Assignment 3

Qi Le 5674954

Tiny Image KNN Classification

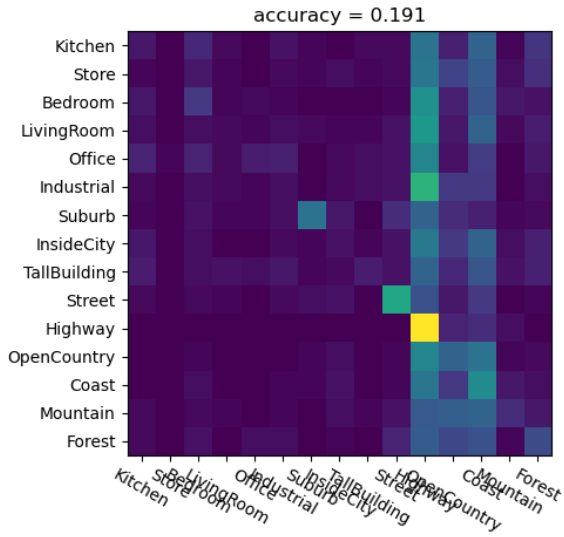

Figure 1: Confusion Matrix for Tiny Image KNN Classification

In **get_tiny_image()** function, I resize the image by dividing the image into small blocks(15*15) and averaging the pixels in each block. Then we normalize the vector in each block.

In **predict_knn()** function, I set the k equals to 8 and train the KNN classifer with training data. Then the KNN classifier is used to predict the testing data.

In **classify_tiny_knn()** function, I combine the aboving two equations. I calculate tiny image representation for all images and predict all the testing images. The fixed resolution is (15*15) and k for KNN is 8. Accuracy of the confusion matrix is 19.1%.

BoW+KNN

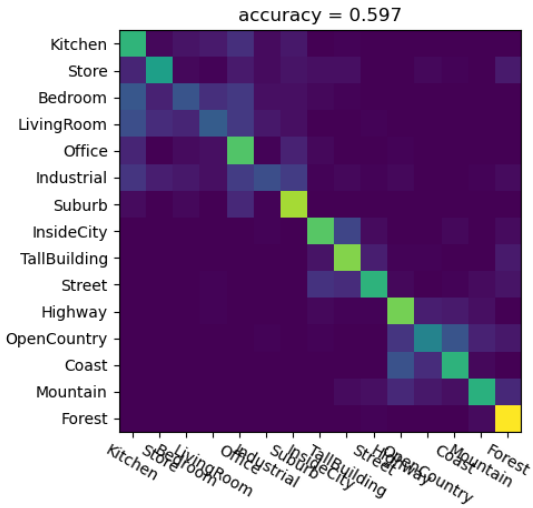

Figure 2: Confusion Matrix for BoW+KNN

In **compute_dsift()** function, I divide the image into small square blocks with length equals to 15. I compute SIFT descriptor of the center of each block at each stride, which is 15. These computations generate the dense feature (n, 128).

In **build_visual_dictionary()** function, I build a visual dictionary from the dense sift feature representation of training images using KMeans algorithm(center = 50). The parameters for training KMeans are: max iteration = 250, n_clusters = 50.
In **compute_bow()** function, the bag-of-word feature is constructed by counting SIFT features that fall into each cluster of the vocabulary.
In **classify_bow_knn()** function, all training dense sift feature representations are calculated and I build a visual dictionary based on the representations. Then bag-of-word features of all images are calculated using the dictionary. In the end, the KNN classifier is trained by the bag-of-word features. The parameter for training KNN classifier is k = 8. Accuracy of the confusion matrix is 59.7%.

BoW+SVM

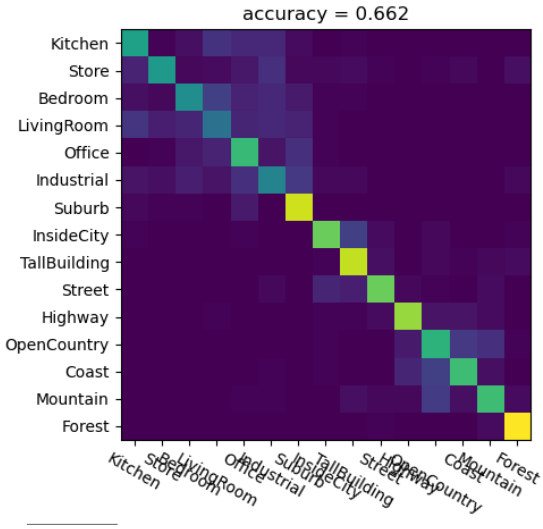

Figure 3: Confusion Matrix for BoW+SVM

In **predict_svm()** function, 15 binary, 1-vs-all SVMs, are trained. Predictions of the testing data are made from the classifier that owns the highest probability.
In **classify_bow_svm()** function, all training dense sift feature representations are calculated and I build a visual dictionary based on the representations. Then bag-of-word features of all images are calculated using the dictionary. In the end, the 15 SVM classifiers are trained by the bag-of-word features. The parameters for training SVM classifier are: total error = 1e-5, C = 1, kernel = “rbf”. Other parameters stay the same. Accuracy of the confusion matrix is 66.2%.