CS591.003 Assignment 4

Bryan Topp

November 28, 2014

1 Parameter Choice

Assume that we want no more than 20% false positives or negatives in collisions, so $P_1 = 0.8$ and $P_2 = 0.2$. Thus, our ρ parameter is 0.1386, meaning we should have $100000^{0.1386} \approx 5$ hash functions. We should have $\frac{\log 100000}{\log 1/0.2} \approx 8$ bits in each. Thus, each of the five hashes will need 256 bins. We will thus need memory for 1280 lists of data points, comprising 500,000 total list entries.

2 Implementation

The locality-sensitive hashing algorithm was implemented in C. The algorithm given in class selects hash bits randomly from a concatenation of unary representations of each dimension. This means, equivalently, each hash bit is a comparison with a random threshold value in a random one of the dimensions. This was implemented directly, without the intermediate conversion to unary first. Each color channel of each pixel is treated as a separate dimension ranging from 0 to 255. The color space is not transformed in any way. In querying, all collisions are collected up-front and deduplicated, to avoid comparing with the same image more than once (if a collision with it occurs in multiple hashes). Dataset images are loaded and discarded individually as hashes are computed; they are re-loaded on demand when a query needs them. For exact comparisons between a query image and a dataset image, a Euclidean distance is used in the aforementioned dimensions.

3 Performance

Using the parameters as calculated, there were an average of 25993 actual image comparisons performed for each query. This is a large number, but is only around one quarter the total dataset size. Additionally, the chosen parameters should guarantee a high probability of getting a good match. If many queries were to be performed on larger images, this would represent a significant savings.

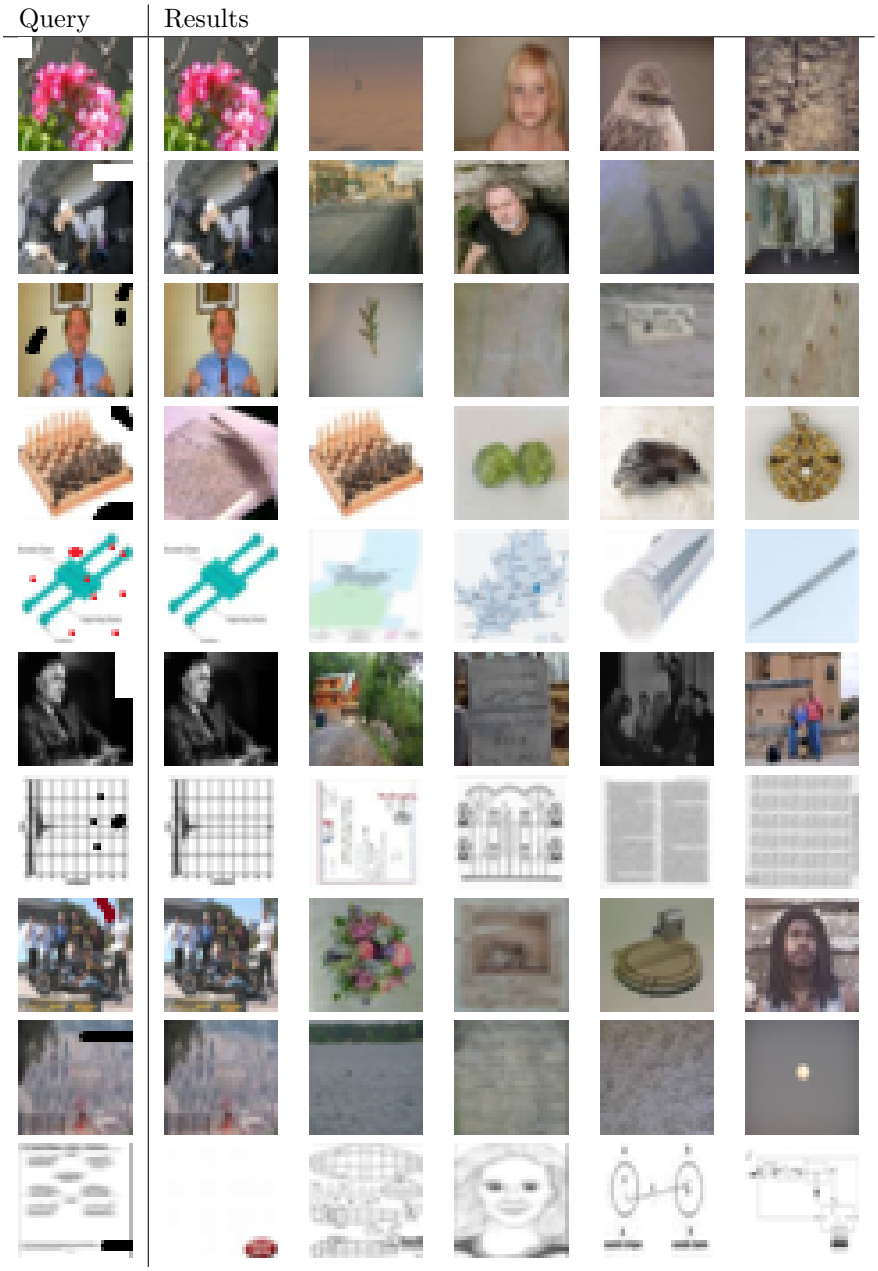
In this case, on a single hardware thread of an Intel Core i7 920, the hashing and queries together took just over 11 seconds. The total memory use was under 4 MB.

An alternate set of parameters was tried, doubling the size of each hash to 16 bits and reducing the number of hashes to 4. With this modification, an average of 3820 real comparisons were made per query. This is a very significant reduction. Memory usage is slightly higher at around 5 MB; run time is drastically reduced to 4 seconds.

4 Results

The top 5 matches for each query are shown below.

4.1 5 hashes of 8 bits each



4.2 4 hashes of 16 bits each

Query	Results