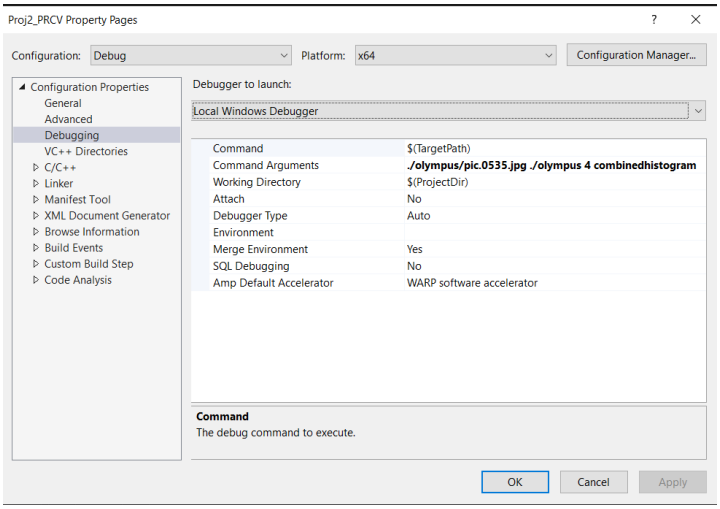
Project 2: Content-based Image Retrieval

Project Overview:

This project involves content matching and pattern recognition. Here I have explored many feature extraction methods and distance metrics with top 'n' matches. I have computed baseline matching, histogram, and multi-histogram matching, and color-texture-based matching, etc. The assignment takes 4 command line arguments: <target_image> <image_directory> <N_results> <matching_type>

Here, the matching type options are: baseline, histogram, multihistogram, combinedhistogram and CBIR computed for each one of the tasks respectively.

I am using Visual Studio 2019's Debugging mode to input the command line arguments for faster execution of code.



1. **Baseline Matching**

Here a 9x9 square was used from the center of the image as the feature vector and I computed the distances by the sum of squared differences.

Original Image: pic.1016.jpg

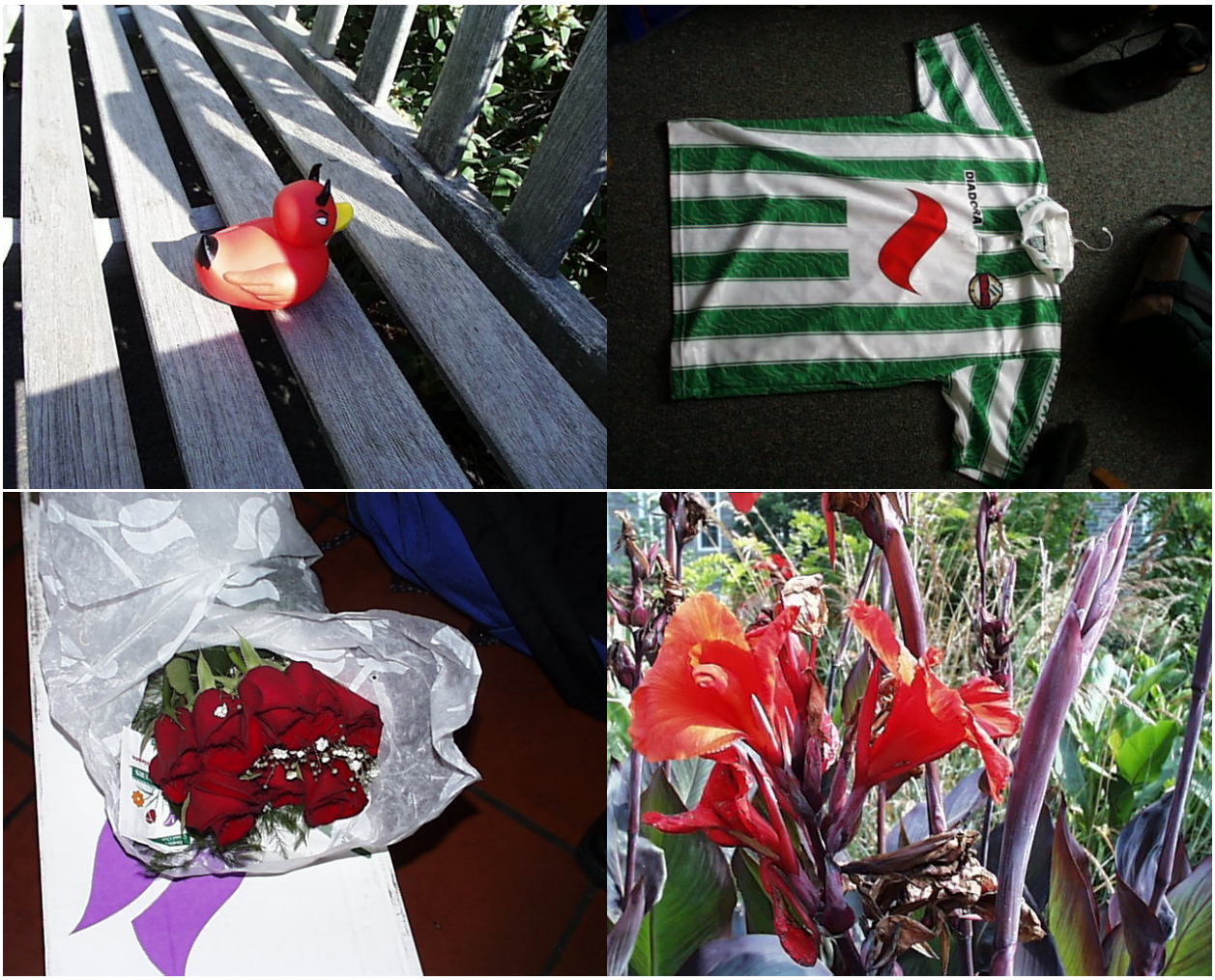
Matched Images:

