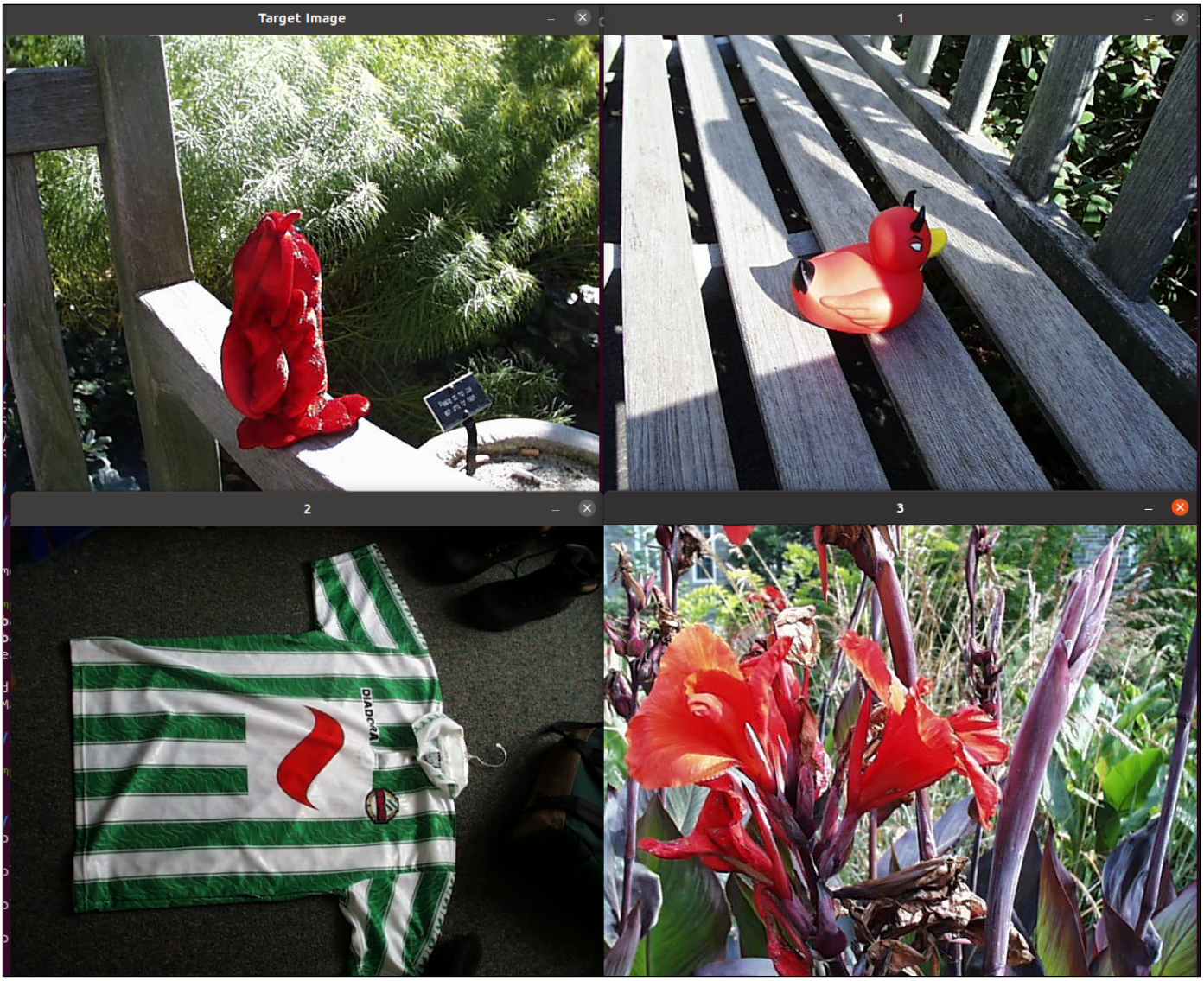
Project 2: Content-based Image Retrieval

Created by Ravina Lad, last modified on Feb 15, 2022

Summary :

