1. **Introduction**

In this homework, we are given 20 different classes and each of them having 20 to 50 instances. For each class, 20 images are randomly selected. Using descriptors, we create a codebook with varying codebook size based on the whole training data and encode each image in the training set according to the codebook. The encoding is histogram of codebook assignments of the image. The same procedure is done for test images and once their encoding is complete, a NN classifier based on histogram intersection kernel is applied.

Each image is normalised to 128x128 size.

The descriptors are sparse and dense SIFT. For descriptors, VLFeat library is used.

Codebook generation is done via kmeans clustering. Kmeans is applied to the whole training set. The resulting cluster centres are codebooks. Kmeans is applied with 100 and 500 cluster numbers.

Having generated the codebook, each test image is encoded. A histogram showing cluster centre assignments for a given test image descriptor is the histogram encoding. There is also a binary encoding, which is nothing but the binary version of the histogram encoding. For test images, histogram and binary encoding is obtained as well.

Lastly, test images are compared with train images in terms of histogram kernel matching metric. This metric is:

Using this metric, we search for the nearest neighbour by finding the maximum intersection; and assign the label.







1. **Results**

|  |  |
| --- | --- |
| sift\_histogram\_100 | 0.38 |
| sift\_binary\_histogram\_100 | 0.31 |
| dsift\_histogram\_100 | 0.93 |
| dsift\_binary\_histogram\_100 | 0.03 |
| sift\_histogram\_500 | 0.79 |
| sift\_binary\_histogram\_500 | 0.77 |

While vl\_sift returns a framex128 matrix for each image (and frame size varies between 20-50 frames) vl\_dsift returns a 14161x128 matrix. As mentioned above, these matrices are stack side by side to build a big train matrix, where vl\_kmeans is run. Unfortunately, due to high size of vl\_dsift matrix (for 20x20 images, the matrix becomes 5664400x128) I cannot run vl\_kmeans for k=500.

The table shows only the single-level matching. I will try to implement spatial pyramid matching.

Based on this table, the best option looks using dense sift descriptor. This might be no surprise because dense sift descriptor returns all the points in the image. Sift on the other hand selects interest points. Sift uses ~1/100 amount of points compared to dsift; hence it’s performance is worse than dsift.

It’s interesting to see that, for sift, histogram and binary histogram encodings yield close results; however, for dsift binary histogram encoding performs very poor. This is because disft populates almost all the cluster points when k=100 (because each dsift descriptor has 14161 points) and binary histogram encoding becomes almost a 1’s vector. The performance of dsift becomes superior for histogram encoding.

Finally, when k is increased from 100 to 500, sift descriptor performance increases. The same might be true for dsift.

Below is the Heat Map of best results from the table.

../heatmap_sift_500.pdf../heatmap_dsift_100.pdf