CSCI 5561 Homework 3

Scene Classification

Ashvin Pidaparti, 5421833

To make K-Nearest Neighbor classification faster, resolution of input images was scaled down, but the appropriate size was not known. Figure 1 shows the accuracy of K-Nearest Neighbors classification of testing images against multiple scaled resolutions, where the x axis is the new size lengths of tiny images and the y axis is the corresponding accuracy. We see a significant spike at 16x16, resulting in 18.4% accuracy. K-Nearest Neighbors has a free parameter k. The k was tested from 5 to 45 to determine the optimum. It was found, per Figure 2, that k did not significantly change the accuracy of the algorithm. Figure 3 shows the confusion matrix of K-Nearest Neighbors Classification of Tiny Images.

A dictionary of visual words was constructed from every training image and clustered using K-Means. Each training and testing image was converted to a Bag of Words representation with a dictionary size of 50. Bag of Words representations were classified by K-Nearest Neighbors with k=24. To determine the stride between and size of keypoints to compute dense SIFT features for each image, multiple configurations were tested. Figure 4 shows the accuracy of the K-Nearest Neighbors Classification. The x axis is the stride and size (stride and size were set equal to each other for simplicity's sake), and the y axis shows corresponding accuracy. We see the optimum of the stride=size=15 to stride=size=65 to be stride=size=15. To optimize time-efficiency in debugging, the dictionary, dense SIFT feature representations of training images,

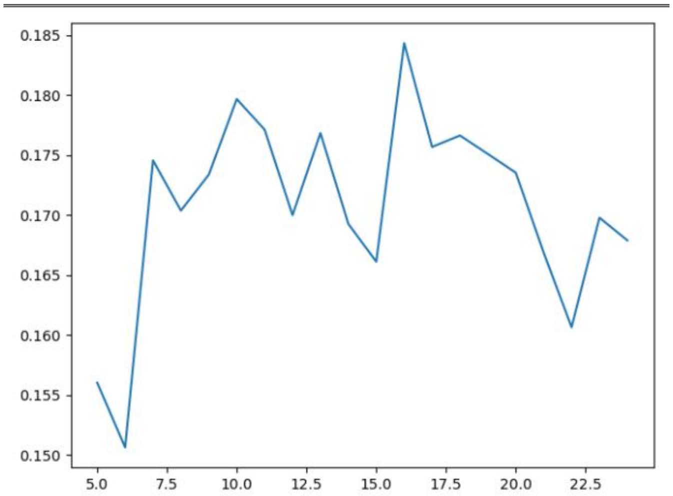

Figure 1. Accuracy of KNN classification of tiny images against different tiny image sizes.

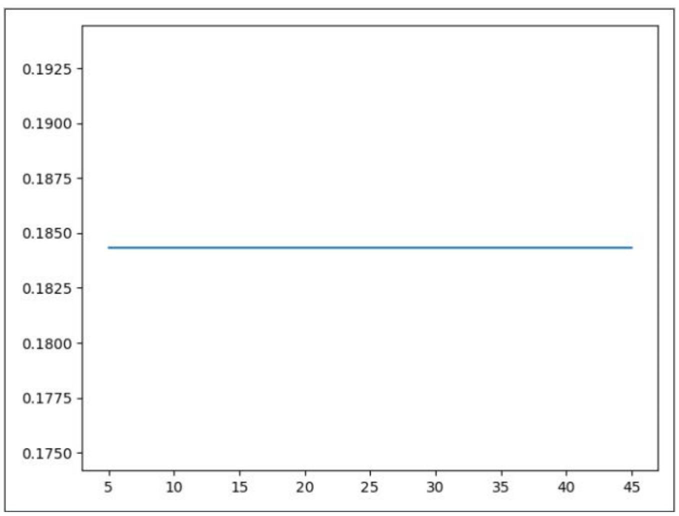

Figure 2. Accuracy of KNN classification of tiny images against different values for k.

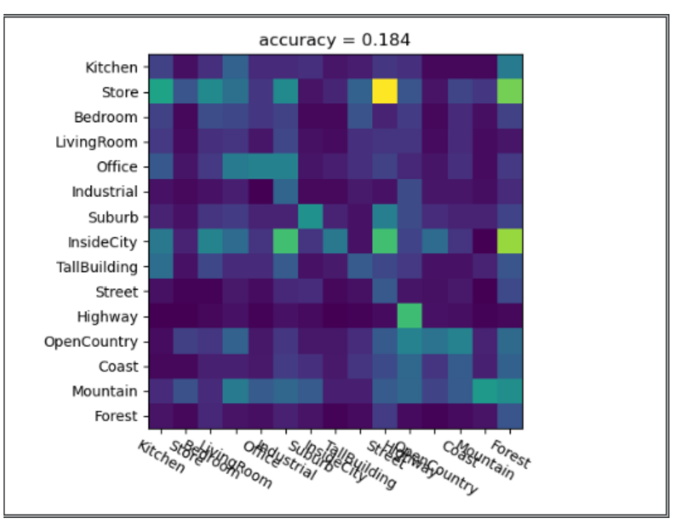

Figure 3. Confusion matrix of KNN classification of tiny images with k=24 and size=16x16.

and the Bag of Words representations of training and testing images were cached, resulting in significantly less processing time. Figure 5 shows the confusion matrix of K-Nearest Neighbor classification of Bag of Words representations of images, achieving an accuracy of 59.6%.