pic.0986.jpg,

pic.0641.jpg,

pic.0547.jpg,

pic.0233.jpg



2. Histogram Matching

The `extract_features_histogram` function calculates the normalized color histogram of an image and stores it in a vector. It divides the image into 8x8x8 color bins, counts the number of pixels in each bin, and then normalizes the resulting histogram.

The `distance_histogram` function calculates the distance between two histograms using the histogram intersection method. It sums the minimum values of each bin between the two histograms and subtracts the result from 1 to obtain the final distance value.

Original Image: pic.0164.jpg

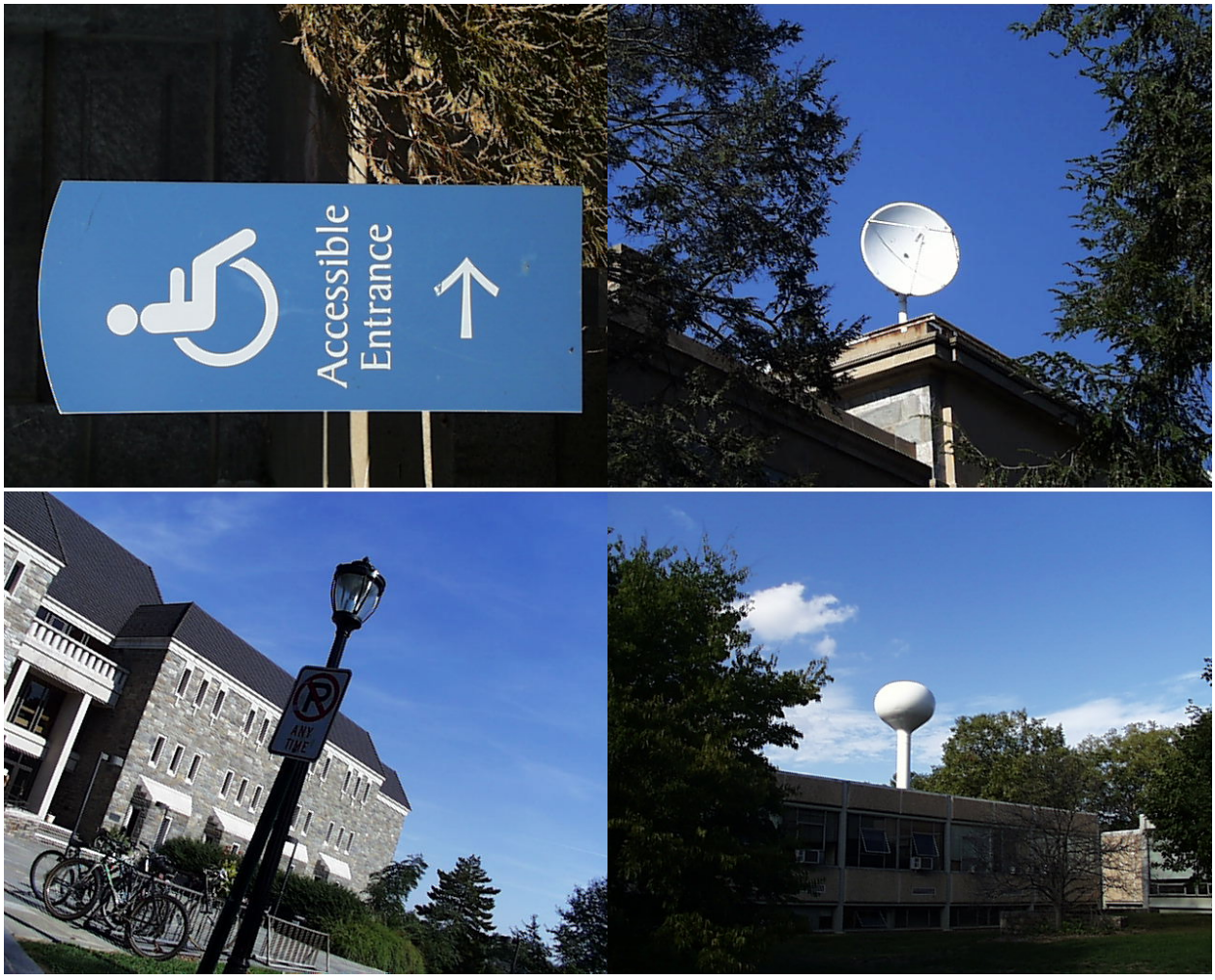
Matched Images:

pic.0976.jpg,

pic.0110.jpg,

pic.0426.jpg,

pic.0274.jpg



3. Multi-Histogram Matching

Original Image: pic.0274.jpg

Matched Images:

pic.1055.jpg,

pic.0955.jpg,

pic.0209.jpg,

pic.0273.jpg

I used a whole-image histogram and a second histogram that looks at only the center of the image and Manhattan distance as the distance metric.

4. Texture and color

Original Image: pic.0535.jpg

Matched Images:

pic.0853.jpg,

pic.0980.jpg,

pic.0629.jpg,

pic.0950.jpg

5. Custom Design and Extension

