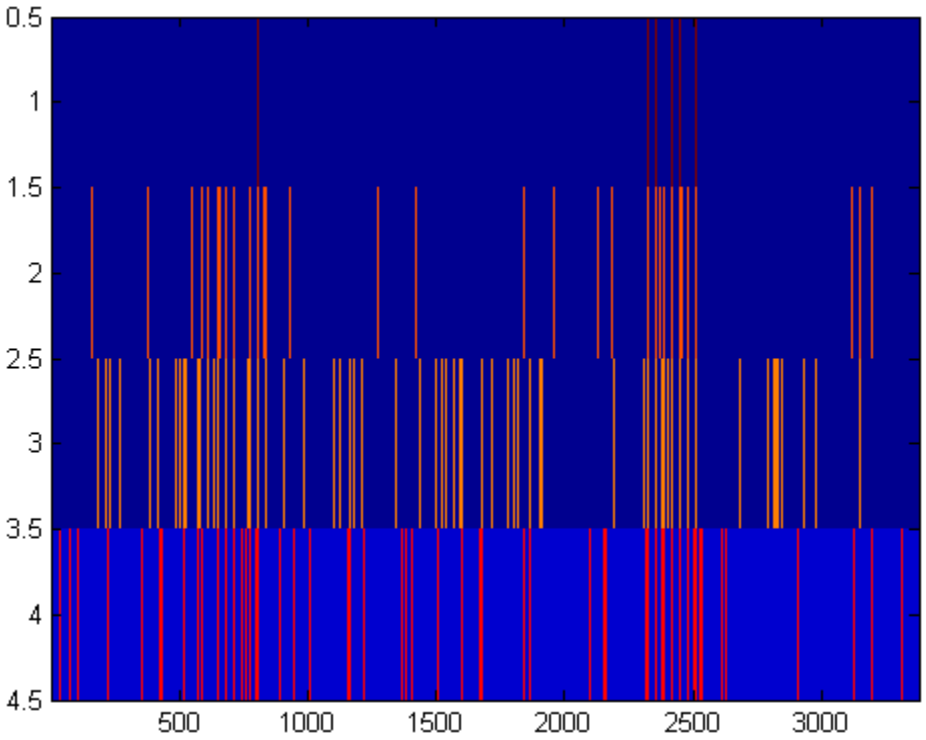
5, 30 and 100 weak classifiers:



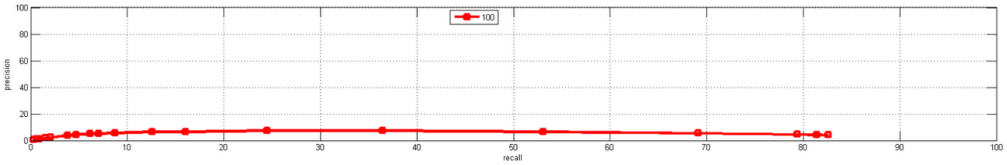
2-4. After boosting:



5. Weak classifier:



With randomized th:



6. There are different optimal histograms for different dimensions, and we have to compare classifiers across all dimensions. One would think that scrambling the dimensions would show wide variations between the different features, but actually the results are fairly close across all dimensions. When we change the parameters, it doesn't break the classifier completely (e.g. it doesn't go just to chance.) The classifiers work fairly well even in situations they weren't designed for.

7. In boosting, if we have 10 different classifiers, the weights are set each round differently. what was optimal for one dimensions for one round of boosting might not be optimal for another round because the weights have changed. We may need to revisit the same dimension again because the weights have changed on the points our classifier is visiting. Each has four parameters: a, b, th, k. When we fit greedily we only need to fit in a 4D space. but with 10 rounds of boosting in a global solution our dimensions would increase to 40, and is therefore a much harder problem.

8.

$$F(x) - Strong \qquad W^1 = W^0 \times e^{-(y \times f_1(x))}$$
$$f_1(x) - Weak\,1 \qquad W^2 = W^1 \times e^{-(y \times f_2(x))}$$
$$f_2(x) - Weak\,2 \qquad W^2 = W^0 \times e^{-(y \times f_1(x))}$$
$$W^2 = W^0 \times e^{-(y \times F(x))}$$

The strong classifier is composed of the non-greedy runs that the greedy weak-classifiers make up. The Strong Classifier therefore finds the best result using the output of the weak classifiers as its result.

Part 2:

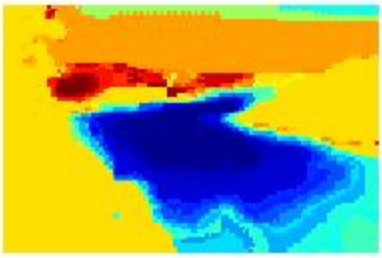
Window size .5:

Input Segmented

Window Size .3:

Input

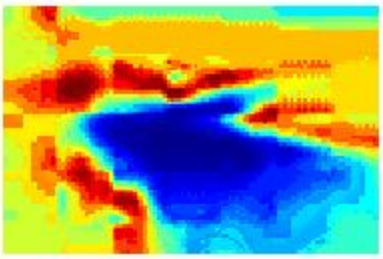
Segmented



Window Size .1:

Input

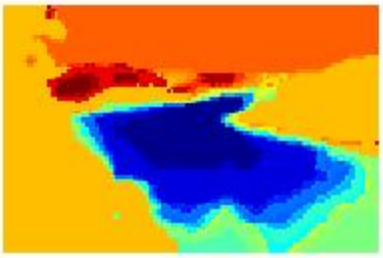
Segmented



Window Size .4:

Input

Segmented



I think the .4 window size was the best at capturing without overfitting the data. The lake in this picture is tricky, since it's got so many tones, but if we're trying to filter out lake from trees, mountain, etc, I think the .4 is the best choice.