Project 2: Content-based Image Retrieval

Student: Yunyi Chi

Description:

In this project, we achieve Content-based Image Retrieval in a given database. Depending on the image or object we want to retrieve, we adjust the region of the image we want to analyze (center, bottom, top _halve, bottom_halve), the histogram (RGB color histogram, rg chromaticity histogram, 1d greyscale histogram), and the distance matrics.

For the custom design part, we design a customed CBIR for green dumpsters.
For the extension part, we recall pictures which contain blue trash can bins.

Introduction to our file system:

main.cpp:
The main file in this project. it includes functions for different tasks (task1(), task2(), …extension()) and a main function which takes argument and triggers tasks.

processing.cpp(with header file)
Includes many functions related to image processing, such as choosing region of image, calculating histogram, flattening histograms and calculate distance by histograms.

filter.cpp(with header file)
Includes filters such as SobelX, SobelY and get magnitude images.

Overall Method:

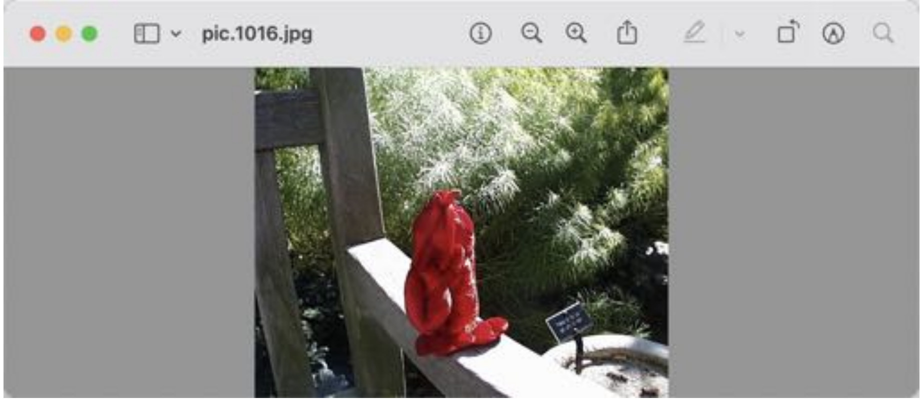
For this project, we choose the first method, which is to make a single program that does everything. Here are our general steps:

1. Get image database, target filename, desired number of output images, and task we want to trigger from command line.
2. Read all image files in database, save the image filename with path in an unordered map, pass information and this image map to our desired task.
3. On task, first read, process and calculate our desired histogram for the target image, then loop through the image map, doing the same for the other images. Calculate the distance between the target image and other images and save the distance in another unordered map in the form of <filename, distance>.
4. Sort the distance map to get the best matches.

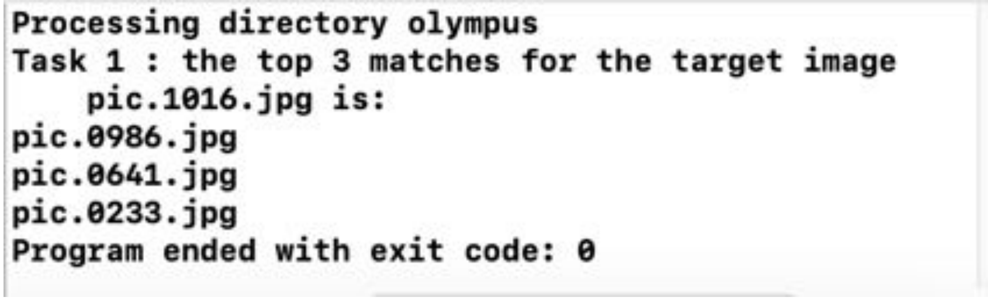
Task 1: Baseline Matching

Feature vector: Use the 9x9 square in the middle of the image as a feature vector
Distance metric: Use sum-of-squared-difference as the distance metric

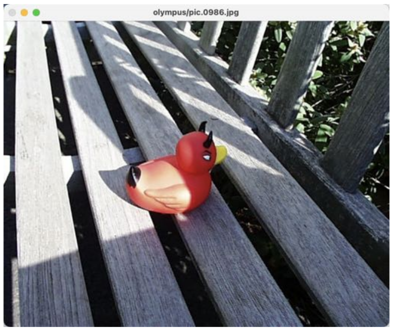
Target image: pic.1016.jpg

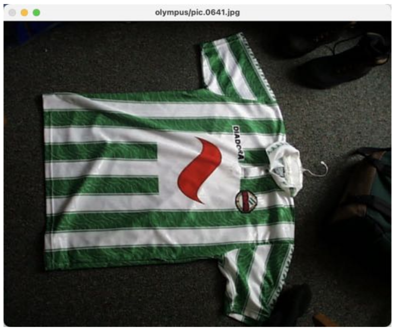


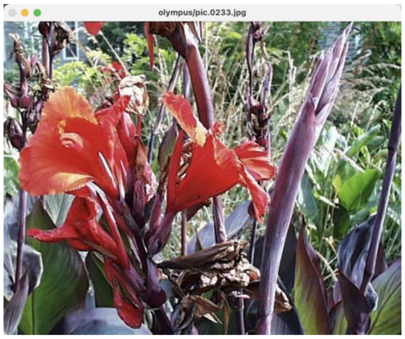
Result:



```
Processing directory olympus
Task 1 : the top 3 matches for the target image
    pic.1016.jpg is:
pic.0986.jpg
pic.0641.jpg
pic.0233.jpg
Program ended with exit code: 0
```







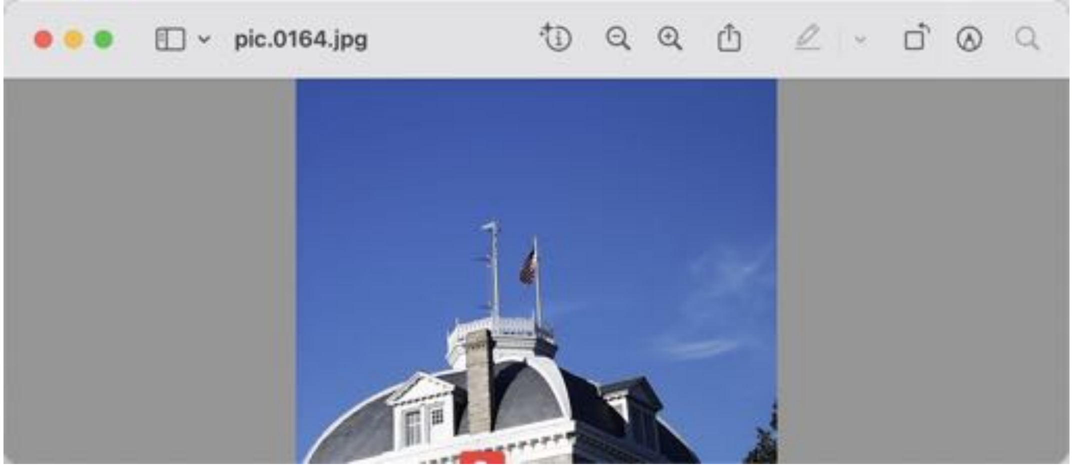
Task 2: Histogram Matching

Intuition: For this task, we do both rg chromaticity histogram using 16 bins and RGB histogram using 8 bins.