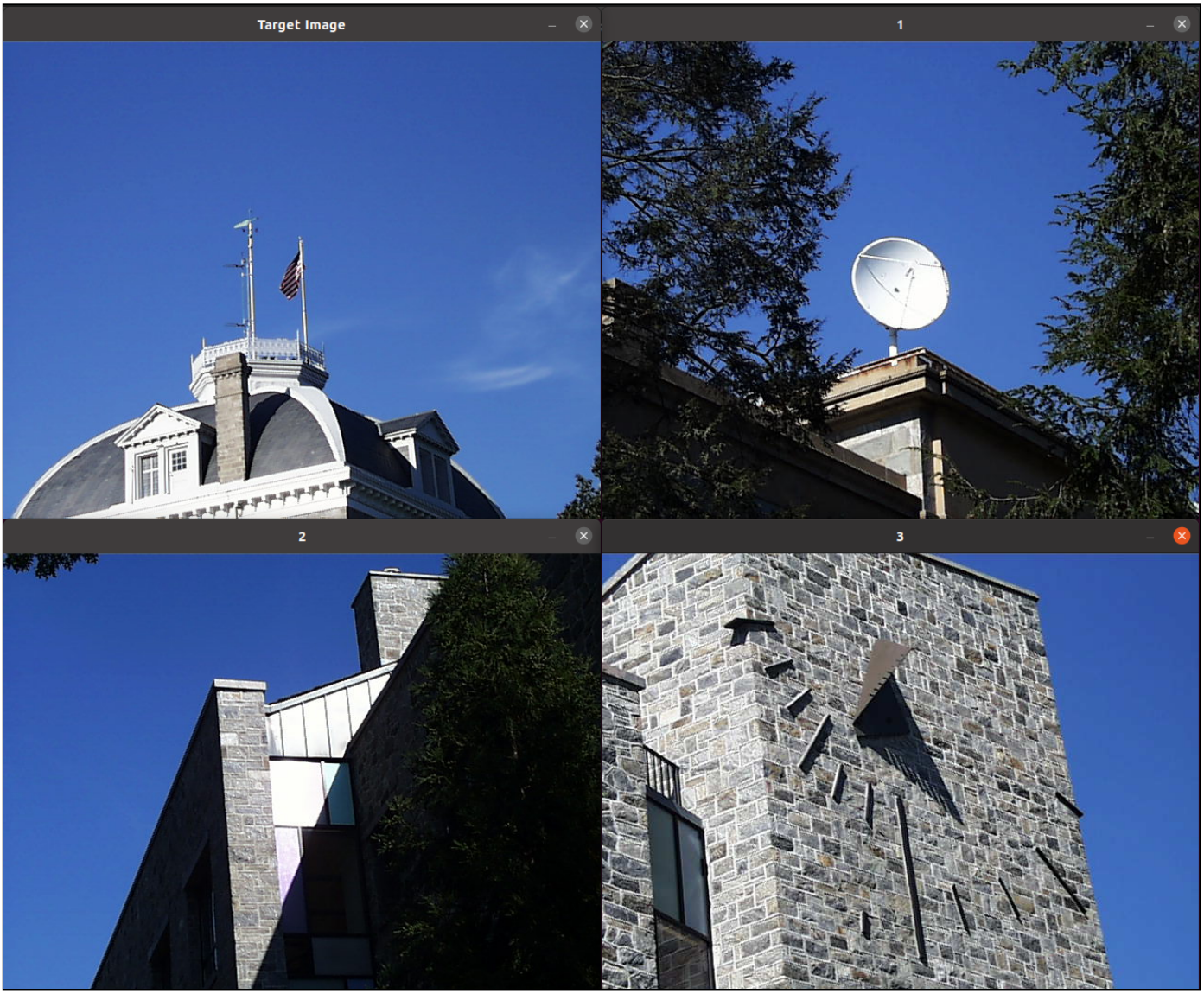
For this project, I implemented baseline matching, RGB, rg chromaticity histograms, also multi-histograms, Sobel magnitude texture, Laws texture, and Gabor texture for different tasks. Also, compared different distance metrics viz. sum of squared difference, histogram intersection, Chi-square, and Correlation distances.

