Cognitive Computing, 2018 Fall
HW2 Report
R07922087 李鈺昇

1 Introduction

In this assignment, we are asked to apply various kinds of techniques to extract useful features for image retrieval. They can be roughly classified into three types:

Type	Methods I try in this homework
Color-based	global/regional color histogram
Texture-based	local binary patterns
Local features	SIFT, SURF, ORB

In the following, I will discuss what I've done, what problems I have encountered, and compare the retrieval performances.

2 Color-based features

It's fairly straight-forward to calculate the global color histograms of the gray-scale images. The function calcHist provided by OpenCV is suitable for this. In this way, for each image we obtain 256 feature values, corresponding to the bin counts of different pixel values in [0, 255].
As expected, the result is quite poor. So a simple improvement can be made by using the 3 RGB channels to get features of length 256*3=768. As can be observed in the table, there is only little improvement.
Next, a further step is to divide the image into four blocks in a 2x2 way. Four histograms for each block are calculated independently, so the resulting length is 768*4=3072. This helps improve the performances, especially for the museum_paintings category.

2.1 Comparing histograms

An issue is how histograms are compared. This problem lies not only in color-based features, but also in other features whenever they are 1-dimensional real vectors. I have tried four commonly used distance metrics:

1. Correlation coefficient
2. Chi-square
3. Histogram intersection
4. Bhattacharyya distance

According to the results, it seems that chi-square and histogram intersection are the two better ones, though the margins are not significant.

3 Texture-based features

For texture features, I tried local binary patterns (LBP), where there is off-the-shelf functions, local_binary_pattern and hog, in scikit-image.

3.1 LBP

I choose radius=2 and 8 points for each circle for LBP, and the four distance metrics are again used. The results are quite bad.

4 Local features

This is the most interesting part. Although SIFT is well-known as the golden method, I still found several comparisons on SIFT, SURF, ORB, etc. Thus I tested the three methods and they all gave wonderful results. All of them can be found in OpenCV.

4.1 SIFT

The number of keypoint descriptors of an image can be up to thousands, so in addition to an original version of simply calling the function SIFT_create, I also tried another version SIFT_create(nfeatures=500), which limits the number of descriptors to be at most 500. The feature extraction time doesn't differ much, but the matching time drops a lot. And the performance only drops a little.

4.2 SURF

For a fair comparison, I run SURF_create(hessianThreshold=2000) so that the number of descriptors is similar to that of SIFT. The extraction time is indeed much faster than SIFT, while the matching time lies between my two versions of SIFT. The performance is similar to SIFT.

4.3 ORB

Both the extraction time and the matching time are much shorter than SIFT and SURF, but the performance is worse. It may be due to the shorter descriptor length.