HOMEWORK ASSIGNMENT # 5
DUE: Tuesday, November 25, 2014
CSCI574: Computer Vision, Prof. Nevatia
Fall Semester, 2014

This is a programming assignment to implement and test a "bag of features" method for object recognition. The described method is a simpler version of the method described in the paper by Csurka *et al* and discussed in class.

You are given a set of 20 training images each for five object categories (car, cougar, face, pizza, and sunflower) and 10 test images of each category. Your task is to implement a classifier using bag of features approach and test it on the given images. The training images and test images are in .jpg format and can be downloaded from the class website.

Following are the steps in construction and use of the method:

a) Local Features: Compute SIFT features for each image using the OpenCV function (or any other available to you.

SIFT features are 128-dimensional; to reduce complexity, you are asked to perform principal component analysis (PCA) on these features. PCA can be implemented easily in OpenCV as functions for computing co-variance matrix and eigenvectors are available. These functions are as follows:

calcCovarMatrix: Calculates the covariance matrix of a set of vectors

eigen: Computes eigenvalues and eigenvectors of symmetric matrix

For details of the above functions, refer to

http://docs.opencv.org/modules/core/doc/operations_on_arrays.html?highlight=calccovarmatrix#void calcCovarMatrix(const Mat* samples, int nsamples, Mat& covar, Mat& mean, int flags, int ctype)

http://docs.opencv.org/modules/core/doc/operations_on_arrays.html?highlight=eigen#bool eigen(InputArray src, OutputArray eigenvalues, OutputArray eigenvectors, int lowindex,int highindex)

Once the principal directions (eigenvectors corresponding to some number of largest eigenvalues) have been computed, the new feature vectors are determined by taking a dot product of these principal directions with the original feature vectors. It is recommended that you use about 20 components though this choice is left to you. We can call the new features as the PCA-SIFT features.

b) Codewords: Cluster the PCA-SIFT features from all the training images (regardless of their category) by using the k-means clustering algorithm. The means of the clusters define the *code words* for vector quantization. Choice of the number of clusters may have a critical effect on the performance of the recognizer. You should test with a few values; it is

recommended that you initially choose k to be around 100. For each feature in each image, find the codeword closest to it and assign it the code (or label) of that codeword.

c) Object Feature Vector: Compute a histogram of code words for each image; this defines the feature vector for the image.

d) Object Recognition: Compute the feature vector for each test image and use it to classify the category of that image. Note that the objects are not segmented from the background in this method; the entire image is classified as one. It is recommended that you use the *n*-nearest neighbor classifier. A suggested modification is that vote of each neighbor is weighted by its distance from the sample to be classified. The number of neighbors to be used is another parameter that you are encouraged to experiment with; we suggest that you set it to be no less than 10.

e) Error Analysis: For each test image, verify whether the given answer is correct or not. Show the detection rates for images in each category. Optionally, you may also want to compute a "confusion matrix".

f) Experimental Evaluation: You are encouraged to experiment with a number of parameters for each module (*e.g.* number of principal components, number of clusters and number of neighbors used in classification). Note that some steps, such as clustering may take significant computation time due to the dimensionality of the space and the number of data points. Thus, you may not have time to explore many combinations but you should try at least a few and see how the results are affected.

SUBMISSION:

You should turn in the following in softcopy:

1. A brief description of the programs you write, including the source listing
2. Step by step results for one example to illustrate your method.
3. A summary and discussion of the results, including effects of parameter choices.