

CS 4476/6476

PS 5

Sreyans Sipani
ssipani6@gatech.edu
903310164

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)

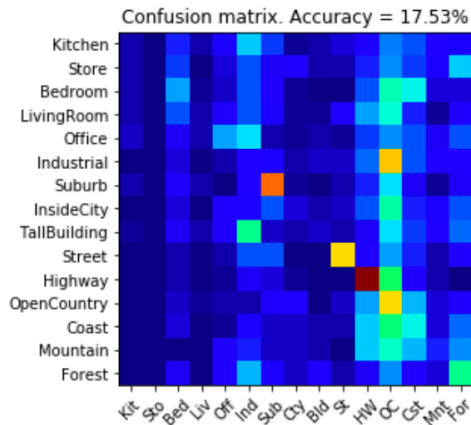
```
In [9]: size = 16
        k = 3

        train_image_feats = get_tiny_images(train_image_arrays, size)
        test_image_feats = get_tiny_images(test_image_arrays, size)

        print('Using NEAREST NEIGHBOR classifier to predict test set categories')
        predicted_labels = sc.nearest_neighbor_classify(train_image_feats, train_labels, test_image_feats, k)
```

Using NEAREST NEIGHBOR classifier to predict test set categories

```
In [10]: show_results(test_labels, categories, abbr_categories, predicted_labels, 'cm.png')
```



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: 18.67%

16 x 16: 19.00%

32 x 32: 19.47%

k:

1: 18.93%

3: 19.00%

5: 21.33%

10: 21.00%

15: 22.00%

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?

The processing time increases with increasing k as we need to vote among more neighbours but the accuracy slightly increases as well. The accuracy seems to slightly increase with image size as more information can be stored.

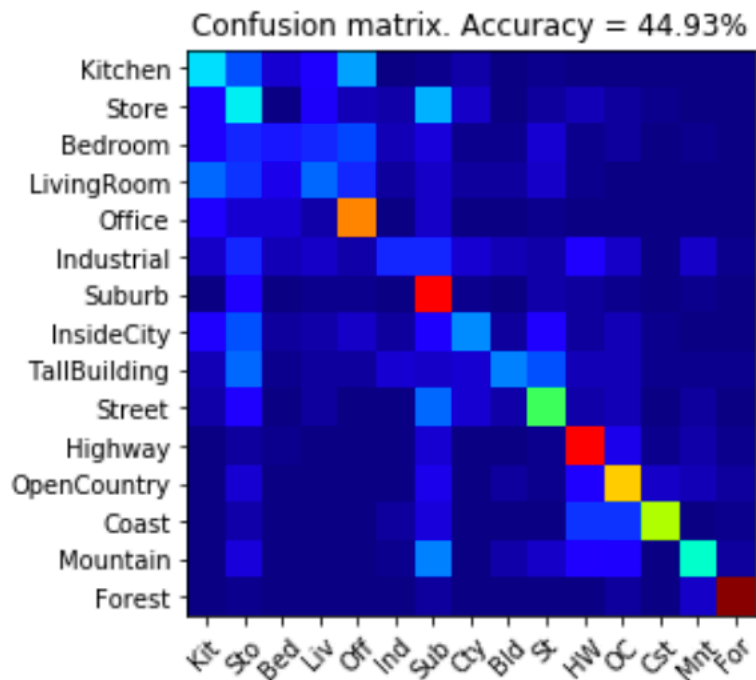
Part 2: Bag-of-words with SIFT Features

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?

Tiny image loses a lot of information while Bag of Words keeps the important features of the image. Therefore the clustering is better with the Bag of Words representation.

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: 44.93%

100: 43.87%

200: 44.93%

k:

1: 40.3%

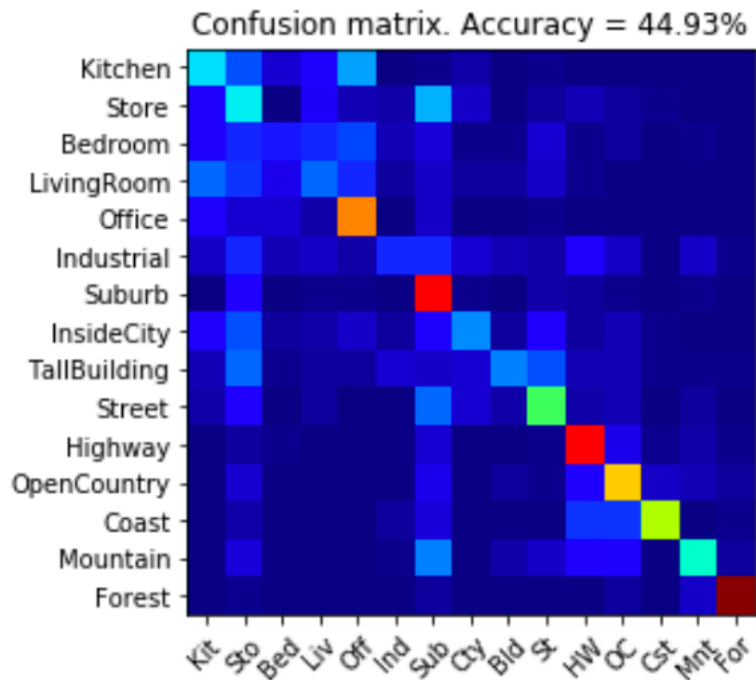
3: 44.93%

5: 41.87%

10: 42.93%

15: 42.01%

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



vocab_size: 50

k: 3

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?

The performance is much better for Bag of Words but increasing k only slightly increases the accuracy and does not affect the overall performance a lot. This is fairly similar to 1.3 where we saw similar accuracies for different values of k .

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:

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.

<Plot here>

Description of your model: