CS5330: Pattern Recognition and Computer Vision - Project 2 Report

Project 2: Content-Based Image Retrieval

Brief Description of the Project:

Building upon the last project, this project aims to develop the learning process further by introducing us to histogram matching or pattern recognition. We are still manipulating and analyzing images at pixel level while trying to develop our own data types. The overall task for the project is to take a target image from the database provided and find images with similar content. We are not to use neural networks or object recognition methods. While this complicates the task by a significant measure, it is to help us develop a core understanding and develop intuition on how images and pixels behave when subjected to different filters. This focuses more on generic characteristics such as colour, texture and their spatial layout aiming to develop more understanding for colour spaces, histograms, spatial features, and texture features.

(Note: The first image is the target image and the images that follow are their top matches in the increasing order of the distance. Same results can be derived by running the code as well.)

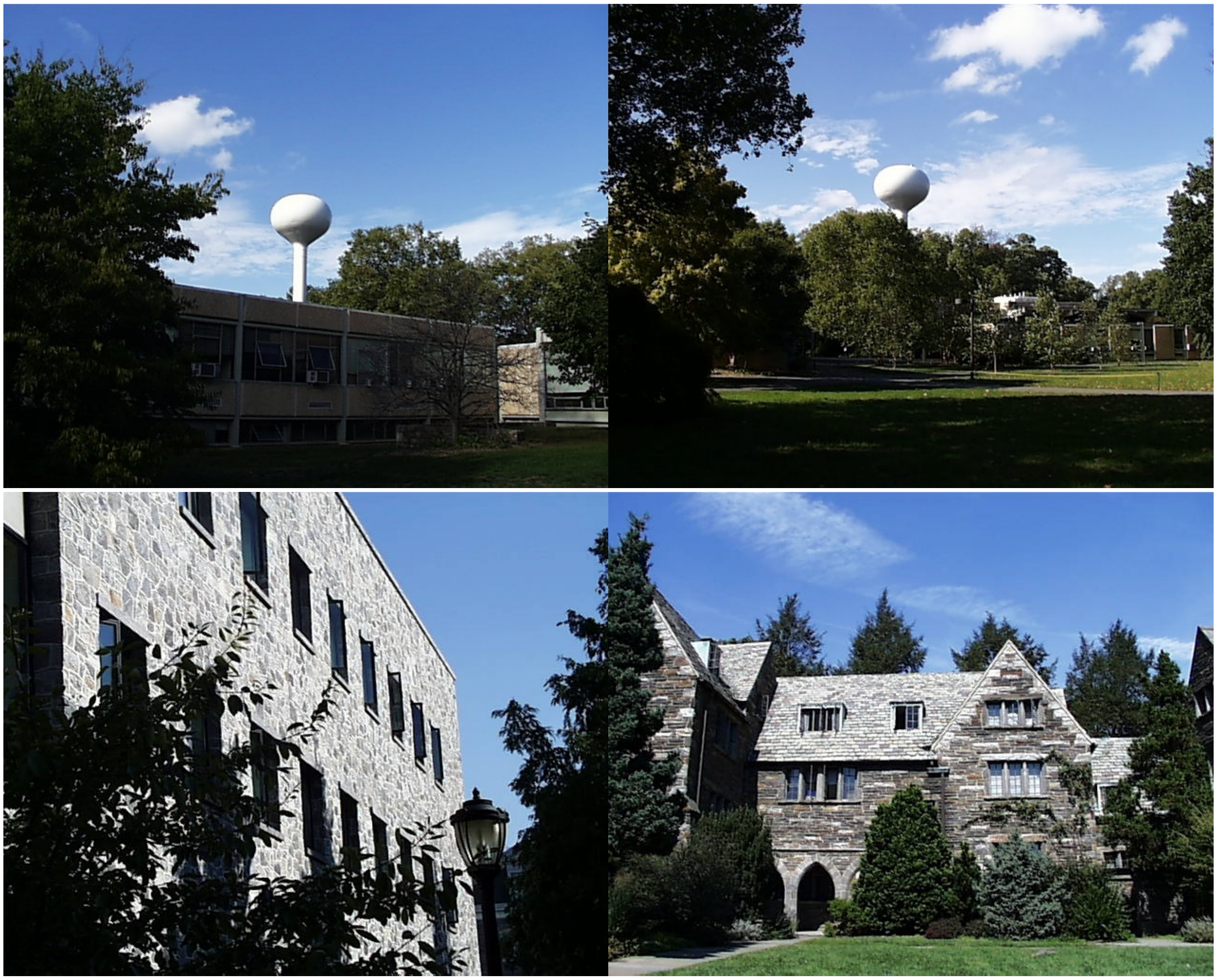
Required Results 1: Baseline Matching

In this program, 9x9 square matrices were placed in the middle of the image to be then used as a feature vector. The distance metric used is the sum squared difference. The first one is the target image while the rest are the top 3 matches in the increasing order of the distance.

Required Results 2: Histogram Matching

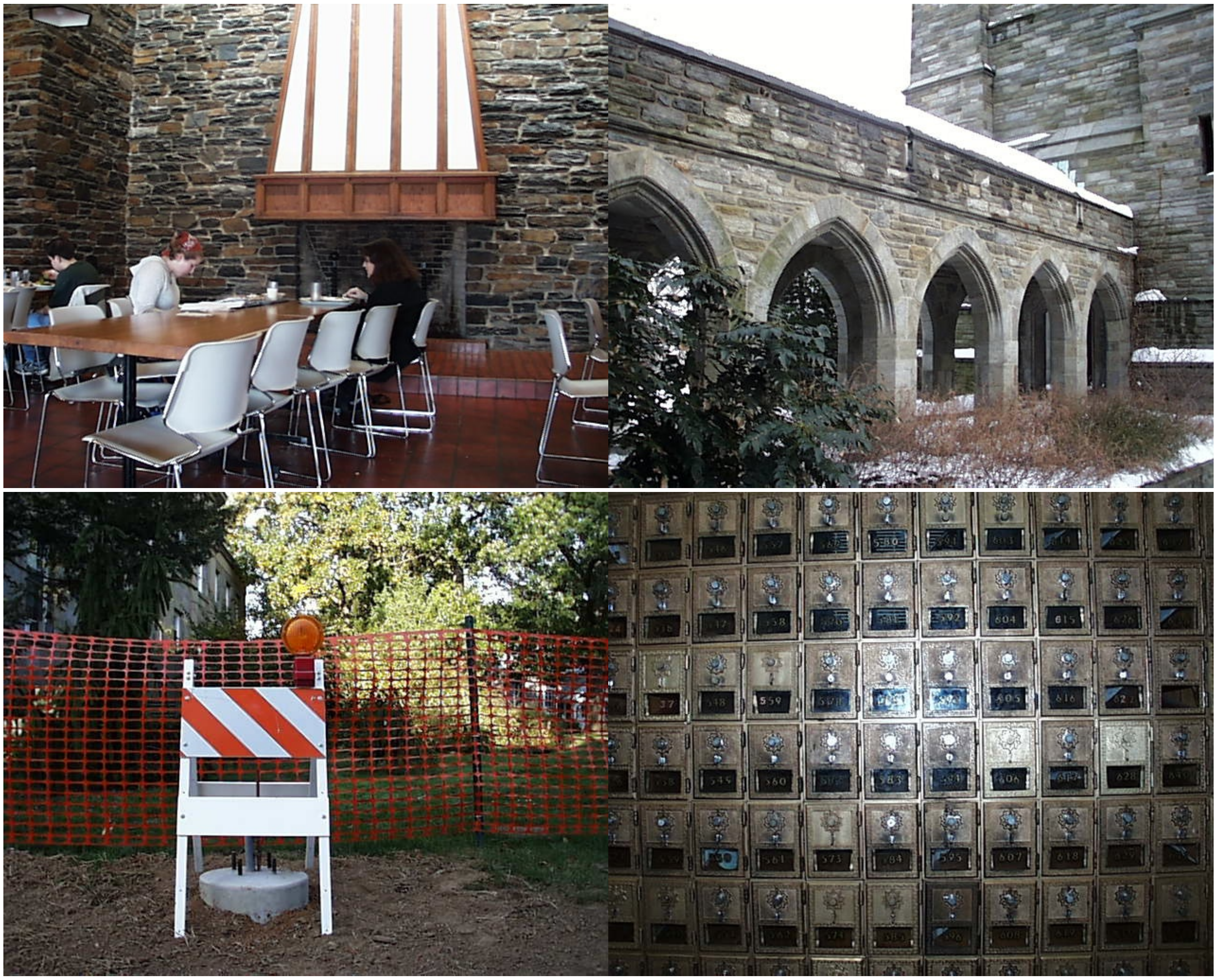
A single normalized 2D rg chromaticity histogram with 16 bins for each r and g channel is used as the feature vector for the above images. Histogram intersection is used as the distance metric. The first image is the target image and then the images are in the increasing order of distance.