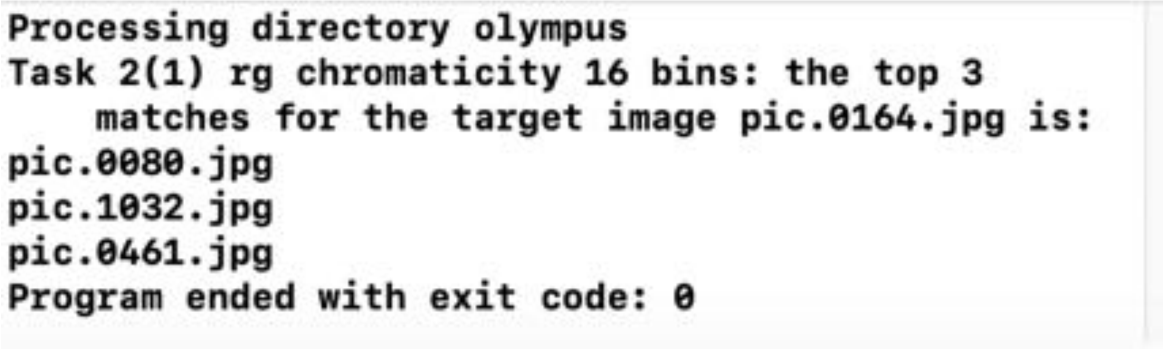
Section 1 (rg chromaticity histogram using 16 bins):

Feature vector: whole image rg chromaticity histogram using 16 bins
Distance metric: histogram intersection

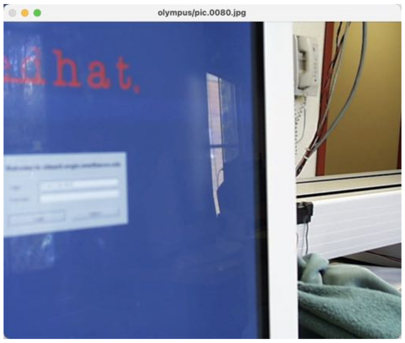
Target image: pic.0164.jpg

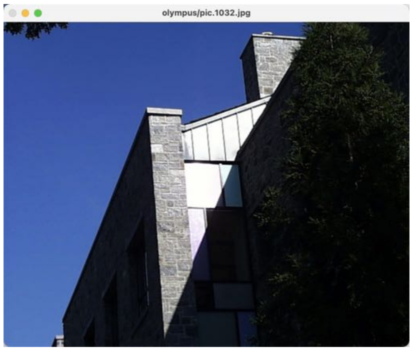


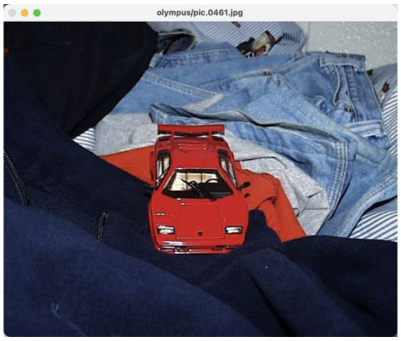
Result:



```
Processing directory olympus
Task 2(1) rg chromaticity 16 bins: the top 3
    matches for the target image pic.0164.jpg is:
pic.0080.jpg
pic.1032.jpg
pic.0461.jpg
Program ended with exit code: 0
```



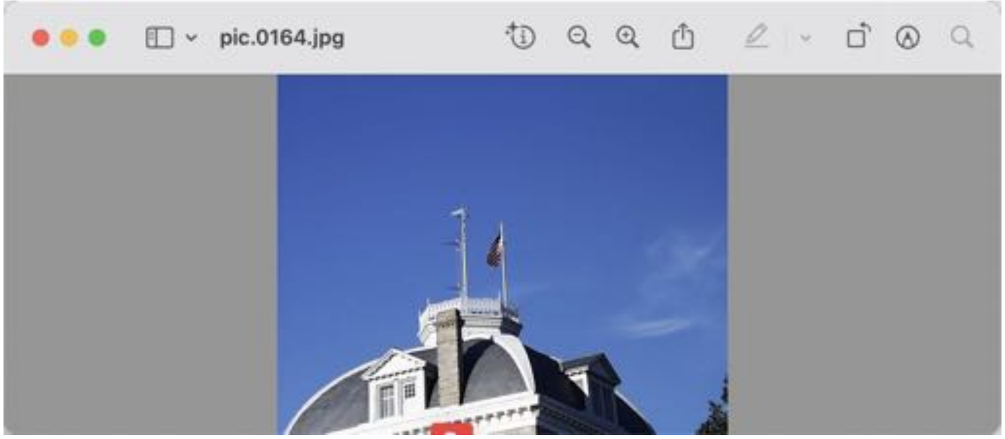




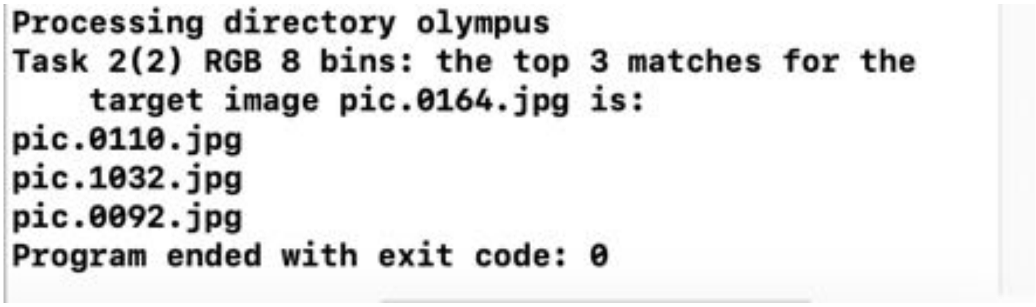
Section2 (RGB histogram using 8 bins):