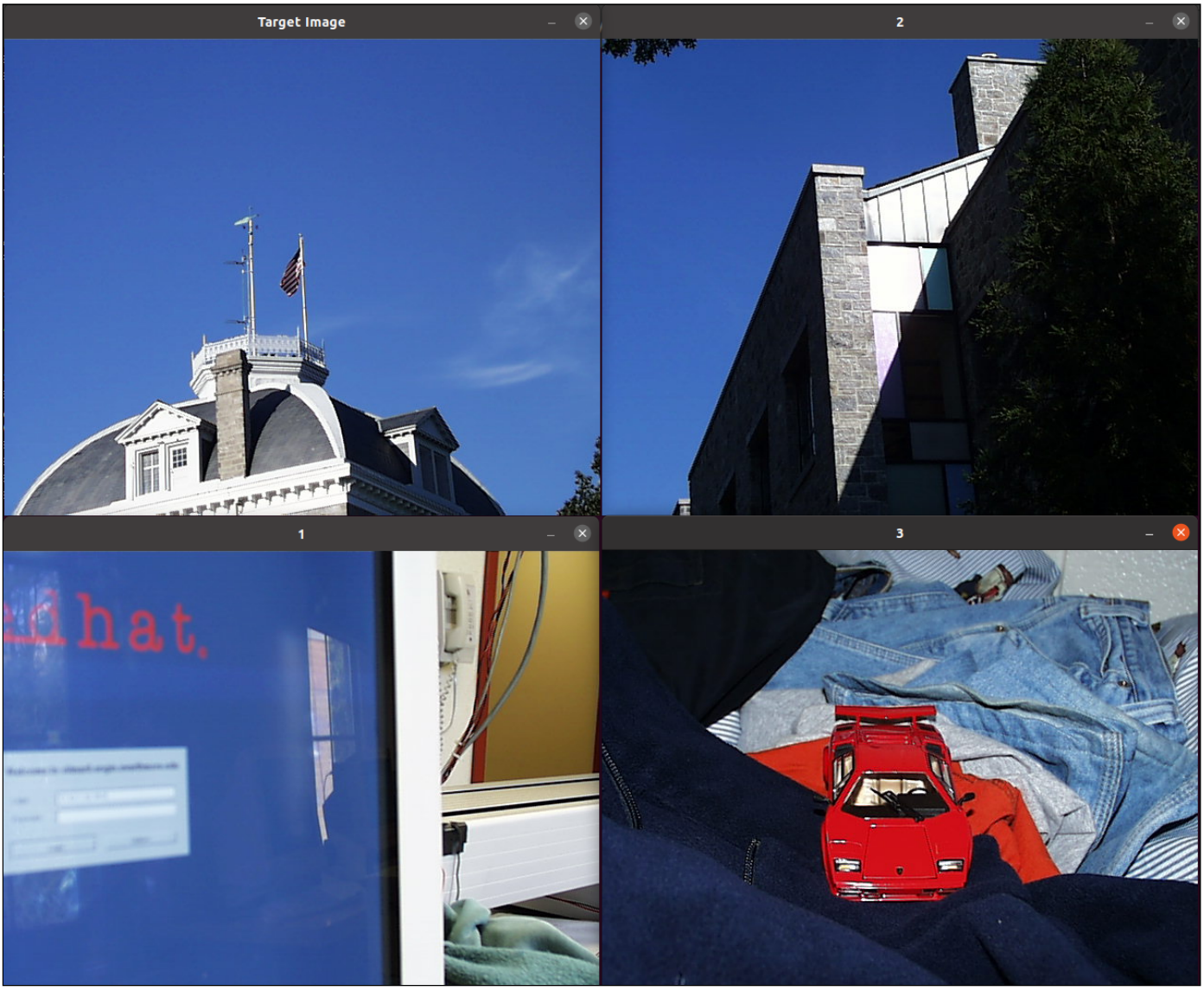
Task : 1



Task 2 :

