

Overview

Andrew Showers

I was able to significantly reduce the amount of time computing the histogram by implementing numpy's multidimensional histogram function, histogramdd. This allowed multi-class classification, where if there are more than two labels, the classifier defaults to building multiple one vs. the rest classifiers. This allowed me to simplify my code by using sklearn's implementation instead of defining multiple classifiers.

I decided to use cost = 5.0 after experimenting with various other values. I also found that $b_h = b_w = 4$ was ideal. Any larger and the results seemed to degrade. t on the other hand could be increased significantly and provide marginal improvements. These improvements came at the cost of compute time due to the increased size of the feature vectors. Below are two experiments, one with $t = 4$, and another with $t = 16$. Error rates are shown as well as the total runtime.

Performance Results

Classification Performance (Experiment 1):

Parameters: $t = b_h = b_w = 4$, $C = 5.0$

CLASS	ERROR RATE ON TEST DATA
GRASS	36.00%
OCEAN	25.50%
RED CARPET	10.00%
ROAD	35.00%
WHEAT FIELD	37.00%

Average Error Rate: 28.70%

Total Runtime: 89.69 Seconds

Classification Performance (Experiment 2):

Parameters: $t = 16$, $b_h = b_w = 4$, $C = 5.0$

CLASS	ERROR RATE ON TEST DATA
GRASS	28.50%
OCEAN	25.00%
RED CARPET	10.00%
ROAD	34.00%
WHEAT FIELD	35.00%

Average Error Rate: 26.50%

Total Runtime: 199.03 Seconds