Cmpe-537
Computer Vision
Homework 3
Bag of features Image Classification

Giray Eryilmaz

December 2, 2019

Contents

1 Problem

Image classification using bag of features. It has 2 main parts: Representation learning and classification.

2 Method

2.1 Representation learning

This part has 3 subparts: Computing sift descriptors (feature extraction), finding codewords and building histograms as representations of images.

2.1.1 Feature Extraction / Computing SIFT descriptors

I have used cv2 shift detect method to get interest points and their 128 length descriptors. The method yields various number of these 128 length vectors. Sometime in the scale of 30 - 50 and sometimes thousands of them 3k -5k. When I used all the training images total number of these point descriptors reached more than 2.5 millions. That is why I had to limit the number of vectors extracted from each image by using nfeatures parameter of shift. If provided, the method only returns most important nfeatures many vectors per image. This reduced accuracy by approximately 9 percent but otherwise I was not able to make the experiments.

2.1.2 Finding codewords / Finding the dictionary

Codewords are chosen features that will be used to represent images as histograms. Other non-chosen vectors are counted as their closest codeword. I have used k-means clustering to find the codewords (centroids are the codewords) and assigning the vectors to the features. I have used kmeans from scikit-learn library.

2.1.3 Building Histogram / Feature quantization

Each vector of size 128 was assigned to a codeword. From this point on the 128-sized descriptors themselves are not needed. Every image now has some labels corresponding to the labels of the descriptor vectors it had. By this way each image is now represented by a histogram ,or a vector, of length k. These k values correspond to the frequencies of the codeword labels and some of them may be zero. Also since the number of descriptors extracted were different from image to image, these histograms should be normalized. L1 normalization of 1 was said to be used commonly with chi-square kernel in the scikit learn docs so I have used it instead of l2 norm.

2.2 Classification

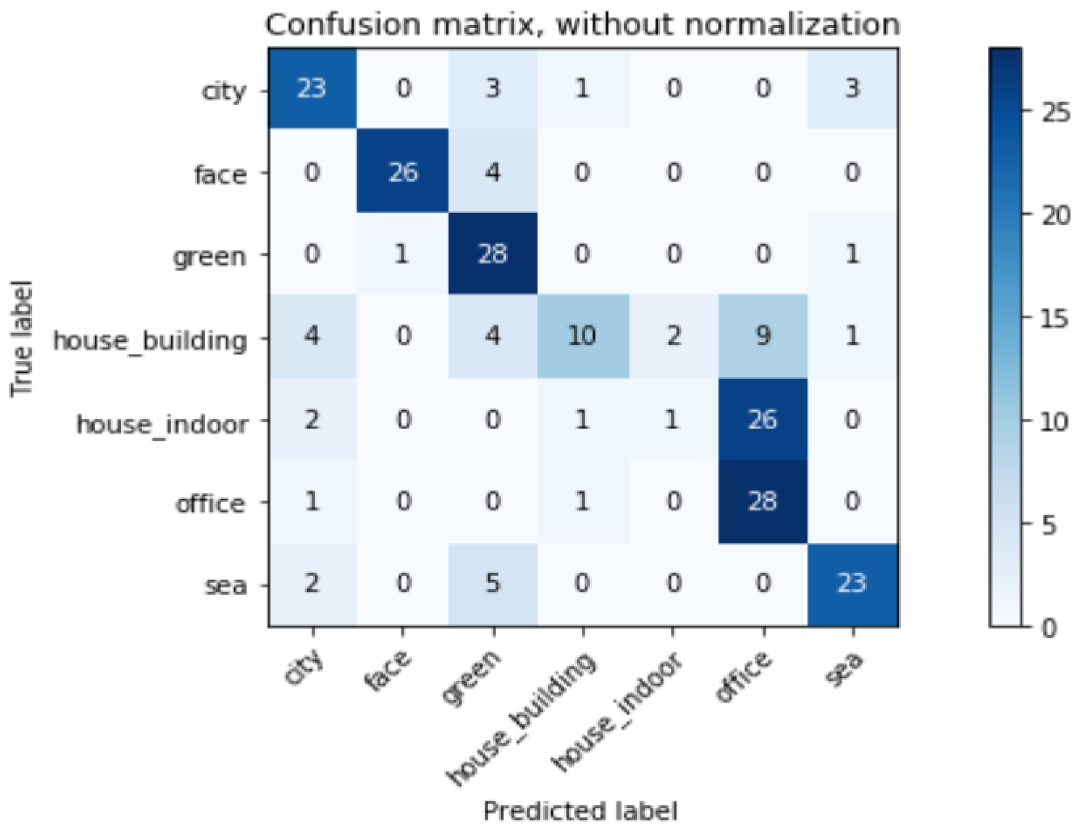
I have used SVM from scikit learn as instructed. I have also used chi2 kernel from sklearn metrics pairwise as a custom kernel as requested.

I have used GridSearchCV for parameter search as well. You can see the results in the next section. I have considered C value between 1 - 15. I have tested 3 kernels : linear, rbf and chi2 and chi2 was the best every time. I have also tested decision function shape ovo and ovr. Validation prefers ovo over ovr (with C=7) but my additional test results in the test set showed that it was not very meaningful.

3 Results

K	Best C	Best Kernel	Test Accuracy	Train Error
50	3	chi2	0.647	0.05
100	5	chi2	0.6428	0.03
500	4	chi2	0.66	0.002

The results clearly indicate that chi2 is good for this task as expected. Also C value should be between 3 - 5 or similar.



(a) Confusion matrix

The confusion matrix shows that some categories are easily distinguished, city, face, green, see... The biggest confusion comes from house indoor which is incorrectly labeled as office which I believe is not very surprising because of the nature of the images.

4 Notes and Remarks

My computer was not powerful enough to run the algorithms due to the number of features ie number of extracted descriptors. That is why I had to use Colab which had its own challenges. Even there for k=500 case it took literally hours so I had to limit the number of features to 500. I have also tried smaller numbers (100, 150 etc) but they reduced the accuracy very much. I could only once run the system with all the descriptors before google disconnected me due to over use I believe. I could see that the accuracy can increase up to $\sim$70 percent if I could use all the descriptors.

I have included the best results for k=50, 100 and 500 as requested.

I could not find average training error metric however I have found training score which I believe to be 1-training error. Reported as such.

4.1 How to run the code

I ran the code at colab so I submit the notebook, the version I submit downloads the dataset then unzips it so that you don't have to upload the dataset to your gdrive and connect to your gdrive from colab. Also I saved the code as a python script since it was asked.
One important thing is both the notebook and the script assumes that they are in the same folder next to train and test folders.

Unfortunately the results are not 100 percent reproducible, some fluctuations do happen but the variance is not much. This is I believe something related to the inner workings of scikit learn library.