Feature vector: whole image RGB histogram using 8 bins
Distance metric: histogram intersection

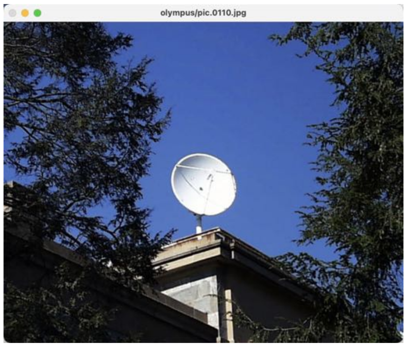
Target image: pic.0164.jpg

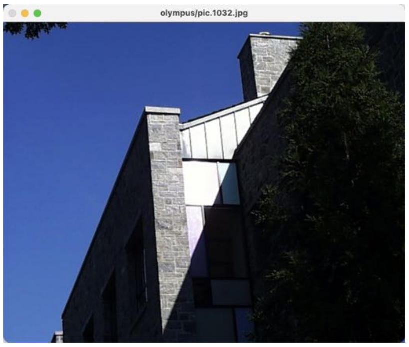


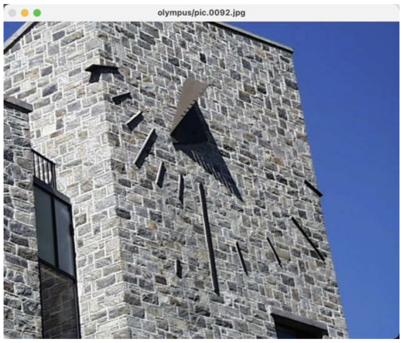
Result:



```
Processing directory olympus
Task 2(2) RGB 8 bins: the top 3 matches for the
    target image pic.0164.jpg is:
pic.0110.jpg
pic.1032.jpg
pic.0092.jpg
Program ended with exit code: 0
```



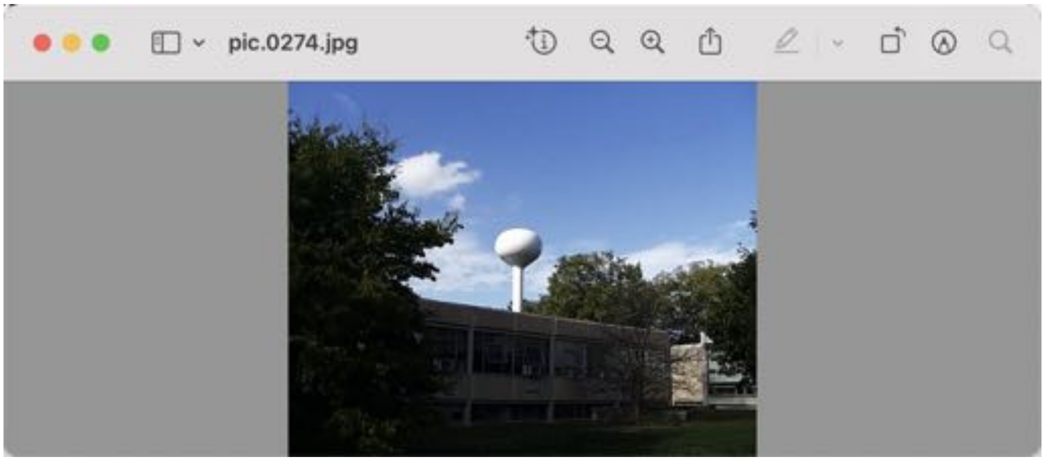




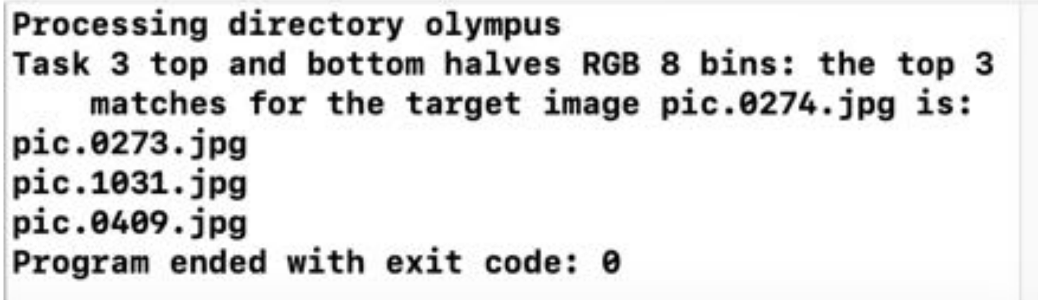
Task 3: Multi-histogram Matching

Feature Vector: Use two color histograms (RGB histogram using 8 bins for top and bottom halves of the image) as the feature vector.
Distance metric: histogram intersection, weight two histograms equally

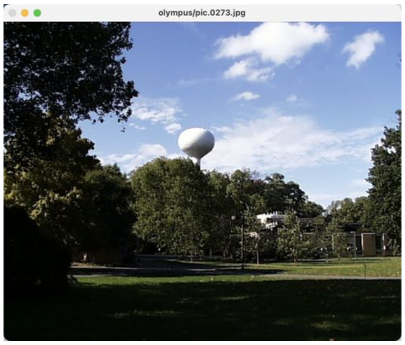
Target image: pic.0274.jpg

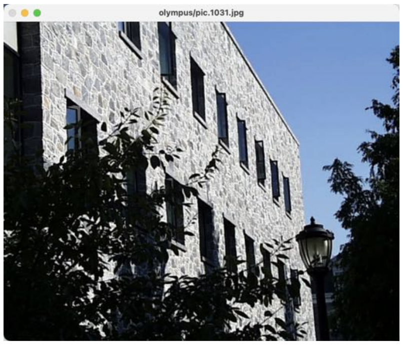


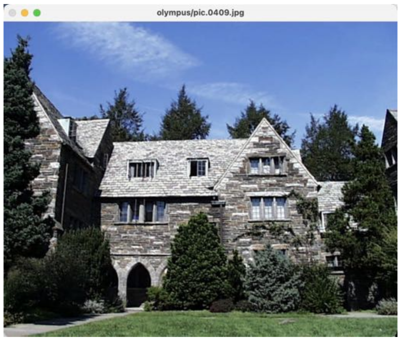
Result:



```
Processing directory olympus
Task 3 top and bottom halves RGB 8 bins: the top 3
    matches for the target image pic.0274.jpg is:
pic.0273.jpg
pic.1031.jpg
pic.0409.jpg
Program ended with exit code: 0
```