Required Results 3: Multi-histogram Matching



The first image is the target image and the second third and the fourth are the top 3 matches generated for the first image in the increasing order of distance. The results were generated using two 3d RGB Histograms, representing the top and bottom halves of the image using 8 bins for each channel of the histogram. The top half was compared with the top while the bottom was compared with the bottom. The distance metric chosen for this purpose is histogram intersection. The parts are disjoint.

Required Results 4: Texture and Color



The first image is the target image and the second, third and fourth are the top matches in the increasing order of distance. A whole image colour 3D RGB histogram is used as a feature vector along with the histogram of a gradient magnitude image. To form the gradient magnitude image, the functions developed in Project 1 have been used. Sobel X and Sobel Y are calculated and then the magnitude gradient image is generated. Histograms are developed of both the original and magnitude gradient image. The first acts as a colour feature while the other works as a texture feature and are equally weighted to calculate the final result.

When compared to tasks 2 and 3, the images differ in the texture component. Analysing the same image for tasks 2 and 3 gives matches that are similar in colours like that of a pile of dirt that a bulldozer is taking out or that of plain furniture which is also brown. But this captures the texture of the image which has bricks and the matches display other images that are similar in texture as well as colour.

Required Results 5: Custom Design + Extension

I added an extension in the title as I wanted to experiment a bit with this part as it left everything to me and I also choose to apply Laws Filter (R5L5) in this case to improve the results. It is implemented as a separable filter.

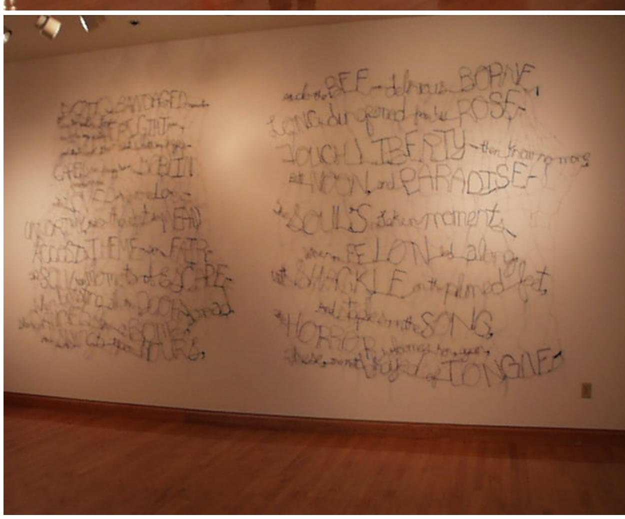
Colour RGB histograms are calculated for both the original image and laws filtered image and the distance metric used is the histogram intersection. While I also experimented with sum-squared difference and other distance metrics (whose functions I have left untouched in the header file in the hope they might come useful in the later projects) but histogram intersection had the best performance.