Here, I extract color histograms from task 2, Hu moments, and Local Binary Pattern (LBP) histograms from an input image, normalize each feature component, combine the features into a single feature vector, and calculate the distance between the feature vectors using a weighted sum of the differences between the feature components.

Input Image: pic.0013.jpg

Top 10 Matched Images:

pic.0024.jpg,

pic.0025.jpg,

pic.0018.jpg,

pic.0017.jpg,

pic.0014.jpg,

pic.0015.jpg,

pic.0019.jpg,

pic.0376.jpg,

pic.0021.jpg,

pic.0545.jpg

Input Image: pic.0746.jpg

Top 10 Matched Images:
pic.0750.jpg
pic.0747.jpg
pic.0755.jpg
pic.0866.jpg
pic.0754.jpg
pic.0101.jpg
pic.0688.jpg
pic.0752.jpg
pic.0159.jpg
pic.0904.jpg

My solution somewhat worked in retrieving green trash cans. It was able to retrieve 5/10 images with green trash cans and the other images retrieved were images with a lot of green color components in it (eg. green bottle and trees/bushes, etc). This can be due to the fact that there might be too many bins in the histogram.

Reflection

This was a fun project to work with. I learned a lot about different types of histograms, matching, textures, and applying distance metrics to obtain similarity matches. My only issue was the later tasks takes about 1-2 minutes to run and obtain the results.