Project 3 Writeup

Instructions

- Provide an overview about how your project functions.
- Describe any interesting decisions you made to write your algorithm.
- Show and discuss the results of your algorithm.
- Feel free to include code snippets, images, and equations.
- List any extra credit implementation and result (optional).
- Use as many pages as you need, but err on the short side.
- **Please make this document anonymous.**

Project Overview

This project implements scene recognition with 3 methods:

- tiny image representation and nearest neighbor classifier.
- bag of words representation and nearest neighbor classifier
- bag of words representation and linear SVM classifier

Implementation Detail

I first implemented `get_tiny_images`. This function was straightforward to code by following the code template instructions. For each image path given in the input I loaded the image, scaled it to 16x16 pixels, reshaped it into a 1-D array, and appended all these features for each image together in a resulting array.

I then implemented `nearest_neighbor_classify`. After getting the distances between the test image features and the training image features, I sorted the distances using `np.argsort` and grabbed the first k elements of each row, resulting in a 2-D array where each row has the k indices of the closest distances for each image. The labels were then grabbed for each image and I used `stats.mode` to get the mode for each image.

Next was to implement `build_vocabulary`. I used `skimage.feature.hog` to get the HOG features of each image (after scaling them to 160x160 px). I used 4x4 pixels

per cell and 2x2 cells per block. Then I reshaped the descriptor into a list of 2*2*9 block feature vectors and added these features to the bag of words. After doing this for every image, I ran `MiniBatchKMeans` on the bag of words (setting max_iter to 200 and n_clusters to vocab_size) and returned the cluster centers.

For `get_bag_of_words`, for each image I similarly got the HOG features using the same settings as in `build_vocabulary`. Then I calculated the euclidean distances between the hog features and the input vocabulary. I used `np.argsort` to grab the index of the smallest distance for each feature and for each smallest distance I added 1 to that bin in the histogram.

Finally for `svm_classify` I used `sklearn.svm.LinearSVC` to fit the training image features and returned the predictions on the test features.

Result

1. Tiny image representation and nearest neighbor classifier accuracy: 20.5%
2. Bag of words representation and nearest neighbor classifier accuracy: 53.0%
3. Bag of words representation and linear SVM classifier accuracy: 50.1%

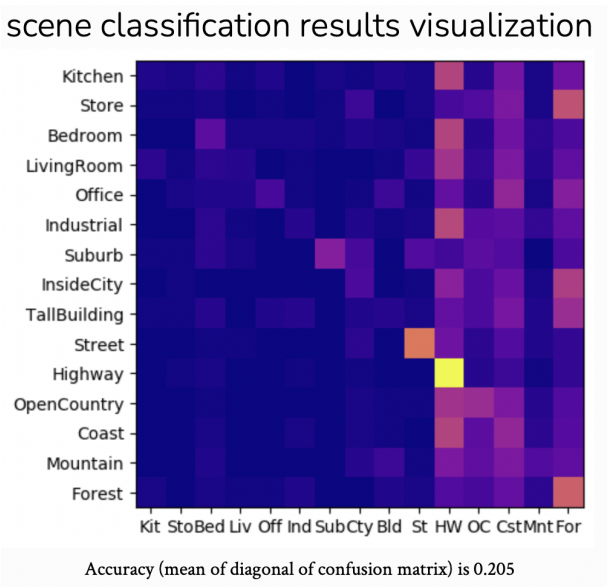

Figure 1: Tiny image and Nearest Neighbor

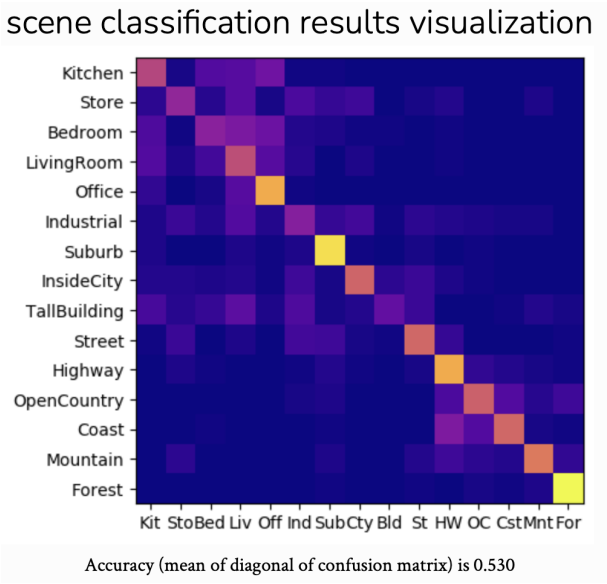

Figure 2: Bag of Words and Nearest Neighbor

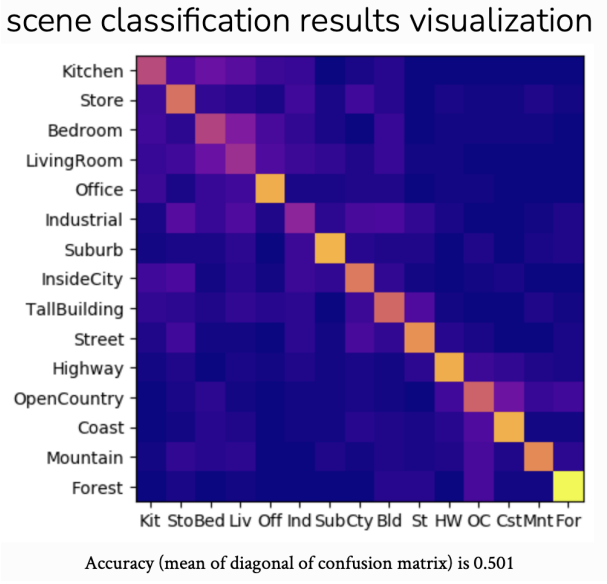

Figure 3: Bag of Words and Linear SVM

Extra Credit (Optional)

1. I trained the SVM with RBF, which increased the bag of words x svm accuracy to 61.9% The only thing I changed was:

```
    # linear classifier:
    # lin_clf = LinearSVC()
    # updating it to rbf classifier instead:
    lin_clf = SVC(kernel='rbf')
```

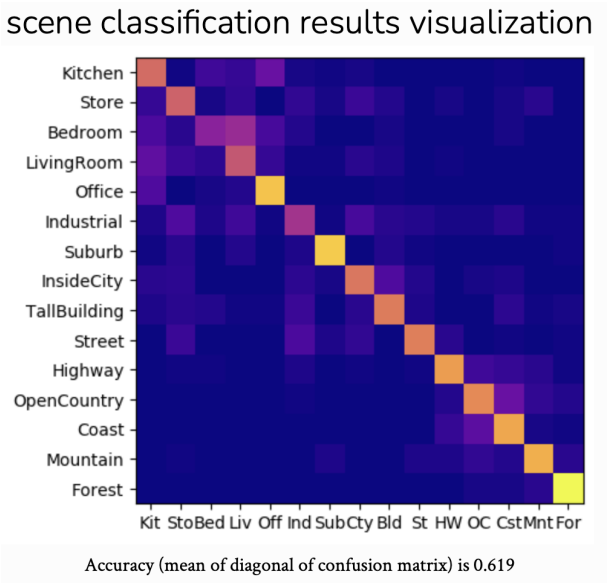

Figure 4: Bag of Words and RBF