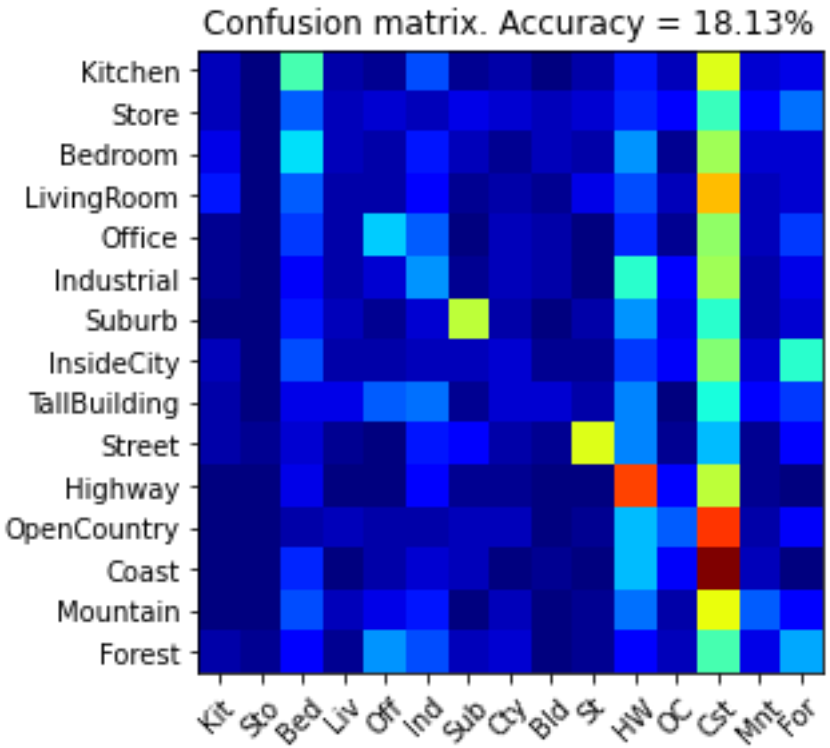
CS 4476
PS 4

Younes Djemmal
ydjemmal3@gatech.edu
903794219

Part 1: Tiny Image Representation and Nearest-Neighbor Classification

Part 1.3.a: Your confusion matrix, together with the accuracy for Part 1 with the standard parameter set (image_size = 16, k = 3)

<Plot here>



Part 1.3.b: Experiments: change image size and k individually using the following values, and report the accuracy (when tuning one parameter, keep the other as the standard (16 x 16, 3)):

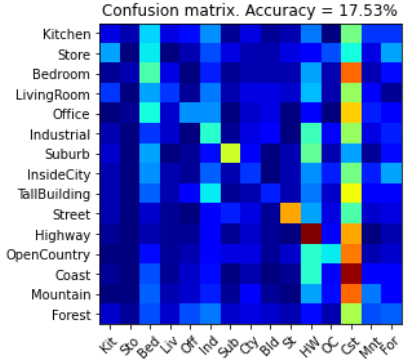
ie. when you're tuning image size, keep k at 3, when changing k, keep image size as 16x16

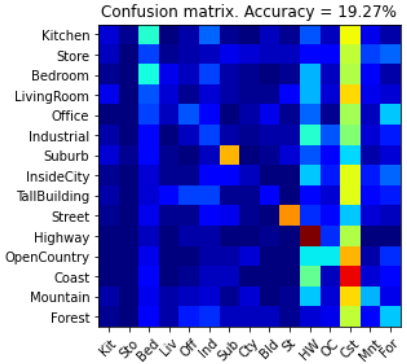
image size:

8 x 8: 17,53

16 x 16: 18,13

32 x 32: 19,27





k:

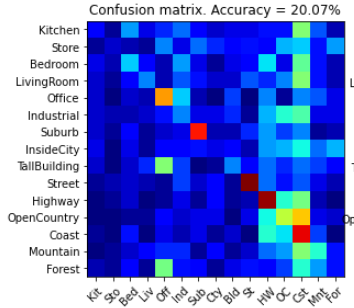
1: 20,07

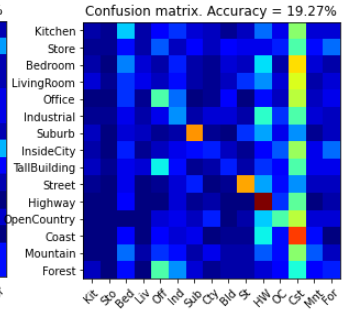
3: 18,13

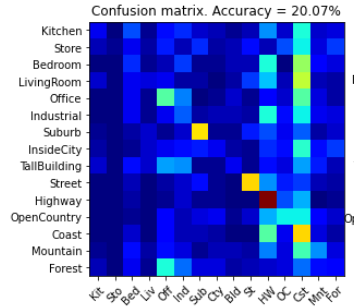
5: 19,27

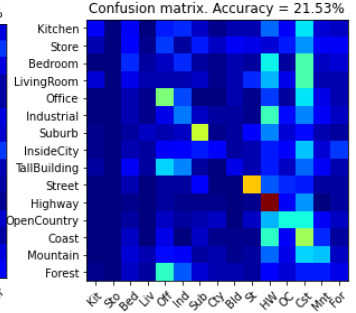
10: 20,07

15: 21,53









Part 1.3.c: When tuning the parameters (image size and k), what did you observe about the *processing time and accuracy*? What do you think led to this observation?

Processing time and accuracy both increase with the increase of image size and k. This is due to the increase in the size of our stacked feature vectors with the increase of the size of our images. Calculations will take longer with bigger feats arrays, especially the distance calculation.

Part 2: Bag-of-words with SIFT Features

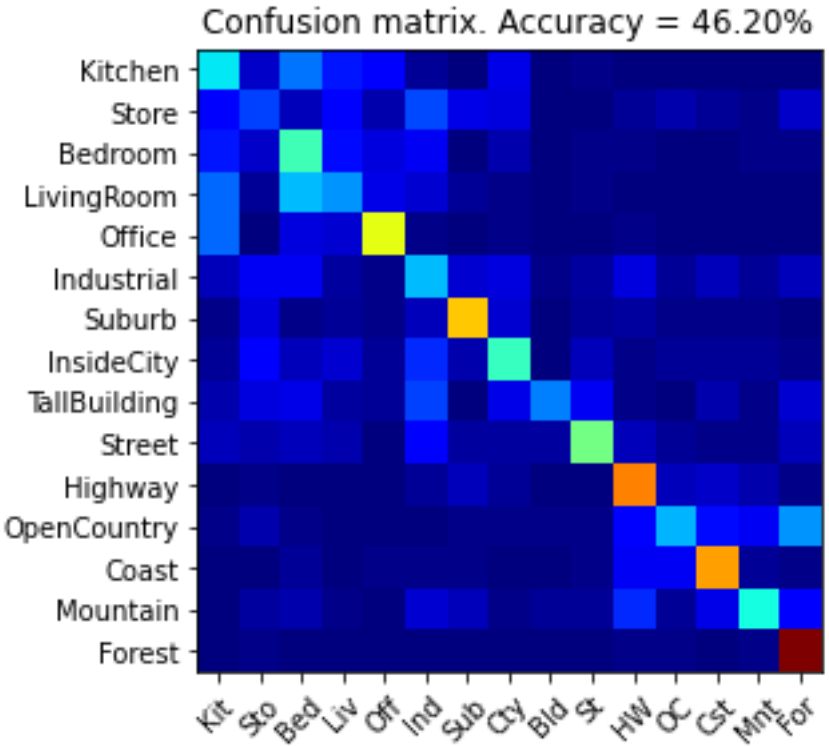
Part 2.3: Reflection on Tiny Image Representation vs. Bag of Words with SIFT features:

Why do you think that the tiny image representation gives a much worse accuracy than bag of words? Additionally why do you think Bag of Words is better in this case?

