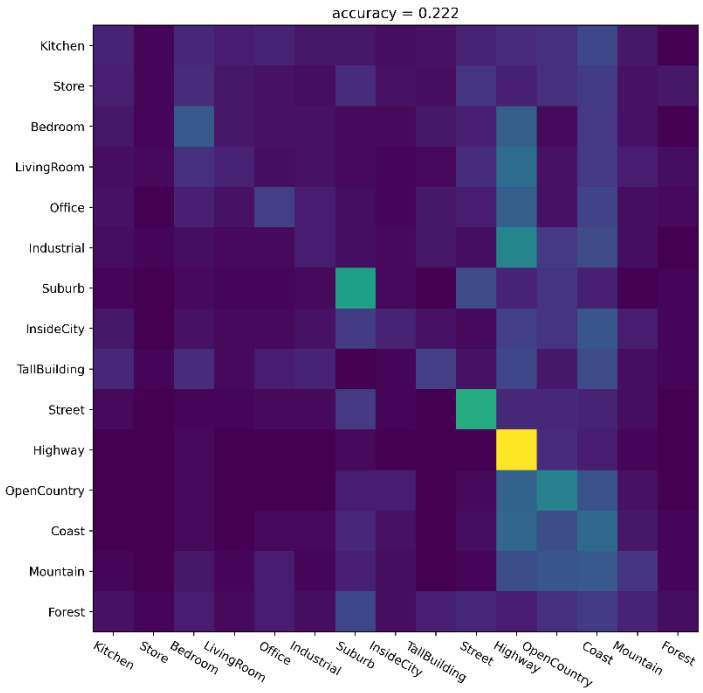
4. Tiny image KNN classification

All the functions get_tiny_image(img, output_size) and predict_knn(feature_train, label_train, feature_test, k) and classify_knn_tiny(label_classes, label_train_list, img_train_list, label_test_list, img_test_list) were written. At the end of it, main aim was to get more than 18% accuracy. I got 22.2% accuracy. With various different values we can get different accuracies.



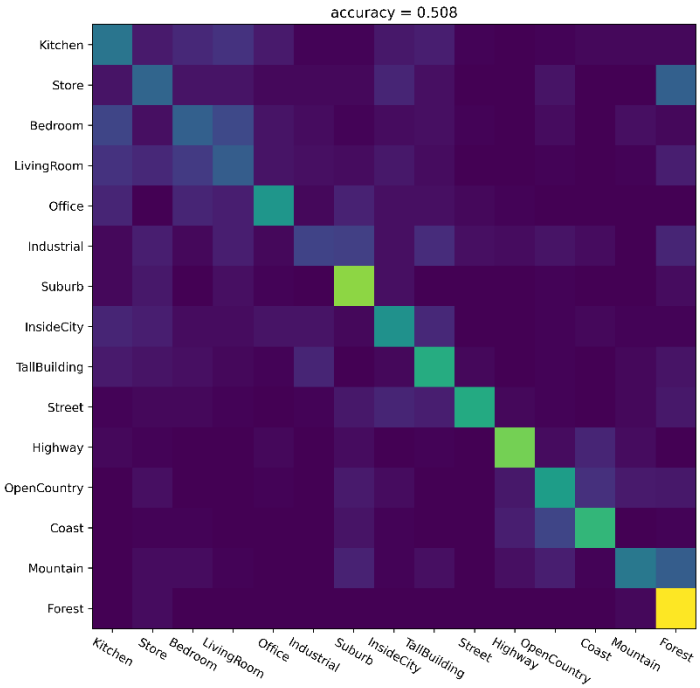
I made another function called classify_function which works for all classify functions in this assignment. But each individual classify function works with given requirements of input and output. So that I can comply with the specification rule.

5. Bag-of-word visual vocabulary

All the functions, compute_dsift(img, stride, size), build_visual_dictionary(dense_feature_list, dic_size), compute_bow(feature, vocab), classify_knn_bow(label_classes, label_train_list, img_train_list, label_test_list, img_test_list) were written. At the end of it, main aim was to get more than 50% accuracy. I got 50.8% accuracy. With various different values we can get different accuracies. I made another function called classify_function, which works for all the classify functions in this assignment. But that doesn't break the rule about individual classify functions. So, requirements were met and number of lines were reduced because of that.

I also added already trained vocab.txt file. So that you don't have to train again.

In build_visual_dictionary function comments were written to tell about the how to use already trained vocab.txt file.



6. BoW + SVM

All the functions, predict_svm(feature_train, label_train, feature_test), classify_svm_bow(label_classes, label_train_list, img_train_list, label_test_list, img_test_list) were written. At the end of it, main aim was to get more than 60% accuracy. I got 60.2% accuracy. With various different values we can get different accuracies. I made another function called classify_function, which works for all the classify functions in this assignment. But that doesn't break the rule about individual classify functions. So, requirements were met and number of lines were reduced because of that.