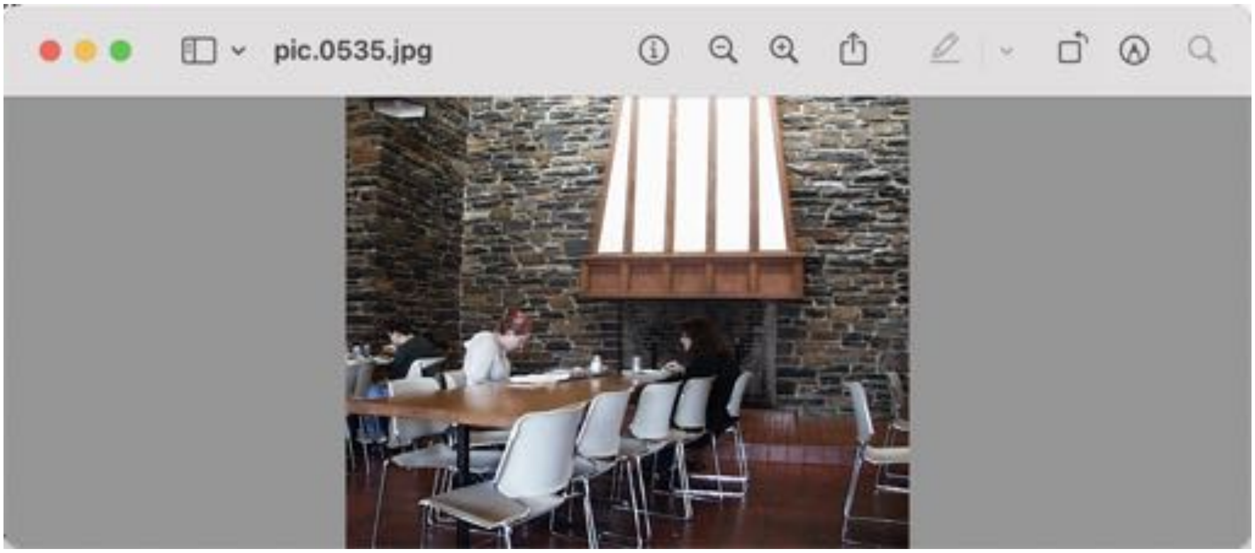


Task 4: Texture and Color

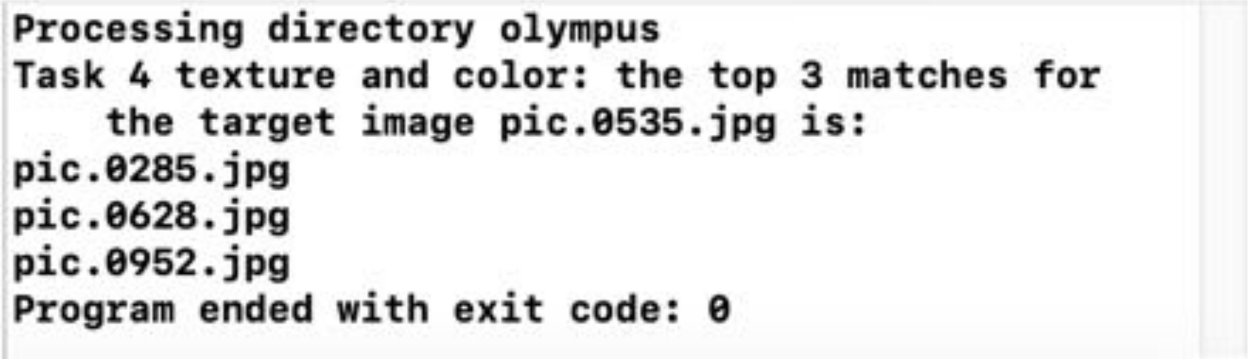
Feature Vector:
1. A whole image color histogram (whole image RGB histogram using 8 bins)
2. A whole image texture histogram (calculate the Sobel magnitude image and use a histogram of gradient magnitudes as your texture feature)

Distance metric:
histogram intersection, weight two histograms equally

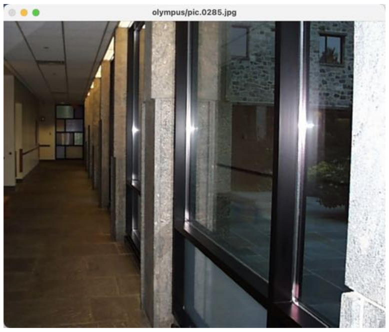
Target image:
pic.0535.jpg

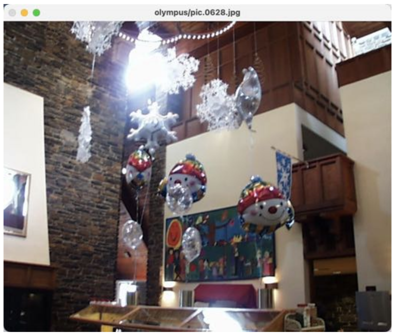


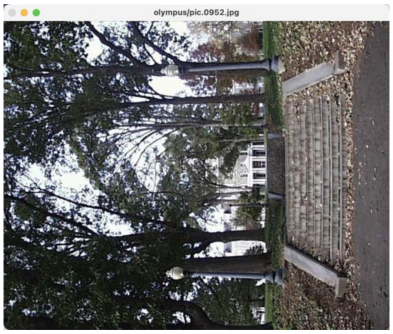
Result:



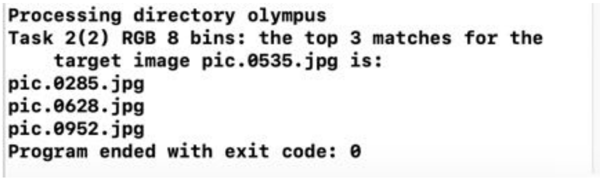
```
Processing directory olympus
Task 4 texture and color: the top 3 matches for
    the target image pic.0535.jpg is:
pic.0285.jpg
pic.0628.jpg
pic.0952.jpg
Program ended with exit code: 0
```