Using the same Bag of Words representations above, testing images were classified using a Support Vector Classifier. Support Vector Classifiers are inherently binary, so 15 classifiers (1 for each class) were trained with the labels '[class]' or 'not [class]'. The training set was classified by each classifier and the probability of each image being labeled as a given class was computed. For each training image, there existed 15 probabilities. The highest probability among those 15 was assumed to be that image's true label and the image was classified as such. Support Vector Classifiers have a free parameter, lambda. To determine the optimal lambda, multiple were tested, shown in Figure 6. We see an optima at lambda=2, and that was selected to compute the confusion matrix shown in Figure 7, achieving an accuracy of 66.3%. Similar to K-Nearest Neighbor classification, stride=size=15 and the dense SIFT features of training images, the dictionary, and the Bag of Words representations of training and testing images were cached for time efficiency.

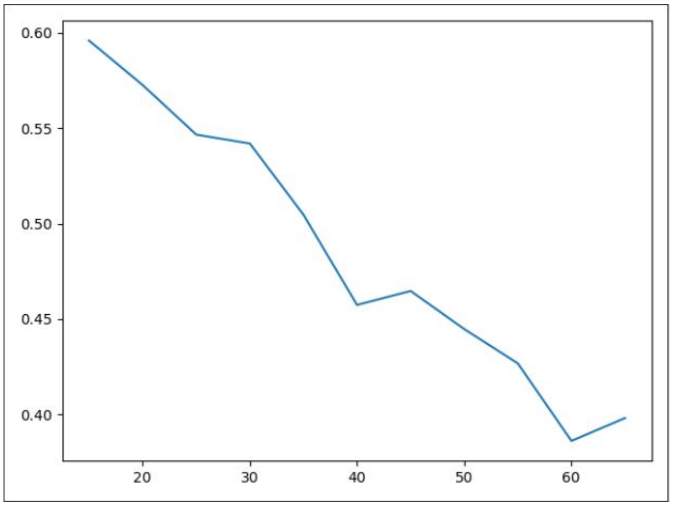

Figure 4. Accuracy of KNN classification of BoW features of images against stride between keypoints and size of keypoints (stride and size equal)

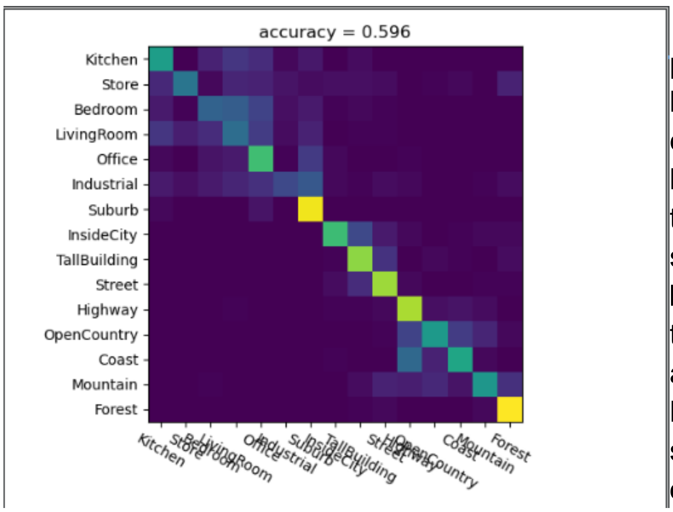

Figure 5. Confusion matrix of KNN classification of BoW features (k=24, stride=15, size=15)

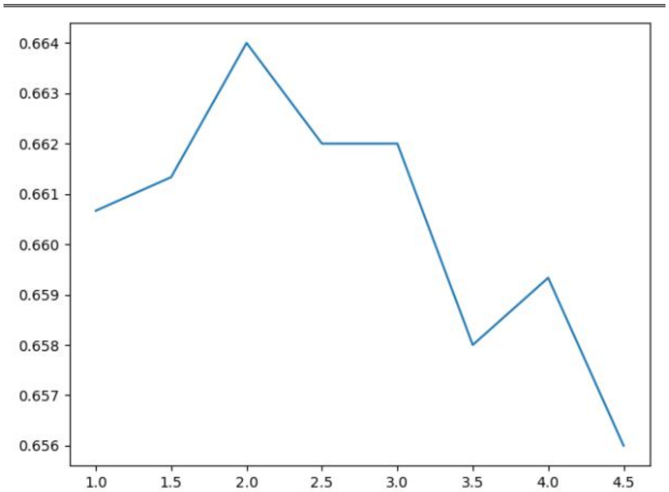

Figure 6. Accuracy of SVM classification of against lambda

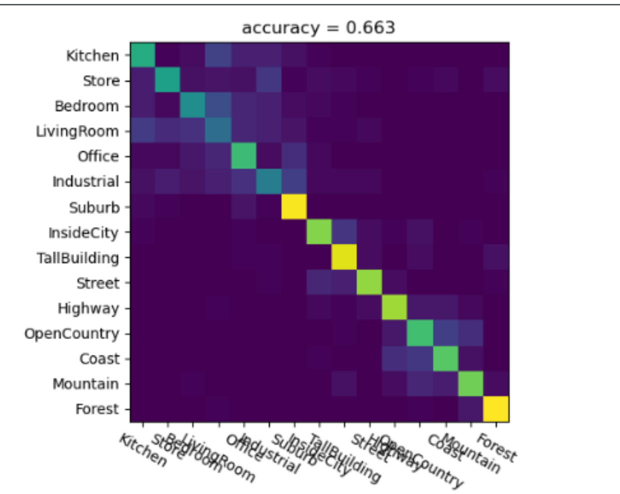

Figure 7. Confusion matrix of SVM classification of BoW features (lambda=2, size=15, stride=15)