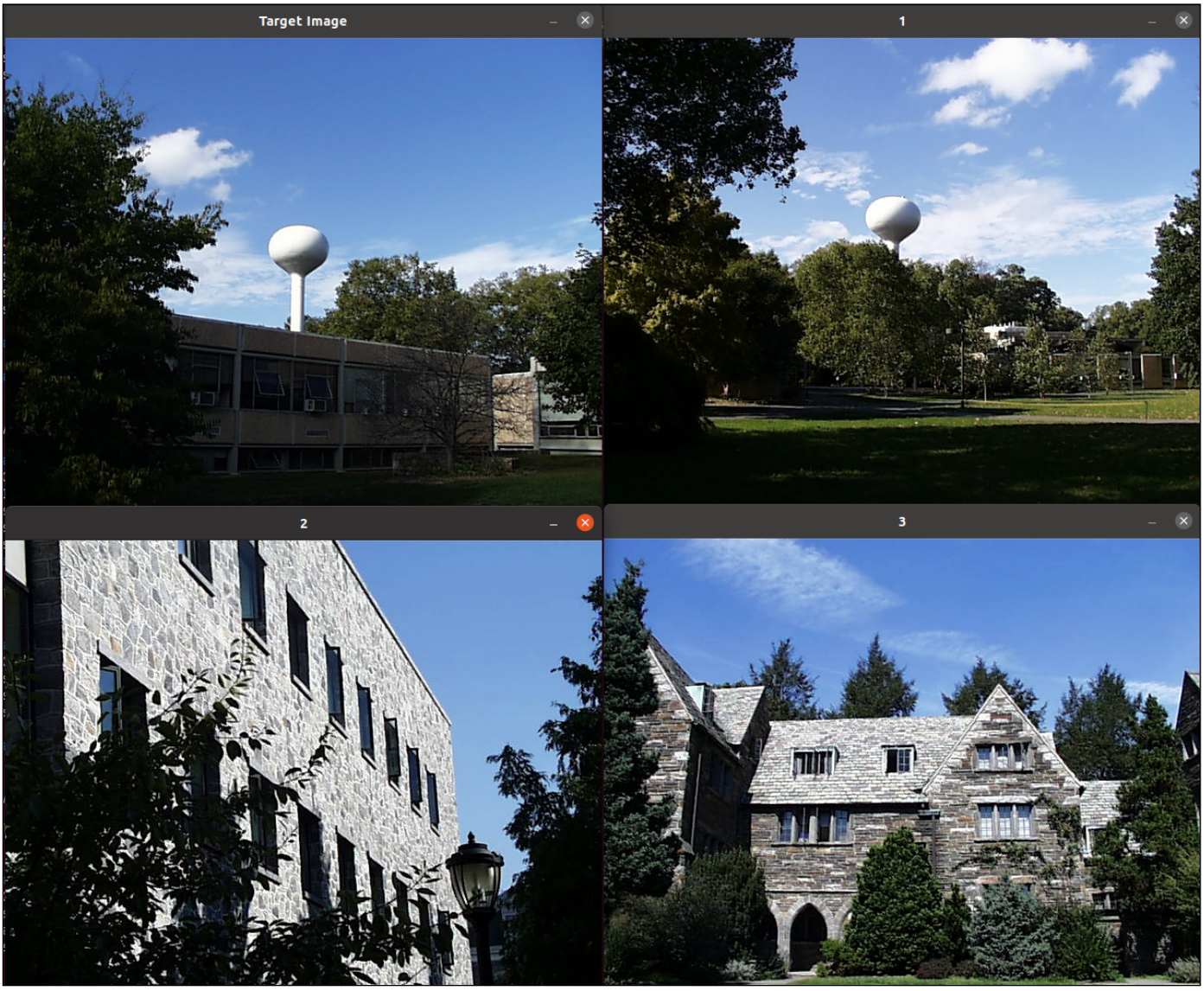
The results were obtained with a whole image rg chromaticity histogram using 16 bins for each r and g and distance metric is histogram intersection.