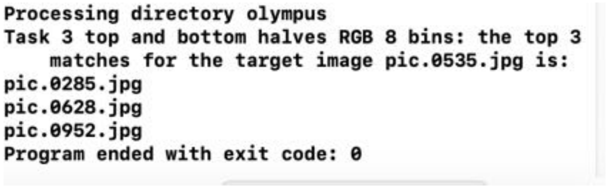




Compared with tasks 2 and 3:
We get the same result if we run pic.0535.jpg in task 2(RGB histogram 8 bins) and task 3.



```
Processing directory olympus
Task 2(2) RGB 8 bins: the top 3 matches for the
    target image pic.0535.jpg is:
pic.0285.jpg
pic.0628.jpg
pic.0952.jpg
Program ended with exit code: 0
```



```
Processing directory olympus
Task 3 top and bottom halves RGB 8 bins: the top 3
    matches for the target image pic.0535.jpg is:
pic.0285.jpg
pic.0628.jpg
pic.0952.jpg
Program ended with exit code: 0
```

Task 5: Custom Design:

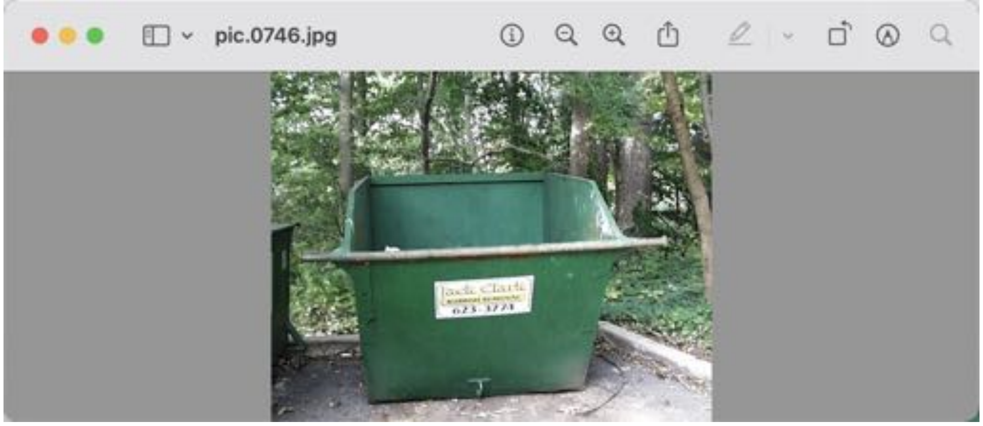
Intuition:
For this task, I chose the green dumpsters as the training and detection dataset. Here are features of the green dumpsters I observed:

1. The dumpsters in the dataset are green.
2. The texture of the dumpster is smooth.
3. The dumpsters in the dataset are all in the center of image
4. The dumpsters in the dataset are all placed on grey, concrete ground, the ground usually at the bottom of the image(1/10 bottom part).

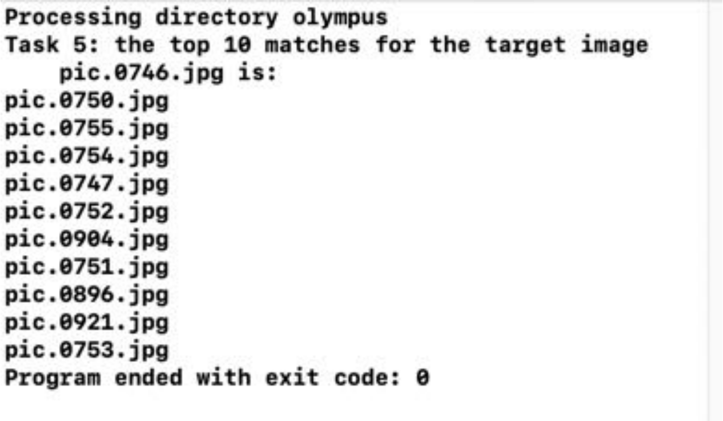
Therefore, I cut the center and bottom part of the target image, calculate the RGB histogram and texture histogram for both parts, so I have four feature vectors. In distance matric, I weight more the center part than the bottom part, because our focus is dumpsters. By these way, I got good results.

Feature Vector:
1. A color histogram for image center (RGB histogram using 8 bins)
2. A texture histogram for image center (calculate the Sobel magnitude image and use a histogram of gradient magnitudes as your texture feature)
3. A color histogram for the image bottom 1/10 part (RGB histogram using 8 bins)
4. A texture histogram for the image bottom 1/10 part (calculate the Sobel magnitude image and use a histogram of gradient magnitudes as your texture feature)
Distance metric: histogram intersection, weight 4 histograms in 4:4:1:1

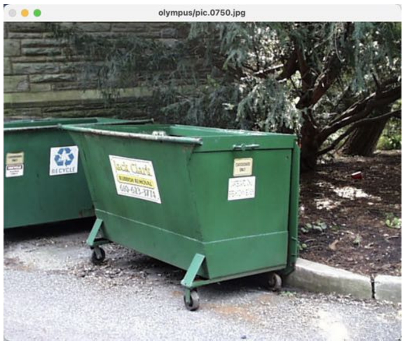
Target image1: Pic.0746.jpg

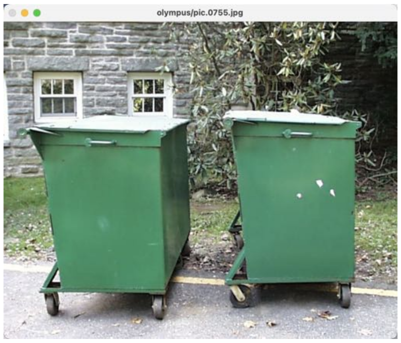


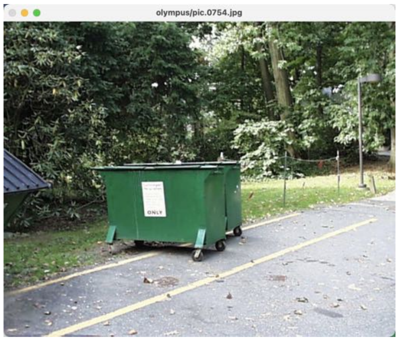
Result:

