1 TINY+KNN

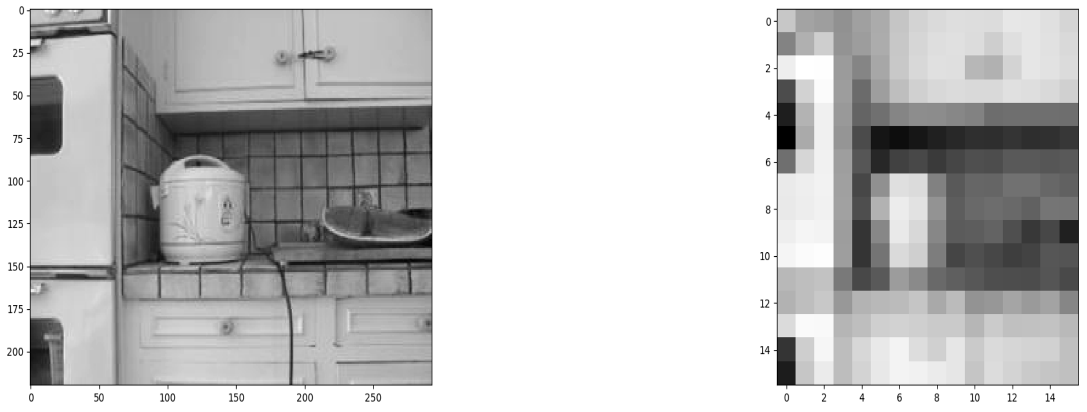

Figure 1: Tiny image representation of an image

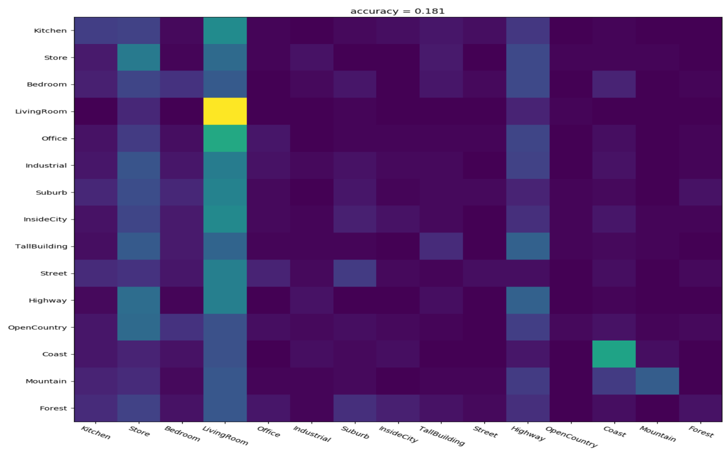

Figure 2: Confusion matrix for tiny image representation with KNN classification

In **get_tiny_image()**, image is resized to a smaller shape by averaging pixels of individual blocks. Then the image is reshaped into a vector. This vector is subtracted by its mean and divided by its norm to get the mean to zero and length to a unit. Figure 1 shows tiny image representation of a sample image.

In **predict_knn()**, KNN classifer is trained with training data. Then predictions are made on testing data using the given K.

In **classify_tiny_knn()**, for all training and testing images tiny image representation is calculated and then KNN classifier is used to predict labels for testing images using the following parameters, TINY_IMAGE_SIZE = (16, 16) K = 6 to get an accuracy of about 18.1% with confusion matrix shown in Figure 2

2 BOW+KNN

In **compute_dsift()**, image is divided into sqaure patches of side length (=given stride). For each patch we compute a SIFT descriptor of the lower right corner, with keypoint size (=size), which is of 128 dimensions. This way each image yields a dense feature of shape (n, 128).

In **build_visual_dictionary()**, given dense representation of images KMeans clustering is used on dense feature space to get clusters centers which correspond to visual words. This way a visual vocabulary is built.

