Introduction to Computer Vision

Assignment #4

Due: Dec.-12 (Mon.) (before 11:59pm) No late submission allowed.

Instruction

a. Submit your source codes, report, and descriptor in a single compressed file "CV_A4_*StudentID*.zip" to iCampus.

b. Python 3.7 or higher / OpenCV 3.4 or higher will be used to execute your submitted codes.

c. You can submit at most 7 files.

- Mandatory files: *A4_StudentID.des, A4_compute_descriptors.py, A4_report.pdf*

d. Any work that you turn in should be your own.

Part #1. Image Retrieval [100 pts]

The requirements will be evaluated based on your descriptor file, '***A4_StudentID.des***'

1-1. Dataset: a subset of UKBench

a) Pre-extracted SIFT features of 1,000 images, *00000.sift – 00999.sift*, are given by a compressed file, *CV_A4_Feats.zip*. They should be placed in the subdirectory './feats/' from your *.py* files. Your source codes should refer them by a relative directory address.

b) For each image, there are 4 relevant items including itself. Those 4 images are ground truth retrieval results.

c) Each SIFT feature corresponding to an image (*00xxx.sift*) has the following binary file format:

$$\begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,128} \\ v_{2,1} & v_{2,2} & & v_{2,128} \\ & \vdots & \ddots & \vdots \\ v_{n,1} & v_{n,2} & \cdots & v_{n,128} \end{bmatrix}$$

The image has n SIFT features and each SIFT descriptor is 128-dimensional vector. Note that, each value $v_{i,j}$ is stored as 'unsigned char' format (1 byte representing a range from 0 to 255).

d) Note that, 100 sample features (*s00000.sift – s00099.sift*) and their corresponding images (*s00000.jpg – s00099.jpg*) are also provided by *CV_A4_Sample.zip*. You can utilize them to verify and debug your implementations.

1-2. Task: computing image descriptors for the retrieval task

a) Your task is to compute image descriptors that reflect similarities among images by L_1 or L_2 distances. You should store the descriptors as the following <u>binary format file</u>:
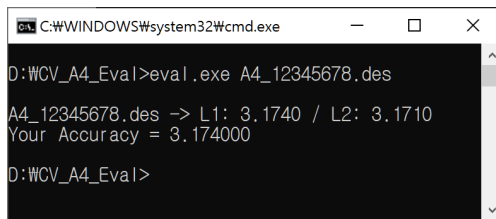
$$\begin{array}{l} N\ D \\ d_{1,1}\ d_{1,2}\ \dots d_{1,D} \\ d_{2,1}\ d_{2,2}\ \dots d_{2,D} \\ \dots \\ d_{N,1}\ d_{N,2}\ \dots d_{N,D} \end{array}$$

, where N and D are the number of images (=1000) and the dimensionality of each descriptor, respectively. And $d_{i,j}$ is j^{th} element of ith descriptor. The required data types of variables are summarized as follows:

Variable	Type
N	Signed 32 bits integer (**int** in C/C++)
D	Signed 32 bits integer (**int** in C/C++)
$d_{i,j}$	32 bits floating point (**float** in C/C++)

Note that, the <u>dimensionality of your descriptor cannot be greater than 1,024</u> (i.e. $D \leq 1024$). In other words, the size of your descriptor file '*A4_StudentID.des*' can be up to 4,096,008 bytes ($= 2 \times 4 + 1000 \times 1024$ bytes).

b) You need to submit a code to produce the descriptors. Your script should run when they are located at any directory containing the './feats/' sub-directory. In other words, you should specify the data path as a relative directory path in your implementation. Name your code as '*A4_compute_descriptors.py*'. If your algorithm utilizes codeword vectors based on the clustering (taking long computation time), you are asked to precompute/store and load them in your script. In this case, you should include these precomputed files in your submission package.

c) Once you produced the descriptor file, you can evaluate your descriptors by using provided executable 'eval.exe'. The program will evaluate your descriptors according to both L_1 and L_2 distances and take the higher one. It only works in Windows OS. You have to specify your descriptor file name as the following example:

 Note that, the accuracy is ranging from 0 to 4.

d) You are required to submit a report '*A4_report.pdf*' that describes your approach to compute the image descriptors. It should be longer than 1 page but shorter than 3 pages.

e) It is strongly prohibited to produce image descriptors by abusing the characteristics of the dataset. One typical example is setting $d_1 = d_2 = d_3 = d_4 = [1,0,0,...,0]$, $d_5 = d_6 = d_7 = d_8 = [0,1,0,...,0]$.