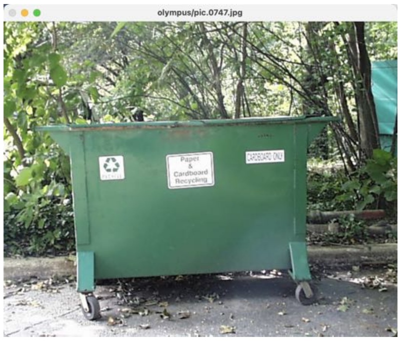


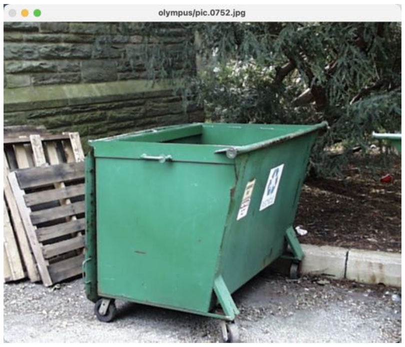
```
Processing directory olympus
Task 5: the top 10 matches for the target image
    pic.0746.jpg is:
pic.0750.jpg
pic.0755.jpg
pic.0754.jpg
pic.0747.jpg
pic.0752.jpg
pic.0904.jpg
pic.0751.jpg
pic.0896.jpg
pic.0921.jpg
pic.0753.jpg
Program ended with exit code: 0
```

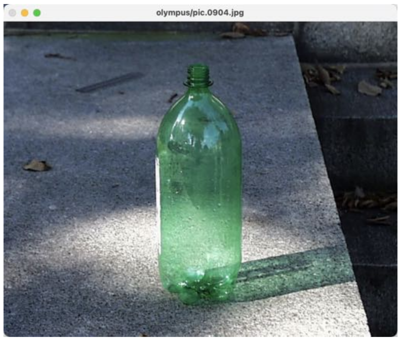


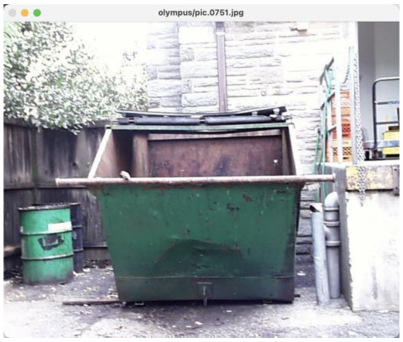


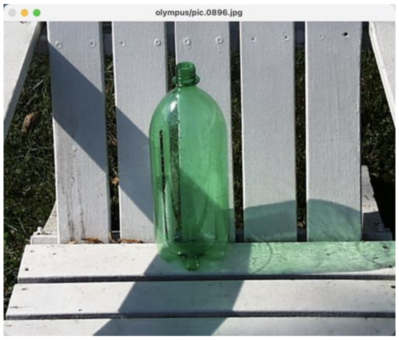


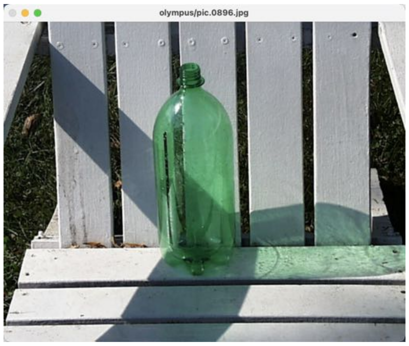


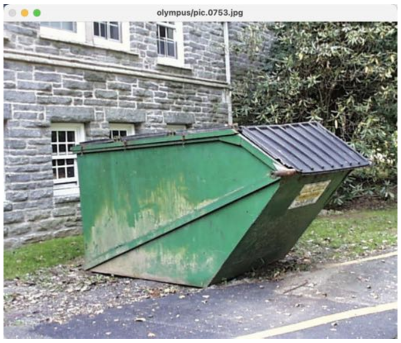




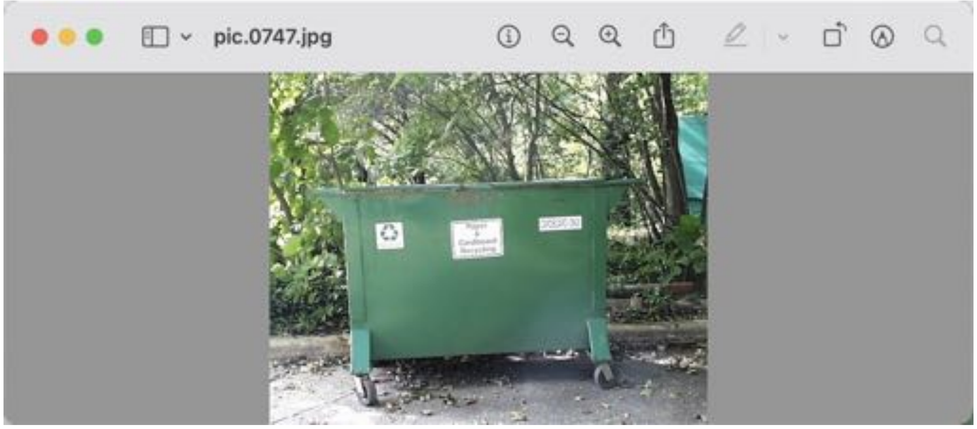




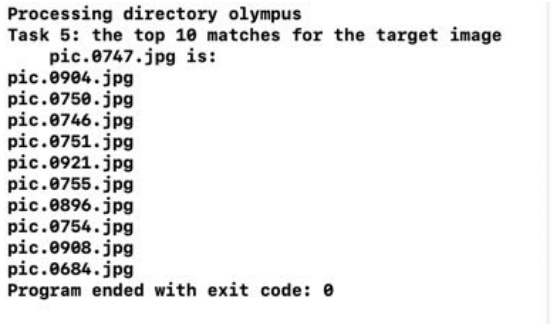




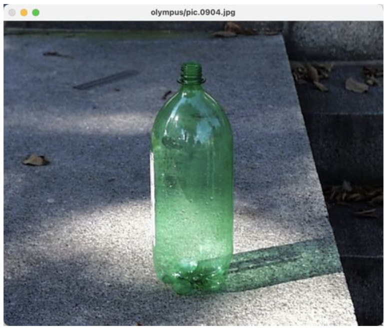
Target image2: Pic.0747.jpg

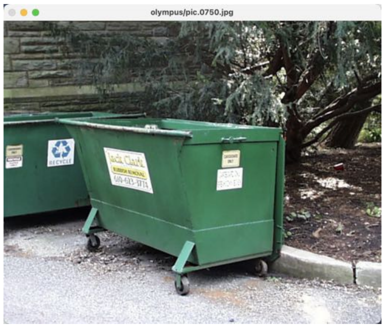


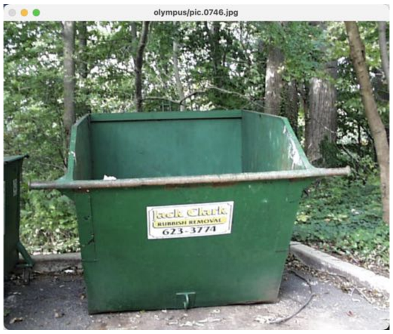
Results:

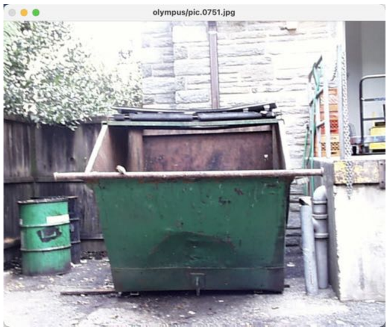


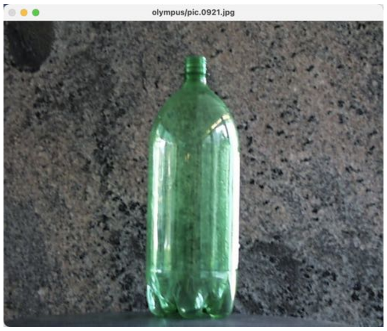
```
Processing directory olympus
Task 5: the top 10 matches for the target image
    pic.0747.jpg is:
pic.0904.jpg
pic.0750.jpg
pic.0746.jpg
pic.0751.jpg
pic.0921.jpg
pic.0755.jpg
pic.0896.jpg
pic.0754.jpg
pic.0908.jpg
pic.0684.jpg
Program ended with exit code: 0
```

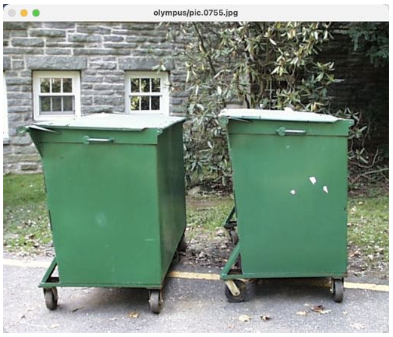


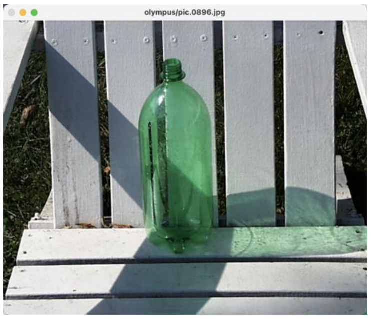


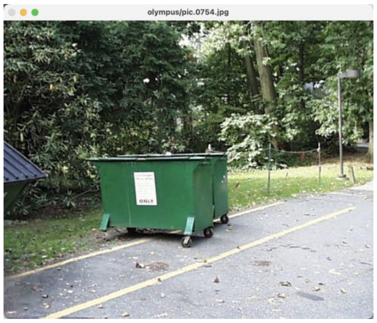


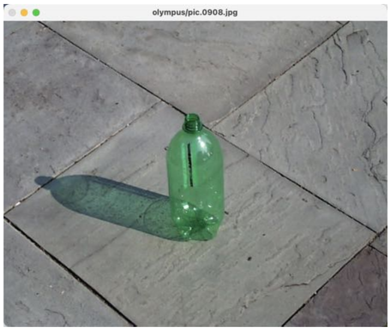


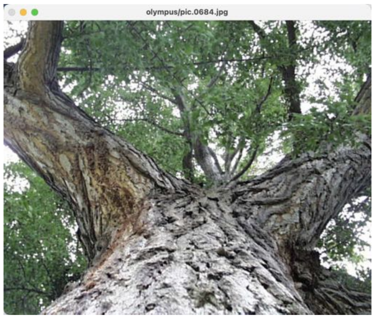












Extension:

“There are also lots of pictures of blue trash can bins. How many of these can your system recall given a target image that contains one?”

Intuition:

Here are features of the blue trash can bins I observed:

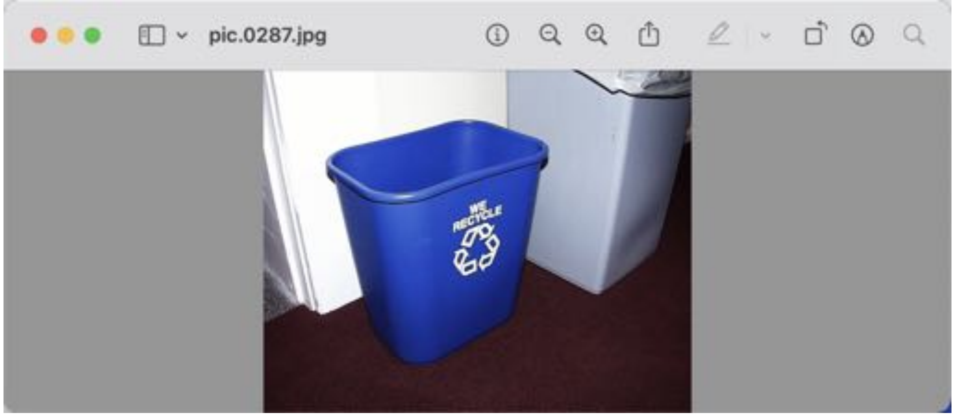
1. The blue trash can bins in the dataset are blue.
2. The texture of the blue trash can bins is smooth.

3. The blue trash can bins in the dataset are mostly in the center of image

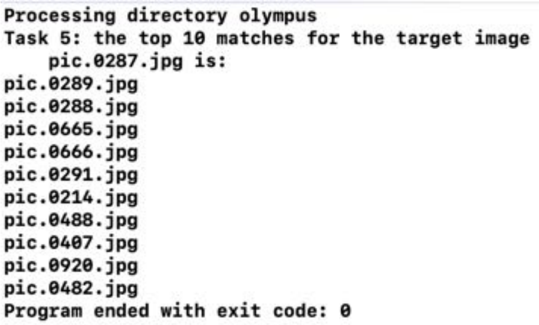
Therefore, I cut the center of the target image, calculate the RGB histogram and texture histogram for the center, I have two feature vectors. In distance matric, I weight the texture histogram and the RGB histogram equally. By this way, I got these results.