In **compute_bow()**, given a dense feature representation of an image and exisiting vocabulary a histogram is built in which x-axis corresponds to vocabulary and y-axis corresponds to number of such visual words. This representation is called bag of words representation of image.

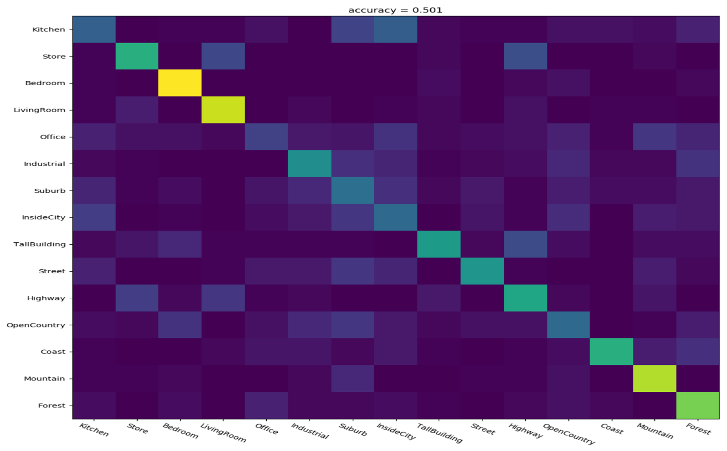

Figure 3: Confusion matrix for BOW image representation with KNN classification

In **classify_bow_knn()**, for all training images dense representation is calculated, a visual vocabulary is built and then bag of words representation for both training and testing images is calculated using the vocabulary. The bag of word representation of training images is used to train a KNN classifier and predictions are made for testing images. The exact parameters used are the following, KMeans: N_ITER = 10, MAX_ITERATIONS = 300, STRIDE = 32, SHAPE = 32, DIC_SIZE = 50, K = 8 to get an accuracy of about 50.1% with confusion matrix shown in Figure 3

3 BOW+SVM

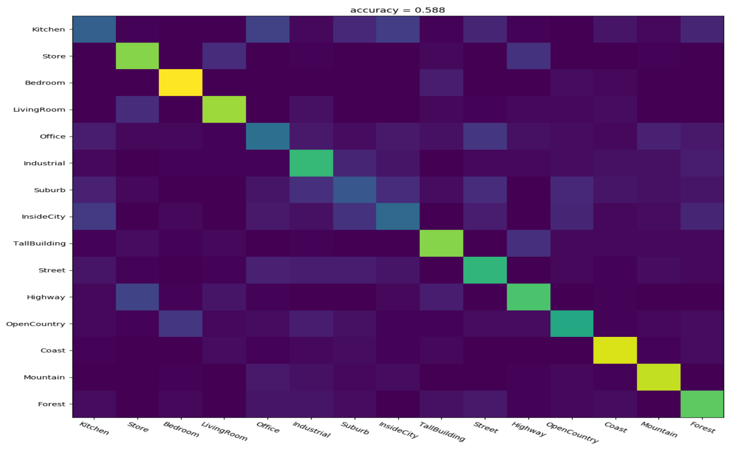

Figure 4: Confusion matrix for BOW image representation with SVM classification

In **predict_svm()**, n_classes (=15) support vector machines are trained using the trianing data using a one-vs-all fashion and predictions are made on testing data for each classifier. Each testing sample is assigned class of classifier that predicts highest probability.

In **classify_bow_svm()**, for all training images dense representation is calculated, a visual vocabulary is built and then bag of words representation for both training and testing images is calculated using the vocabulary. The bag of word representation of training images and training images are sent to predict_svm() function to obtain predictions on testing images. The exact parameters used are the following, STRIDE = 32, SIZE = 24, KMeans: N_ITER = 10, MAX_ITERATIONS = 300, vocab: DIC_SIZE = 64, SVM: kernel = 'rbf', C = 4, GAMMA = 'scale', to get an accuracy of about 58.8% with confusion matrix shown in Figure 4