Homework 3
Scene Recognition with Bag of Words

Due date: 23:59 Sunday January 9th (2022)

The goal of this project is to give you a basic introduction to image recognition. Specifically, we will examine the task of scene recognition starting with a very simple method, e.g., tiny images and nearest neighbor classification, and then move on to bags of quantized local features.

Bag of words models are a popular technique for image classification inspired by models used in natural language processing. The model ignores or downplays word arrangement (spatial information in the image) and classifies based on a histogram of the frequency of visual words. The visual word "vocabulary" is established by clustering a large corpus of local features.

For this project you will be implementing a basic bag of words model. You will classify scenes into one of 15 categories by training and testing on the 15 scene database (introduced in [Lazebnik et al. 2006], although built on top of previously published datasets).

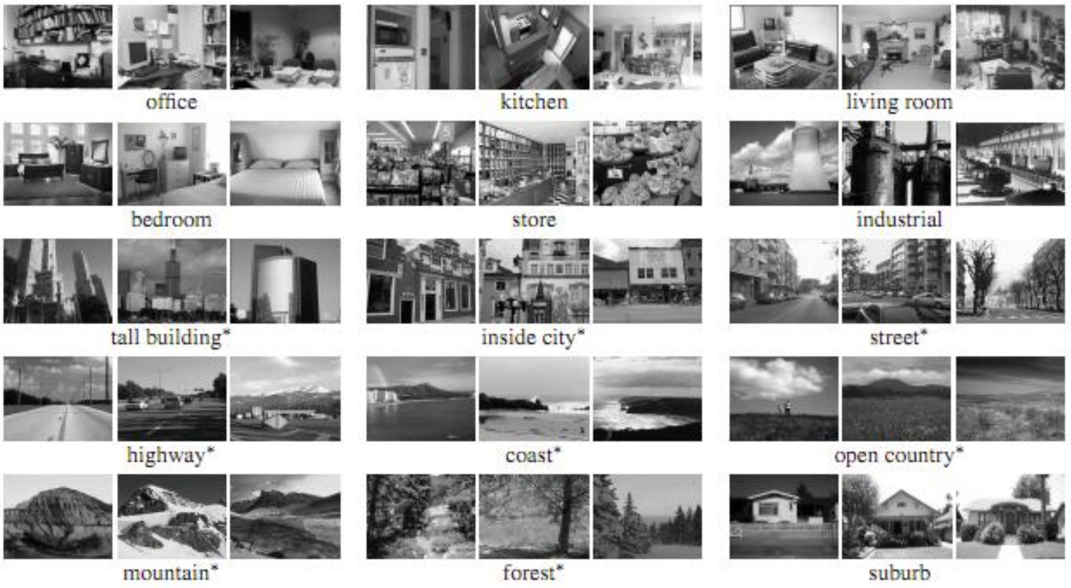

Figure 1. Example scenes from each category in the 15 scene dataset.

1. Tiny images representation

You will start by implementing the tiny image representation and the nearest neighbor classifier. They are easy to understand, easy to implement, and run very quickly for our experimental setup.

[10 marks]
"Tiny image" feature: The "tiny image" feature is one of the simplest possible image representations. One simply resizes each image to a small, fixed resolution (we recommend 16x16). It works slightly better if the tiny image is made to have zero mean and unit length. This is not a particularly good representation, because it discards all of the high frequency image content and is not especially shift invariant.

[20 marks]
Nearest neighbor classifier: The nearest neighbor classifier is equally simple to understand. When tasked with classifying a test feature into a particular category, one simply finds the "nearest" training example (L2 distance is a sufficient metric) and assigns the test case the label of that nearest training example. The nearest neighbor classifier has many desirable features: it requires no training, it can learn arbitrarily complex decision boundaries, and it trivially supports multiclass problems. It is quite vulnerable to training noise, though, which can be alleviated by voting based on the K nearest neighbors (but you are not required to do so). Nearest neighbor classifiers also suffer as the feature dimensionality increases, because the classifier has no mechanism to learn which dimensions are irrelevant for the decision.

Together, the tiny image representation and nearest neighbor classifier will get about 15% to 25% accuracy on the 15 scene database.

2. Bag of SIFT representation

After you have implemented a baseline scene recognition pipeline it is time to move on to a more sophisticated image representation -- bags of quantized SIFT features.

[20 marks]
Building the vocabulary of visual words: Before we can represent our training and testing images as bag of feature histograms, we first need to establish a vocabulary of visual words. We will form this vocabulary by sampling many local features from our training set (10's or 100's of thousands) and then clustering them with ***kmeans***. The number of kmeans clusters is the size of our vocabulary and the size of our features. For example, you might start by clustering many SIFT descriptors into k=50 clusters. This partitions the continuous, 128 dimensional SIFT feature space into 50 regions. For any new SIFT feature we observe, we can figure out which region it belongs to as

long as we save the centroids of our original clusters. Those centroids are our visual word vocabulary. You can use any existing libraries for kmeans.

[20 marks]
Building the vocabulary of visual words: Once the vocabulary is built, we are ready to represent our training and testing images as histograms of visual words. For each image we will densely sample many SIFT descriptors. Instead of storing hundreds of SIFT descriptors, we simply count how many SIFT descriptors fall into each cluster in our visual word vocabulary. This is done by finding the nearest neighbor kmeans centroid for every SIFT feature. Thus, if we have a vocabulary of 50 visual words, and we detect 220 SIFT features in an image, our bag of SIFT representation will be a histogram of 50 dimensions where each bin counts how many times a SIFT descriptor was assigned to that cluster and sums to 220. The histogram should be normalized so that image size does not dramatically change the bag of feature magnitude.

[10 marks]
Nearest neighbor classifier: You should now measure how well your bag of SIFT representation works when paired with a nearest neighbor classifier.

[20 marks]
Comparison: There are many design decisions and free parameters for the bag of SIFT representation (number of clusters, sampling density, SIFT parameters, etc.) so performance might vary from 30% to 40% accuracy. You can compare the performance with different parameters.

3. Grading

You are required to submit both your report and source code (Matlab or C/C++ or Python) to Learning in ZJU (学在浙大). Your report should be in the **pdf** format with no more than 6 pages.

For both the tiny image representation and bag of SIFT representation, your report should include a table to summarize the accuracy for each category and the average accuracy. You should also visualize the results by showing the examples with both correct and incorrect recognition results. A comparison table with different choices of parameters is also expected.

In the report, you are expected to discuss your findings through the experiment. For example, how will the vocabulary size influence the final classification? Which category has the highest/lowest recognition accuracy in this dataset and why? How the implemented algorithm can be improved?

Please zip everything together in **a single file**, and **name it with your student id** before uploading.