Target Image
1
2
3



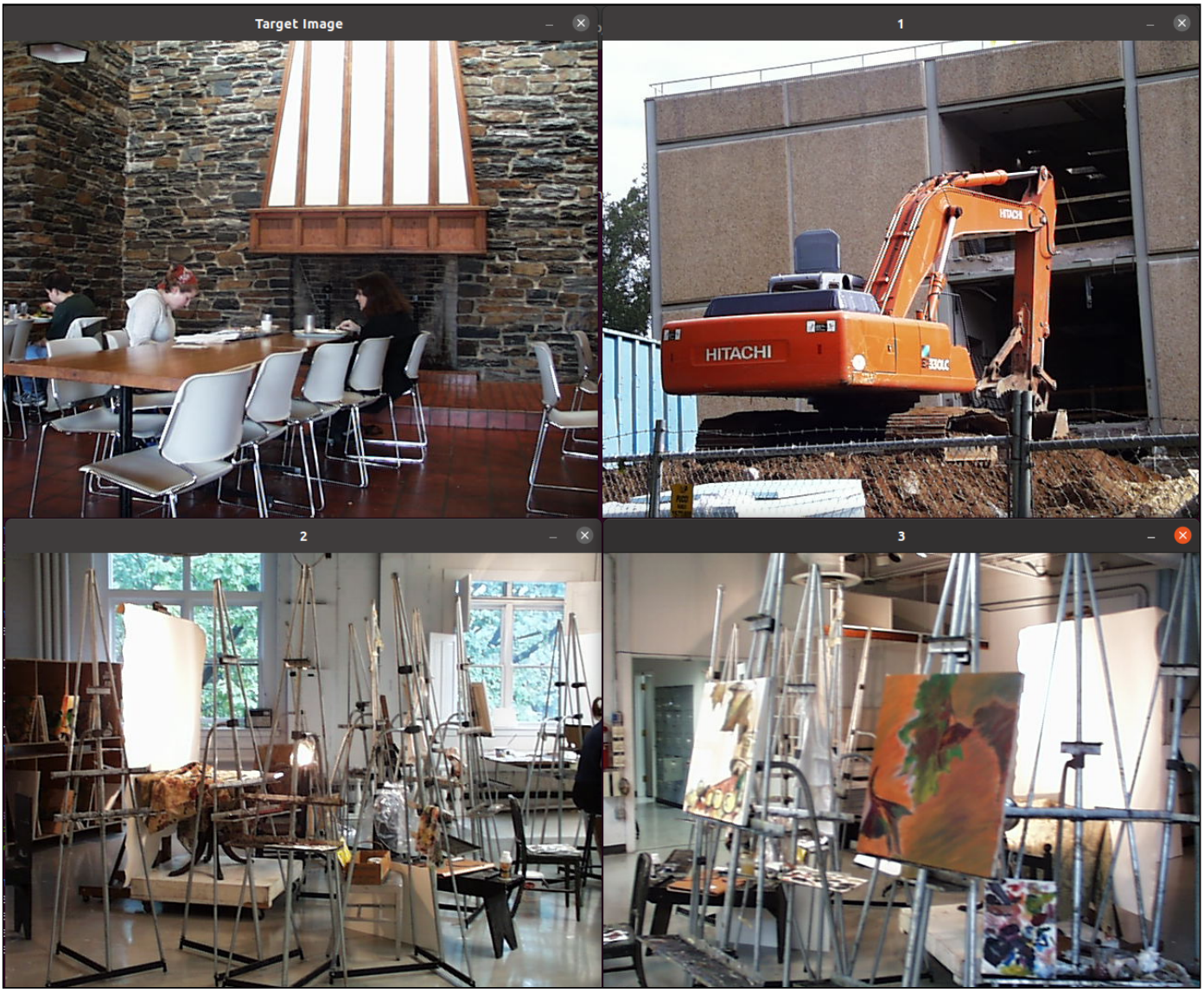
Task : 3

The results were obtained with a whole image RGB histogram using 8 bins and distance metric is histogram intersection.



Task : 4

The results were obtained with sobel magnitude filter which was implemented in project1 with the combination of a whole image rg chromaticity histogram using 16 bins for each r and g and distance metric is histogram intersection.



pic.0745.jpg pic.0004.jpg pic.0011.jpg

Task : 5 :

For this I have implemented Laws L5E5 filter with the combination of color information that is rg chromaticity histogram. To get better results, I have distributed weights as : 0.6 to rg-chromaticity and 0.4 to color information. I have not selected L5L5 combination as it is sensitive to mean brightness values and is not used. Top 10 matches for green garbage containers is better than garden with benches images.

Target 1 :

pic.0752.jpg pic.0755.jpg pic.0747.jpg pic.0750.jpg pic.0869.jpg pic.0753.jpg pic.1091.jpg pic.0749.jpg pic.0933.jpg pic.0141.jpg

Target 2 :

pic.0809.jpg pic.0701.jpg pic.1073.jpg pic.0974.jpg pic.0810.jpg pic.0922.jpg pic.0418.jpg pic.0330.jpg pic.0498.jpg pic.0806.jpg

Extension :

1.Gabor Filter - Implemented Gabor texture filter as my first extension, here, I have created 4 filters with variation in theta angle- 0, 90, 180 and 270 degrees. As you can see the results, all images with same textures are displayed below. I have used histogram intersection as distance

metric. To implement this task, I have first calculated 4 kernels and applied 32*32 kernals on each image. Here, 0th image is the target image.

