Content-Based Image Retrieval HW

In this homework, you will learn how to implement a very simple content based image search system and evaluate its performance.

Dataset: U. Kentucky Benchmark Image Database
Reference: D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 2161-2168, June 2006.

The original database includes 2,550 objects with 4 images for each object (total of 10,200 images). The images were taken to simulate variations of lighting, view, occlusion, scale, etc. The images are numbered sequentially, i.e., images of every 4 consecutive numbers belong to the same object (e.g., 00000-00003, or 00100-00103). Note some objects may belong to the same category (e.g., books, CD covers, etc), but only images of the same object will be considered correct matches in this homework. To make the computation more manageable a subset of 2,000 has been extracted. The subset can be downloaded at the URL: http://www.ee.columbia.edu/~rj2349/index_files/Homework1/)

Problems:
1. Write a program to extract the color histogram of each of the 2,000 images. Choose the parameters required with justifications. Implement your own histogram code and compare its results with open-source API like OpenCV and numpy.
Required submission: source code with comments

2. Write a program to measure the L_2 distance between color histograms of two images.
Required submission: source code with comments.

3. Use 5 images shown above (ukbench00004.jpg; ukbench00040.jpg; ukbench00060.jpg; ukbench00588.jpg; ukbench01562.jpg) as queries. For each query image, find 10 best matches from the 2,000 images based on the color histogram similarity.
Plot the query image and the 10 returned matches (use icons of reduced resolution to save space).
Required submission: source code with comments, plots of matched images for each query.

4. Write a program to measure and plot the P-R curve for each query.
Required submission: source code with comments, plots of P-R curves.

5. Discuss and explain success and failure cases.
Required submission: written report (at least 1 paragraph)

6. (Optional) try to improve the results by using a different feature or distance metrics (bag of words, 3d-color histograms etc) of your choice. Justify your choice with qualitative reasons and back it up with performance comparison results.
Required submission: source code with comments, plots or results and P-R curves, written report .

All tasks should be done in Python with usage of OpenCV as a tool for reading and manipulating the images.

References:

https://acutecaretesting.org/en/articles/precision-recall-curves-what-are-they-and-how-are-they-used