CSCI 5561: Computer Vision
Prof. Hyun Soo Park

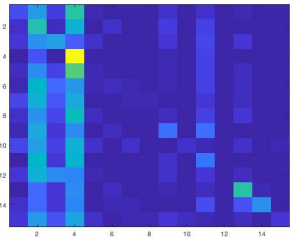
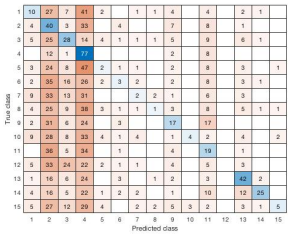
HW3 Report | Name: Abinash Sinha | x500: sinha160

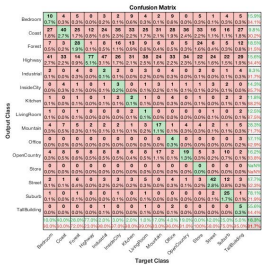
Note: In knn we choose value of k keeping in mind that if k is too large then neighborhood may include points from other classes and if k is too small then it comes sensitive to noise points.

For Tiny Image representation

There are 15 classes. If we randomly assign to an image then there is around 6.67% of accuracy possible. If we re-size the image to a tinier image and vectorizing it following column major format and running k-nn (k = 12) algorithm to find the class of a test image. Then this naïve vector representation seems to produce 18.33% which is better than random classification accuracy of 6.67%.

I have taken k = 12 for running k-NN algorithm; accuracy comes out to be 18.33% and the confusion matrix is as shown below:



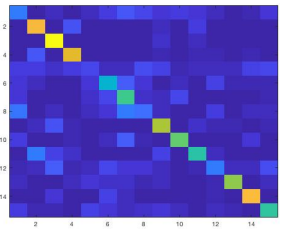
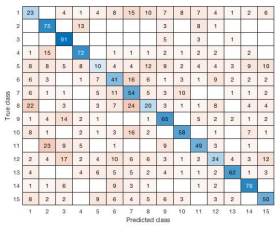
k	Accuracy (in %)
18	18.2
15	18.07
12	18.33
10	17.93
8	18.27
5	18.2
3	18.27
2	17.07
1	20

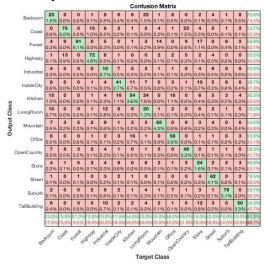
For Bag of words representation:

(idea of bag of words is used because even small patches contain class specific distinctive features and SIFT descriptor is used because it is not sensitive to illumination unlike HoG descriptor and also concatenation of patches' vector representations is not done because then it will introduce spatial relationship which is clearly avoided in Bag of Words concept using frequency of word which in our case is the mean of each of clusters of similar images)

I have chosen 'size' and 'step' parameters' values as 20 and 10 respectively while calling **vl_dsift()** function for getting dense SIFT features for each image. Also, the number of max iterations in k-means algorithms to find the clusters is 400 in order to facilitate converging of the algorithm execution because when running with default value of 100 it wasn't converging.

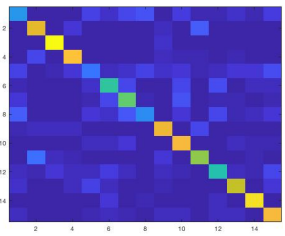
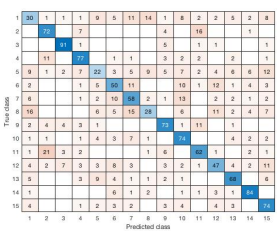
1. When using KNN algorithm to classify: (I have chosen value of k = 10)

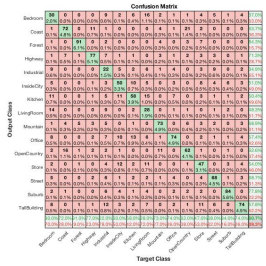





k	Accuracy (in %)
18	51.53
15	51.6
12	51.33
10	51.33
8	50.4
5	49.33
3	45.33
2	41.2
1	45.07

1. When using SVM algorithm to classify: (I have chosen value of lambda = 10^{-5} and loss function to be HINGE2 in order to penalize the incorrectly classified instances more than usual HINGE loss function and maximum number of iterations to be (100 / lambda) for SVM execution in order to converge)



lambda	Accuracy (in %)
10^{-6}	60
10^{-5}	60.67
10^{-4}	60.4
10^{-3}	59.6
10^{-2}	56.4
10^{-1}	44.73
1	11.8
10	7.53

$$\min_{\mathbf{w}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \max\{0, 1 - y_i f(\mathbf{x}_i)\} + \lambda \|\mathbf{w}\|^2 \right\}$$ L2 regularized hinge loss for SVM algorithm