CSS 484 Multimedia Data Processing
Winter 2022 (12%)

Project 1: Implement a Simple Content-Based Image Retrieval System

Preliminary demo: January 13th (Thursday) in class
Program & Report Due By: January 19th (Wednesday) 11:59pm
Present/Demo: in class of January 20th, 3~5 minutes each in groups

Description:

This project is to implment a simple Content-Based Image Retrieval system based on two different color histogram comparison methods. It can be done individually or in pair (i.e., with one another student). If in pair, self-signup on Canvas

1. Test Image Database
   This test image database includes 100 true-color images in .jpg format.
2. Color Histogram
   Color histogram comparison is a simple but effective apporach in CBIR systems. Here are two ways to combine the information from 3 color channels (R, G, B):

   A. Intensity Method

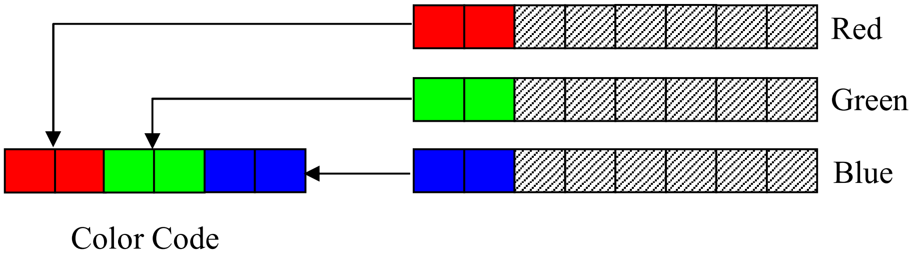
   $$I = 0.299R + 0.587G + 0.114B \qquad (1)$$

   By this way, the 24-bit of RGB (8 bits for each color channel) color intensities can be transformed into a single 8-bit value. The histogram bins selected for this case are listed below:

   H_1: $I \in [0,10)$; H_2: $I \in [10,20)$; H_3: $I \in [20,30)$;
   H_4: $I \in [30,40)$ …… …… …… H_6: $I \in [50,60)$;
   … … …
   H_{25}: $I \in [240,255]$;

   B. Color-Code Method

   The 24-bit of RGB color intensities can be transformed into a 6-bit color code, composed from the most significant 2 bits of each of the three color components, as illustrated in the following figure.



The 6-bit color code will provide 64 histogram bins.

For example, the R, G, and B values for a pixel are 128, 0, and 255 respectively. So the bit representations of them are 10000000, 00000000, and 11111111. Then the 6-bit color code value will be 100011.
In color code, there will be 64 bins with H1: 000000, H2: 000001, H3: 000010, … H64: 111111

3. Histogram Comparison

   You need to implement the distance metrics for histogram comparison. Let $H_i(j)$ denote the number of pixels in jth bin for the i^{th} image. Then the difference between the i^{th} image and k^{th} image can be given by the following distance metric:

   - **Manhattan Distance**

$$D_{i,k} = \sum_{j=1}^{G} \left| \frac{H_i(j)}{M_i * N_i} - \frac{H_k(j)}{M_k * N_k} \right| \tag{2}$$

where M_i*N_i is the number of pixels in image i, and M_k*N_k is the number of pixels in image k.

Preliminary demo (in groups):

Show the graphic user interface is up running. It should allow users to browse the image database, select the query image, and to view the retrieved images. At this stage, it is not required (though it's highly recommended) that the retrieval algorithm is properly implemented so the retrieved images can be any images from the database.

Present/demo, PPT slides preferred but not required (in groups, peer grading):

Every student need to

1. Discuss libraries/tools/techniques you found useful, be specific (e.g., XX library is useful in image processing because …).
2. Show user query interface

   The graphic user interface should allow users to browse all the images in the image database, select the query image, and to view the retrieved images. Given a query image, the retrieved images should be displayed to user according to their similarity ranks to the query image. In particular, the similarity rank decreases from left to right, and top to bottom. Moreover, your user interface should be able to support the "next page" operation, with each page containing at least 20 retrieved images.
3. Demo implementation of two color histogram comparison methods

   You need to demo the query results using the two differenct color histograms as described in Description Section. Users should be allowed to switch between two different query methods within the same application.

Submission Requirements:

The **softcopy of well-commented code (& an executable ready for testing) and report** needs to be submitted as **a zipped file** to Canvas by the due time. The **report** should be **one single document** including: the discussion of how to run your program, how to use your system with step-by-step instructions, whether you use the sample code I provided or develop all by yourself, screen dumps to show the first page of retrieval results for **query images 33.jpg and 93.jpg**.

You will also be asked to do a preliminary demo as well as a presentation to demonstrate your programs (see **Preliminary demo and Present/demo** sections above).

Evaluation Criteria: - 100 points

Correctness (testing):	45 pts
User friendly:	20 pts
Report and comments:	25 pts
Presentation (peer evaluations):	10 pts