CS 884 Homework 1 (due Sept. 17) *J. Košecká*

Bag of Words The goal of the assignment is to implement a system for bag-of-features image classification. The goal is to perform four-class image classification, with the four classes being airplanes, motorbikes, faces, and cars. The data file cs.gmu.edu/∼kosecka/cs884/project_data.tar contains training and validation subdirectories for each category. The training subdirectories contain 40 images each, and the validation subdirectories contain 100 images each. You will train your system on the training images and evaluate its performance on the validation set. Below is the description on the components of the system. Write up the final report of discussing the experiments and approach you have chosen.

1. Feature extraction and description:
   - Features produced by any SIFT detector and associated descriptor;
   - Features sampled by at a sampling grid of fixed size and computed at 3 different scales each associated with the SIFT descriptor. (Do not consider patches with low intensity variance).
2. Dictionary computation. Run k-means clustering (kmeans function in MATLAB) on a subset of all training features to learn the dictionary centers. A reasonable dictionary size is about 500, but feel free to experiment with several different sizes.
3. Feature quantization and histogram computation. For each feature in a training or a test image, find the index of the nearest code vector in the dictionary. You may want to use this code for fast computation of squared Euclidean distances between two sets of vectors (i.e., all descriptors in an image and the codebook). Following quantization, represent each image by the histogram of these indices (check out MATLAB's hist function). Because different images can have different numbers of features, the histograms should be normalized to sum to one.
4. Classifier training. Select the classifier or your choice. The simplest option for this part of the assignment is a k-nearest-neighbor (kNN) classifier. MATLAB has a knnclassify function, but it only appears to work with a few pre-defined distance functions. If you want to experiment with a different distance function such as chi2, you may have to implement your own kNN function. To improve your results you should try to train a support vector machine (SVM) classifier. MATLAB includes SVM training and testing functions: svmtrain and svmclassify. Alternatively, you can download an SVM package from the Web. Here is one SVM package that is fairly easy to integrate with MATLAB http://theoval.cmp.uea.ac.uk/∼gcc/svm/toolbox/ or http://www.csie.ntu.edu.tw/∼cjlin/libsvm/