Feature Vector:
1. A color histogram for image center (RGB histogram using 8 bins)
2. A texture histogram for image center (calculate the Sobel magnitude image and use a histogram of gradient magnitudes as your texture feature)
Distance metric:
histogram intersection, weight histograms equally.

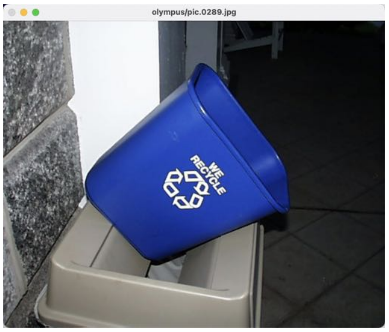
Target image: Pic.0287.jpg

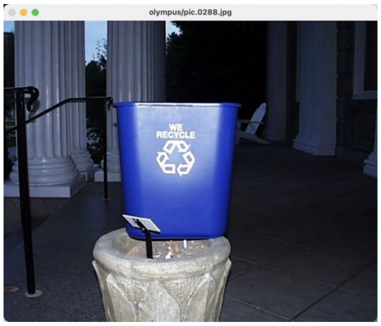


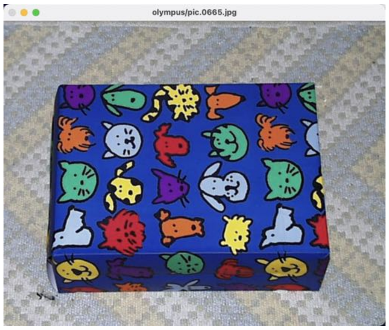
Results:

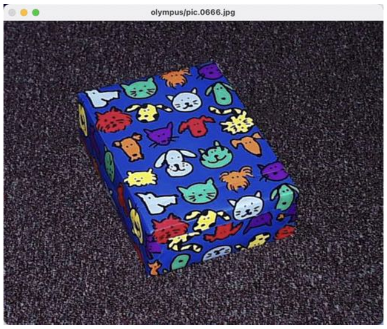


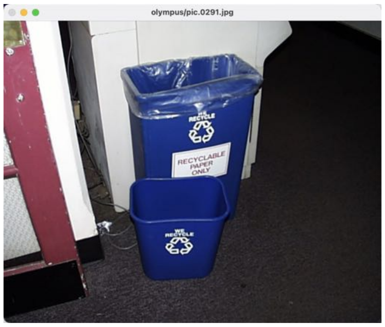
```
Processing directory olympus
Task 5: the top 10 matches for the target image
    pic.0287.jpg is:
pic.0289.jpg
pic.0288.jpg
pic.0665.jpg
pic.0666.jpg
pic.0291.jpg
pic.0214.jpg
pic.0488.jpg
pic.0407.jpg
pic.0920.jpg
pic.0482.jpg
Program ended with exit code: 0
```

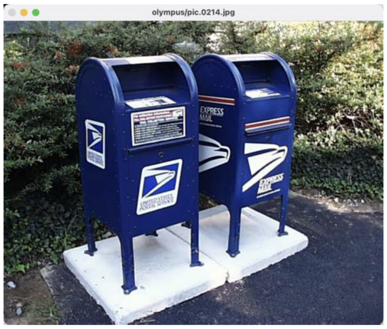


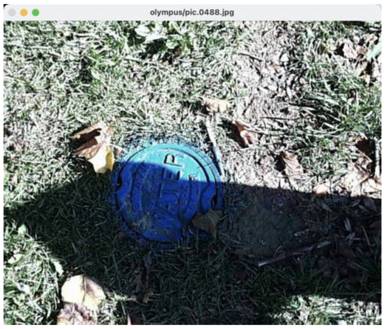


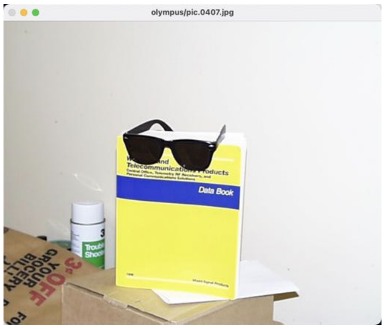


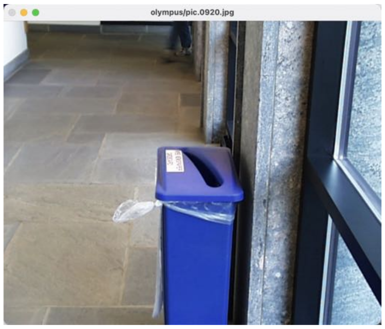


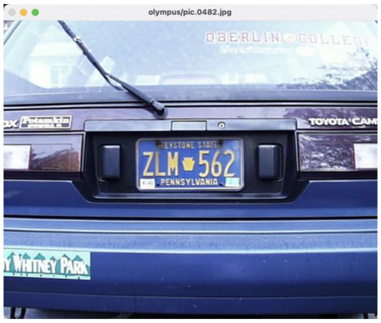












Reflection of what I learned:

From this project, I learned these things:

1. How to segment the specific region of image based on our needs. Sometimes, we want to focus on a specific region of the image and ignore other noise in the background, so cutting and extracting the region in the image is important.
2. how to calculate different histograms such as RGB color histogram, rg chromaticity histogram, 1d greyscale histogram

3. How to calculate different distance metrics such as sum-of-squared-difference and histogram intersection
4. How to weight difference histograms in a multi-histogram matching
5. Find features of a specific object visually, and design different histograms for different parts of the image.

Acknowledgement

Maxwell's lecture notes, by *Dr. Bruce A. Maxwell*

Computer Vision: Algorithms and Applications*, 2nd ed, by Richard Szeliski*

OpenCV Tutorials*: https://docs.opencv.org/4.x/d9/df8/tutorial_root.html*

Computer Vision, *by Shapiro and Stockman*

Texture Classification: Are Filter Banks Necessary, *by Manik Varma and Andrew Zisserman*