CSci 5561 Assignment 3

Luis Guzman

Friday October 6, 2020

In this assignment, I implemented three different algorithms for the task of scene recognition. The first algorithm computes a 16x16 ”tiny image” feature for each image and uses k-nearest-neighbors for classification. To compute the tiny image, I set each pixel to be the mean of the corresponding chunk of the original image. I then normalize it to have zero mean and unit length. For the classifier, I used sklearn's KNeighborsClassifier, with a k value of 20. With this method, I am able to achieve an accuracy of 0.211. The confusion matrix is given in figure 1.

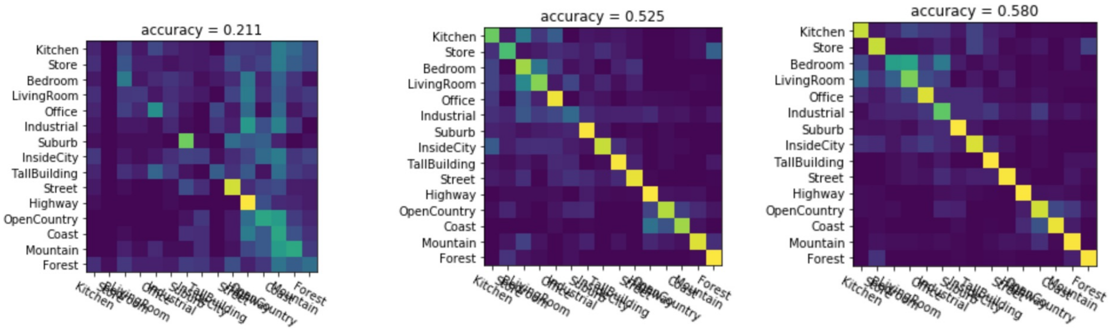

Figure 1: Confusion matrices for the (from right to left) tiny image, KNN+BOW and SVM+BOW classifiers

The next algorithm uses the Bag of Words (BOW) model, which greatly improves the accuracy. I start by computing the dense SIFT feature for each image. Dense SIFT varies from normal SIFT by computing the features at every pixel location (with some stride value), instead of at just the extracted keypoints. I found that a size and stride value of 16 gives the best accuracy rate. Next, I build the ”visual dictionary” (see Algorithm 1) by using k-means to calculate cluster centers of the extracted features. For this assignment, I set the number of clusters to be 50. Lastly, the Bag of Words feature is calculated for every image in the training and testing sets. This is calculated by building a histogram of how each dense sift feature is classified by the visual dictionary. The Bag of Words features are then used to train a KNN classifier. Accuracy for this method was 0.525.

Algorithm 1: Visual Dictionary Building

```
For each image, compute dense SIFT over regular grid
Build a pool of SIFT features from all training images
Find cluster centers from the SIFT pool using kmeans algorithms.
Return the cluster centers.
```

Lastly, I improve on the second algorithm by replacing the KNN classifier with an SVM by using sklearn's LinearSVC class. Using an SVM requires choosing an additional hyperparamerter λ, which controls how heavily the solution is regularized. I tried 40 different values of λ, ranging from 0.01 to 10 and found that $\lambda = 5.133$ gave the best accuracy for this model. The final accuracy was 58%, which is slightly below the expected 60%. I also tried altering the size and stride of the dsift function, but after around a dozen combinations, I found the best combination (size=16, stride=10) to still give an error rate of 58%. Some additional values I tested are given in Table 1.

Size	Stride	Accuracy (best λ)
16	16	0.570
16	10	0.580
10	16	0.560
20	16	0.557
16	5	0.573

Table 1: Accuracy values for the SVM model with different values of stride and size. I could not test more values due to the long (about 30 mins) time it takes to train

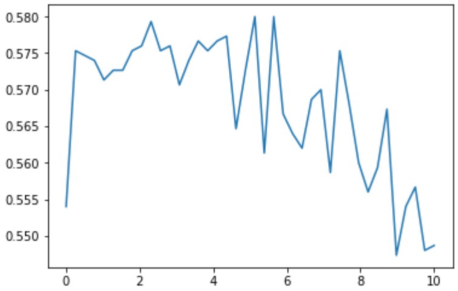

Figure 2: Plot of the model accuracy vs. labmda value for the SVM classifier