EECS 442 Computer Vision, Fall 2014
Homework 5

Due on **Monday,** December 1**, 2014 at 11:59pm**
Please submit your assignment on ctools.

1 [30 points] Clustering

Implement the k-means clustering in RGB space as discussed in lecture (also in FP - Algorithm 14.5 in the first edition or Algorithm 6.3 in second edition). Test your implementation on segmenting the image *segmentation.jpg* using $k = 3$. Initialize the algorithm by randomly picking k points. Try at least 5 different random initializations and show corresponding segmentation results. Describe your implementation and attach your code as an appendix. Also, comment on your different segmentation results.

2 [70 points] Object Recognition

In this question you will implement an object categorization system. We will explore the use of the Bag of Words (BoW) technique to classify images of *faces* and *cars* categories. See the end of the section for instructions.

(a) [**40 points**] [*Standard Bag-of-Words*] Implement the following BoW algorithm: The BoW process involves two stages - training and testing.

Training:

(i) For training, we will use the first 40 images for each category (car and face) from the provided dataset.

(ii) For each training image, extract the SIFT features for the entire image. Use the SIFT detection algorithm provided in the attached VLFeat library.

(iii) Next, using all the features from the training images, derive k codewords using k-means clustering. To do this, you may use your implementation of k-means from problem 1, or alternately, use the k-means implementation from the referred VLFeat library.

(iv) Then, represent each training image using the BoW representation (i.e., histogram of codewords as discussed in class). To do this efficiently, you may want to assign SIFT features to codewords using the attached k-d tree function. Normalize your histogram vectors so the sum of the elements equals 1.

Testing:

(i) For testing, we will use the remaining 50 images for each category (cars and faces) from the provided dataset.

(ii) To classify each test image:

A. Generate a histogram of codewords in the same way as training.

B. Use a k-nearest neighbor classification scheme as discussed in class, where $k = 1$. To do this efficiently, you may want to use the attached k-d tree function.

(iii) Compute classification accuracy as follows. Count how many images are correctly classified (i.e., that are classified by your algorithm with the correct class label). Call this number N_{correct}. Classification accuracy = $N_{\text{correct}}/N_{\text{test}}$, where N_{test} = total number of testing images (faces + cars).

In your solution, you should

(i) Run the BoW with $k = 3, 5, 10$. For each value of k, return classification accuracy. Plot (accuracy vs k) and discuss your results.

(ii) Report the time taken for training and testing for each value of k.

(iii) Describe your implementation in words, any changes to the above described procedure and references to code segments.

(iv) Attach all code written as an appendix.

(b) [**30 points**] [*Spatial Pyramid Matching*] In the standard BoW algorithm, images are represented by an orderless collection of local features. To further improve the result, we can exploit the information about the spatial layout of these features. Spatial pyramid matching (SPM) [1] is a scheme derived from this idea. In this problem, we ask you to implement a simple version of SPM based on BoW classification. Repeat the BoW algorithn with the following difference:

(i) For each traning image, divide it equally into a 2×2 spatial bin.

(ii) For each of the 4 bins, extract the SIFT features and compute the histograms of codewords as you did in (a).

(iii) Concatenate the 4 histogram vectors in a fixed order. You will get a vector with dimensionality $4k$.

(iv) Concatenate this vector with the histogram vector you got from (a), i.e. the histgrom for the whole image. Note that you should weight each of them by $\frac{1}{2}$ before you concatenate them. Your new representation of this image will be this concatentated vector with dimensionality $5k$.

(v) Represent the test images using the same way above, and carry out the k-nearest neighbor classification scheme as you did in (a).

In your solution, you should

(i) Report the classification accuracy.

(ii) Describe your implementation in words, any changes to the above described procedure and references to code segments.

(iii) Compare results with those obtained with BoW and discuss the differences.

(iv) Attach all code written as an appendix.

Instructions on dataset, SIFT & others, and k-d tree:

Dataset: We will be using the faces and cars (from Caltech 101) provided with the assignment.

SIFT & Others: Use the attached VLFeat library. The ReadMe file contains instructions on how to setup. For instructions on usage of the various functions, open the file and read the help comment.

k-d tree: A *k-d* tree is a method for indexing higher dimensional vectors. The construction strategy used enables to expedite the process of comparing and searching for closest matches. The provided library contains an implementation of a *k-d* tree.

References

[1] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.