1. Because it discards all of the high frequency image content and is not invariant to spatial or brightness shifts.

2. Bag of words builds a vocabulary by extracting features from all images in the dataset by sampling sift invariant descriptors from all regions of the image with a given stride. This enables us to encapsulate the high frequency content of the image.

Part 2.4.a: Your confusion matrix, together with the accuracy for Part 2 with the standard parameter set (vocab_size = 50, k = 3, max_iter = 10, stride(build_vocab) = 20, stride(get_bags_of_sift) = 5



Part 2.4.a: Experiments: change vocab_size and k individually using the following values, and report the accuracy (when tuning one parameter, keep the other as the standard (50, 3)):

ie. when you're tuning vocab_size, keep k at 3, when changing k, keep vocab_size as 50. (Other params max_iter = 10, stride(build_vocab) = 20, stride(get_bags_of_sift) = 5)

vocab size:

50: 46,20

100:49,93

200:51,80

k:

1: 44,87

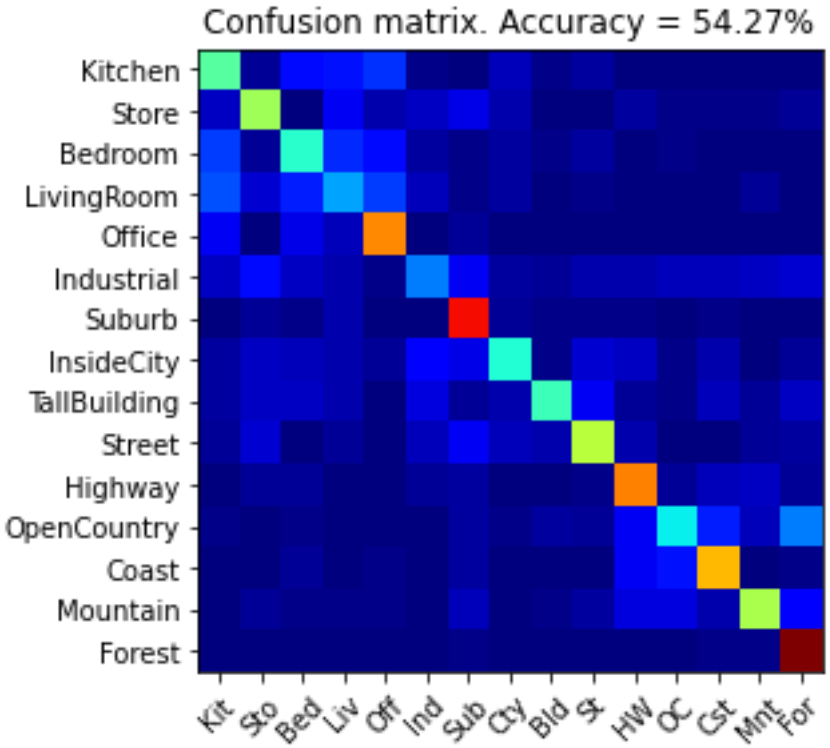
3: 46,20

5: 47

10: 49,60

15: 49,8

Part 2.4.a: Paste the confusion matrix for your best result with the previous experimentation in this slide.



vocab_size: 200

k: 10

max_iter: 10

stride(build_vocab): 20

stride(get_bags_of_sift): 5

Part 2.4.b: Reflection: when experimenting with the value k in kNN, what did you observe? Compare the performance difference with the k value experiment in Part 1.3, what can you tell from this?

When experimenting with K, I observed that the accuracy goes up until it hits an optimal value then it starts going down again. This means that there is an optimal value of K for each different set of parameters. This value of K can be found by running the algorithm with different values of k and picking the value that gives the least error.

Part 3: Extra Credit

EXTRA CREDIT

Part 3.1: Post best confusion matrix, together with the accuracy out of all the parameters you tested. Report the parameter settings used to obtain this result.

