**Image Classification**

**Goal**

The goal of the assignment is to train the program using the given dataset so that it is intelligent to categorize query image.

**Performance**

My implementation is able to classify images into its corresponding category at 30.17% of accuracy

under 2 mins.

**Algorithm**

Image classification is archived in two main phases. Train and Test

**Train Phase**

1) Detection of interest points.

2) Extraction of the Local features.

3) Creation of Codebook.

4) Generate feature descriptor.

5) Determine tf\_idf.

6) Store data for testing.

**Test Phase**

1) Generate feature descriptor of query image

2) Classify image

**Train Phase**

*1) Detection of interest points*

The interest points in given images are detected using inbuilt Matlab function detectSURFFeatures.

*2) Extraction of the Local features.*

The local features of all detected interest points are extracted using Matlab inbuilt function

extractFeatures. SURF features are extracted of dimension 64.And the color histogram of r, g and b

channels are appended.

Local Feature (Dim 64)

B(Dim 24) Dim 64

G (Dim 24) Feature. Dim 64

R (Dim 24)

*3) Creation of Codebook (Bag of words)*

This the process of generating Bag of words i.e. collection of visu words

Codebook of features is created by clustering all features of all data images. Clustering is done by Matlab inbuilt function k-mean. ‘K’ value is set 100 by empirical analysis.

*4) Generate feature descriptor*

Feature Descriptor is K \* D 2d array, K number of clusters and where D is descriptor dimension.

Feature Descriptor of each image is the histogram of features similar to visu words. Similarity is found by Euclidian distance, smaller the distance more similar it is.

*5) Determine tf\_idf*

tf\_idf are generated for each feature descriptor for better accuracy.

But it is not used as there was no major improvement in accuracy.

6) Store data for testing

Feature Descriptor, label of image, codebook and idf are saved for use in testing.

**Test Phase**

*1) Generate feature descriptor of query image*

Same 1 – 4 steps as of Train are followed to generate feature descriptor of query image.

*2) Classify image*

Query image is classified based on the similarity of its feature descriptor with that of feature descriptors of training data. Again here Euclidian distance used to measure similarity. Cosine of the angle between two descriptors was also tried but there was better accuracy with Euclidian distance.

Extras

*1) Sift implementation*

I have implemented my own feature extractor based on sift (my\_sift.m).

Result:

Implementation is slow and accuracy is less (11%) then Matlab function, I’m not using this.

*Reason for slow execution of my\_sift:* Many *for* loops.

*Reason for low accuracy of my\_sift:* Incorrect orientation of patch along dominant gradient. Once the

patch is oriented we need to read gradient occupied by oriented patch. This was a difficult task to

implement in limited time, and hence I have implemented a simple hack which cost the accuracy.

*2) Color as feature*

Implemented color as feature by taking of r, g and b channels separately for the bin size of 24.

Result: Using color as a feature increased accuracy by 3%.

*3)tf\_idf*

Implemented tf\_idf.

Result: No major improvement in accuracy, hence not used.

**Challenges**

*Concept*: Understanding all the pieces of puzzle took some time.

*Debugging:* Because of huge data debugging was difficult.

*Fine Tuning:* I had to run program many time to select proper value for constants like ‘K’, descriptor dimension size etc, this was difficult since execution time was very high.