ASSIGNMENT 1
CS4186 VISION AND IMAGE

INSTANCE SEARCH

You are given a collection of 5,000 images, 10 example query instances and 20 testing query instances (you can download from this link: google driver link). Each image contains one instance (object). Your task is to implement two methods for instance search. Specifically, given a query instance, a method needs to find the images that contain the same query instance from the image collection and then ranks them according to similarity or confidence.

Marking scheme

A. Report (no more than 5 pages): Write a report that briefly describes the two methods and compares their search performance for the 10 example query instances. The performance should be measured by mean average precision (MAP). The MAP program is given to you. You should also show the results for some queries by listing the retrieved images. (20%)

B. Algorithm: Implement any two methods that you learn in class (e.g., color histogram, LBP, filter banks, SIFT) for instance search. (30%)

C. Benchmarking: For each of your methods, show the retrieval result for each of the 20 testing query instances. The mark will be allocated based on the retrieval performance of the method with better performance. (50%)

DEADLINE

The submission should be done before 15-Mar, 11:59pm.

Penalty on late submission: 20% of marks will be deducted per day. No submission will be allowed after five days from the deadline. *Do NOT copy code from the internet and do not borrow other people's code. Remember that PLAGIARISM is a serious offense for which you may fail the class or even be expelled from the university.*

CONTACT PERSON

Please email Teaching Assistant Alan Nguyen (Email: tnguyenhu2-c@my.cityu.edu.hk) for technical questions.

NOTE

1. In the provided zip file, the folders are organized as: **images** – 5,000 images for search; **examples** – 10 example query instances; **Queries** – 20 testing query instances.
2. The instance of an image is given as bounding box. The bounding box is represented as (top left corner in x, top left corner in y, width, height) in the text file. For example, the file *01.txt* specifies the bounding box of the query instance for image *01.jpg*.
3. The location of instance, given as a bounding box, is also provided for 2,000 images.
4. In the **examples** folder, you can find the program *metric_map.py* for measuring search performance in terms of MAP. The value of MAP will be high (at most 1.0) if the images containing the query instances are ranked at top.
5. In the **examples** folder, the ground-truth (or answers) for the 10 examples queries can be found in the file *rank_groundtruth.txt*.
6. You can use any distance or similarity function for measuring the similarity between two images or instances. Examples include:

   Euclidean distance between a query $Q = [q_1, q_2 \cdots q_n]$ and an image $F = [f_1, f_2 \cdots f_n]$:

$$Distance(Q, F) = \sqrt{\sum_{i=1}^{n} (q_i - f_i)^2}$$

   Cosine similarity between a query $Q = [q_1, q_2 \cdots q_n]$ and an image $F = [f_1, f_2 \cdots f_n]$:

$$Similarity(Q, F) = \frac{\sum_{i=1}^{n} q_i f_i}{\sqrt{\sum_{i=1}^{n} q_i^2} \sqrt{\sum_{i=1}^{n} f_i^2}}$$

SUBMISSION

Note: 10 marks will be deducted if (1) rank list is not in the required format, (2) program is difficult to understand.

Please zip the followings and submit to Canvas:

- ✓ *Computer program*: Python3 recommended.
- ✓ *Report*
- ✓ *Rank list*: A text file (rankList.txt) showing the images in descending order of similarity:

```
Q1: 7 12 214 350 …
Q2: 301 501 1990 2 …
Q3: 288 345 389 1290 …
Q4: 248 293 1098 2000 …
Q5: 380 287 392 478 222 …
```

Remark: 20 rows only; Each row should list the names of the 5,000 images (named in number) in descending order of their similarities to a query.