<Plot here>

Parameter settings:

max_iter:

stride(build_vocab):

stride(get_bags_of_sift):

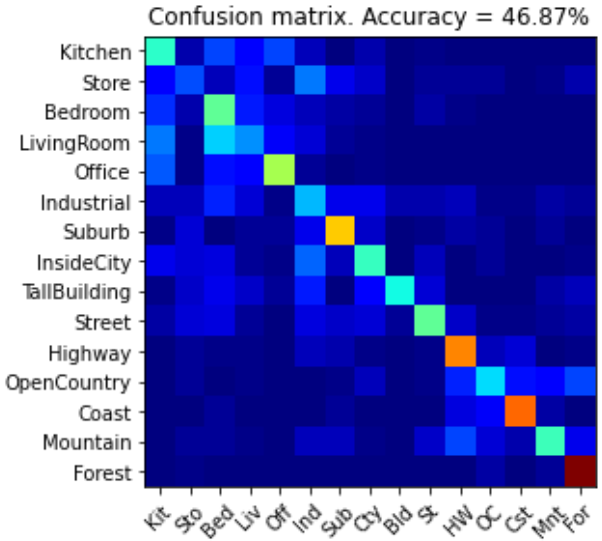
vocab_size:

k (kNN):

EXTRA CREDIT

Part 3.2: Post confusion matrix along with the distance metric that you used for achieving a better accuracy on standard parameters. Why do you think it performs better?

<Plot here>



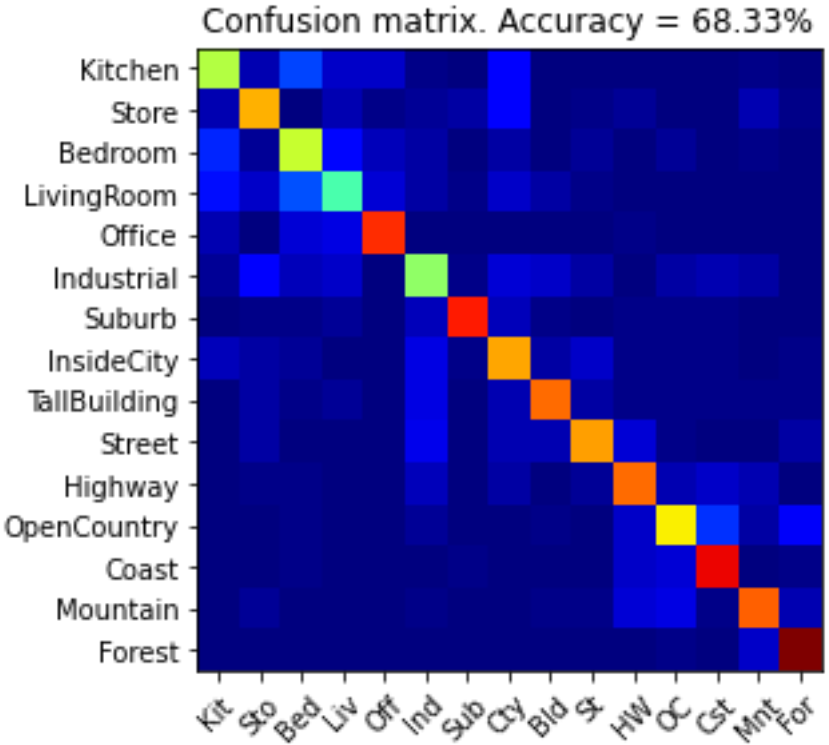
Distance metric and why it works better:

Braycurtis

This metrics quatify how different two instances in terms of the instances found in the whole array. This works well in our bags of words implementation because it is the concept of bags of words. We are trying to find the most similar correspondence according to our vocabulary.

EXTRA CREDIT

Part 3.3: Post confusion matrix along with your explanation of your SVM model and detail any other changes your made to reach an accuracy of 65% or greater.



Description of your model:

svc = SVC(C=0.009, kernel='linear', probability=True)