According to the theory, R5 is designed to notice the ripples and L5 is used for edges. The combined convoluted filter R5L5, thus, becomes good with detecting a combination of ripples and edges which is why I have used the below specific examples to demonstrate. The first result takes a glossy wooden floor with white walls and doors which have defined texture while the reflection gives it a ripple effect and thus this filter does complete justice to retrieve the matching images based on just one training image.

The second result is based on the experimental reading of the R5L5 filter which has been extensively used on finding images with leaves, trees, grass forests again because of the unique combination of the R5L5 filter and this result is also generated based on just a single training image.

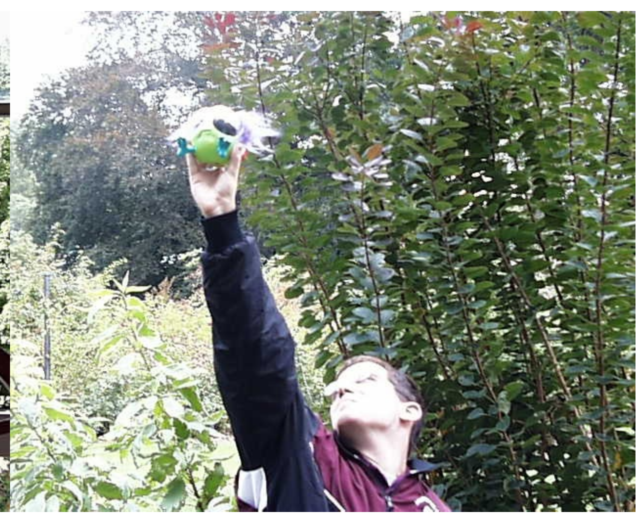
I felt like adding innovation to the custom design to the best of my abilities and hence combined Task 5: Custom Design with an extension of sorts.

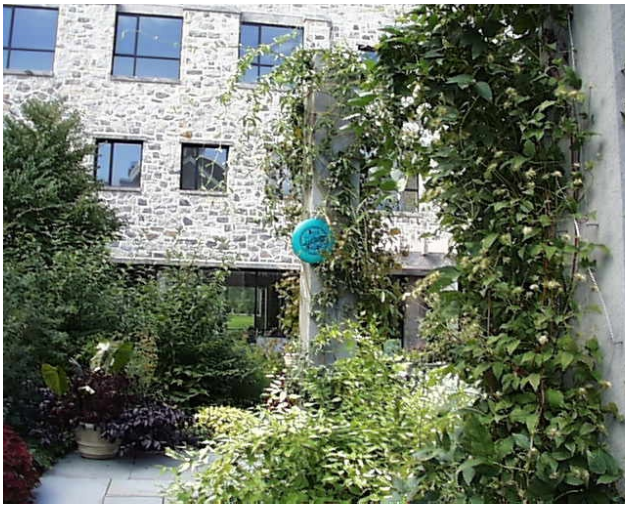
Result 1:



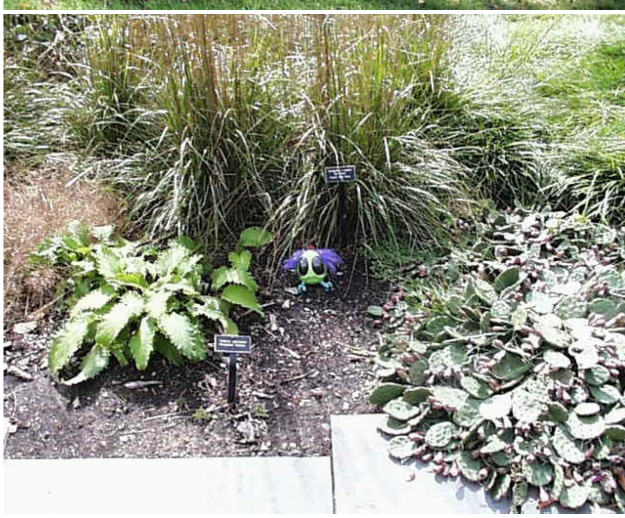
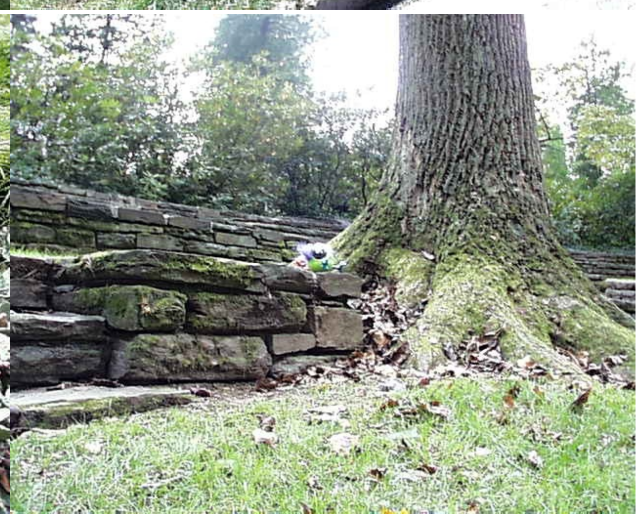
The first image is the target image (pic.0013.jpg) and the rest are the top 10 matches (including the original image) in the increasing order of distance.

Result 2:







The first image is the target image (pic.0065.jpg) and the rest are the top 10 matches (including the original image) in the increasing order of distance.

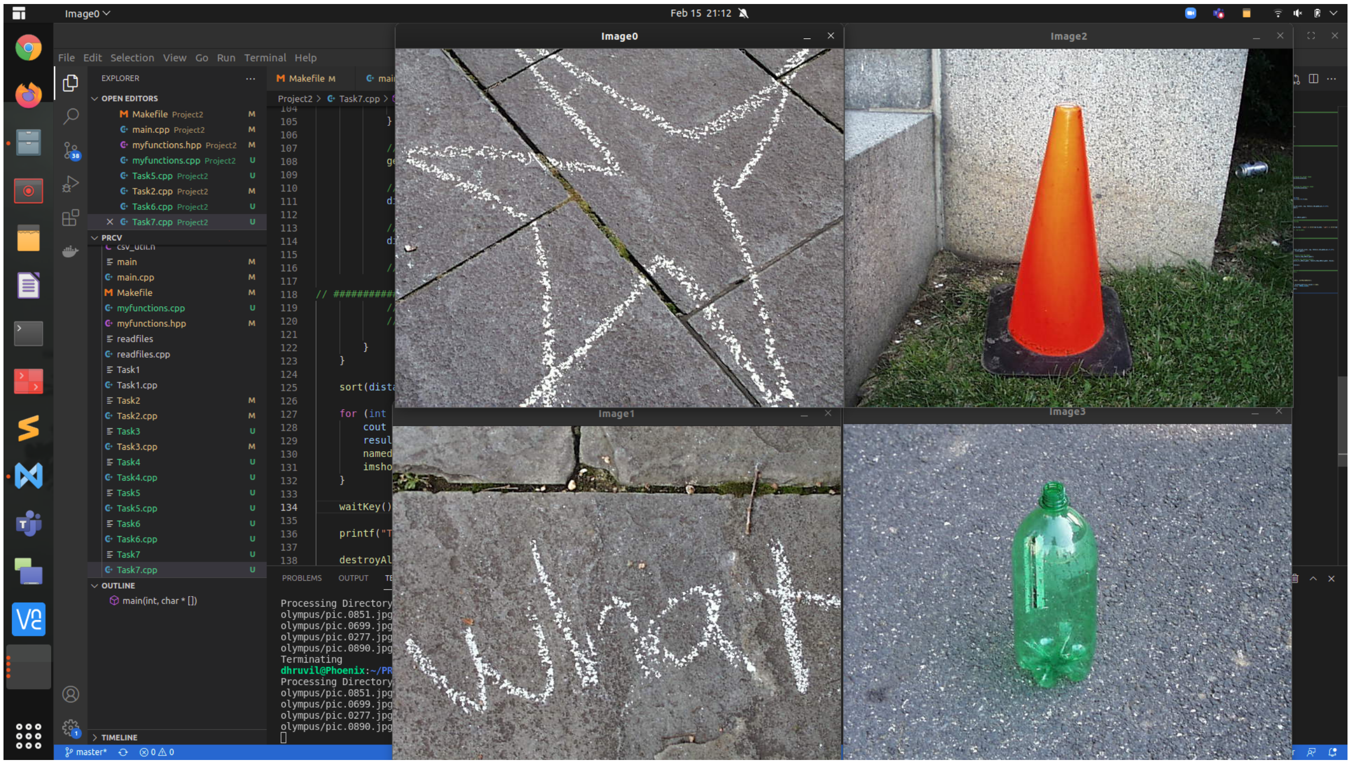
Extension 1: Another approach

This extension is but another approach to Task 2 of Histogram Matching. Instead of 2D rg chromaticity histogram, a complete RGB histogram is taken as the feature vector for this target image (the first one) and histogram intersection is used as a distance metric. Also, the bins are reduced from 16 to 8. Through comparison, we can observe that a complete RGB histogram captures different components as compared to the rg histogram and reducing the number of bins also greatly affects the results. In the result of Task2, the texture of the image or the colour isn't as neatly captured as it is done through this extension. Of course, it also depends on the type of image and the spatial distribution of pixels on how a histogram affects the image but in this case, I can safely conclude that using an RGB chromaticity histogram with 8 bins Is a better choice to capture the essence of the image.

Extension 2: Laws Filter Extension

As I had already seen the results of Laws filter, I couldn't keep myself from trying another combination and seeing their effects on the texture. Hence, I have implemented an L5E5 filter to the target image to get the above results. Two RGB colour histograms are used on the original and laws filtered image and histogram intersection is used as the distance metric. 0.3 weightage is given to the feature metric of original while 0.7 is given to the feature metric of the Laws filtered feature vector.

Extension 3: Gabor Filter Extension



This is a rudimentary implementation of the Gabor filter. This can be improved a lot and this doesn't really capture the beauty of the Gabor filter but I was trying it out and this is how it fares in comparison to the Laws filter.

Reflection:

This project was more of an S-learning curve for me. The first tasks came on intuitively and didn't take much time while the other took a lot of time, work and effort. I am getting more comfortable with using the OpenCV package with C++ and the language itself. I now think better about memory allocation, and suitable data types for a particular purpose rather than just aiming blindly in the dark. I am more comfortable with image manipulation at pixel levels, have an intuitive idea about histograms and how different feature vectors and distance metrics impact the overall results of an image. I also got to learn about laws filters and Gabor filters in practice. It is fun to learn more about thinking of images as signals. While this is not the most satisfactory work I have done, I am happy that I am able to achieve this kind of growth through this course. I am definitely going to improve upon the final extensions.

Acknowledgement:

I would like to acknowledge the contribution of Professor Bruce Maxwell whose lessons and programming session helped me get started on this project as well as helped clear the fundamentals of the histogram, distance metrics and other filters. Official documentation of OpenCV has been referred rigorously to bring this project to fruition. I would also acknowledge the support of my classmates Sumegha, Ravina and Husain for various brainstorming sessions that ultimately helped me to gain a deeper understanding of the subject.