2. Collect all blue trash bins, For this, I have again implemented Laws L5E5 filter with the combination of color information that is rg chromaticity histogram.

pic.0969.jpg pic.0289.jpg pic.0214.jpg pic.0665.jpg pic.0291.jpg

pic.0920.jpg pic.0288.jpg pic.0133.jpg pic.0765.jpg ic.0029.jpg

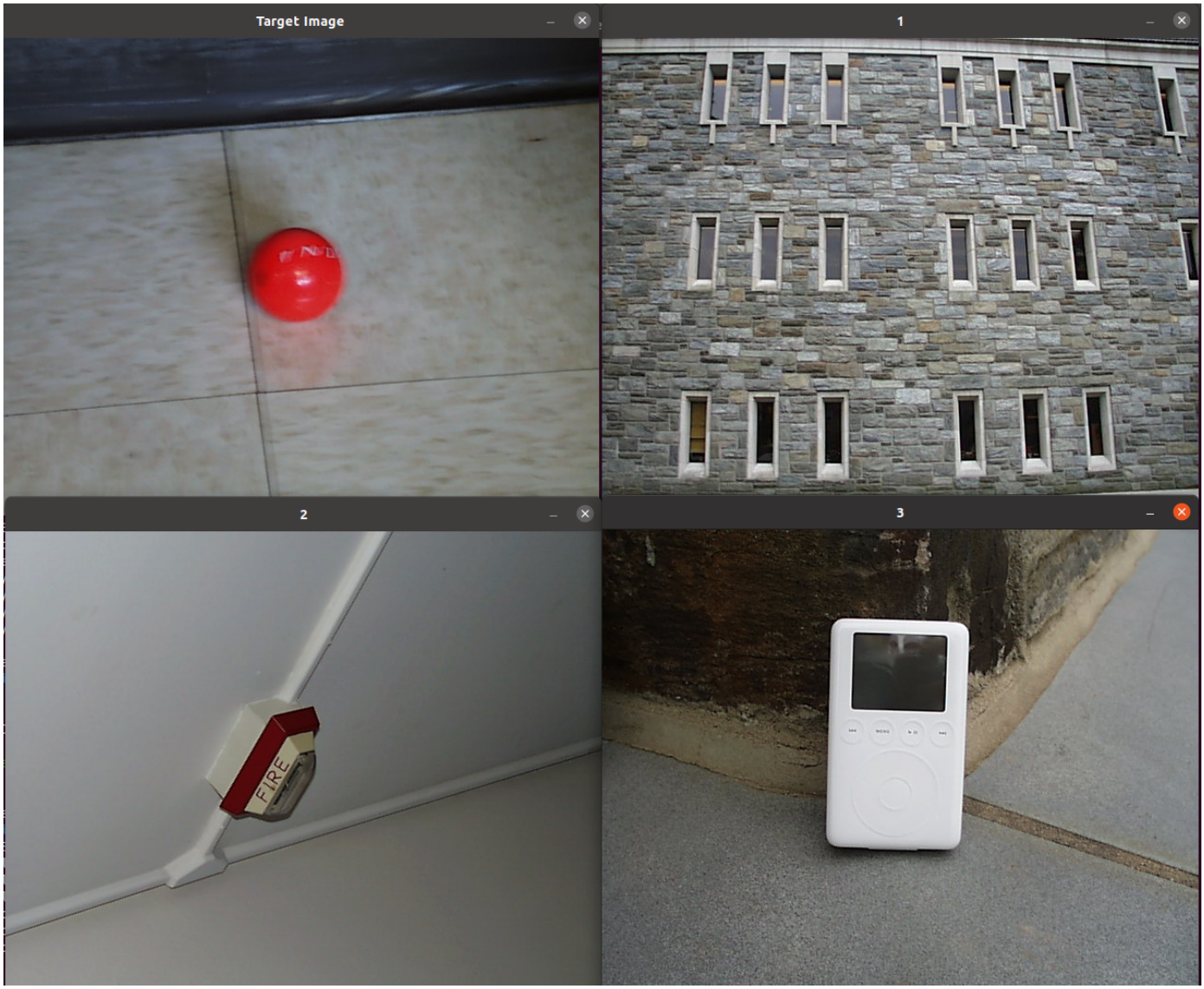
Extension 3 : Comparing 3 distance Metrics : 1. Histogram Intersection 2. Correlation 3. Chi-Sqaure

In this task, I have compared 3 distance metrics listed above on single image. After comparing the distances, chi-square is better distance metric compared to both intersection and correlation. I implemented this on HS 2D histogram. For the *Correlation* and *Intersection* methods, the higher

the metric, the more accurate the match. For chi-square, the less the result, the better the match.

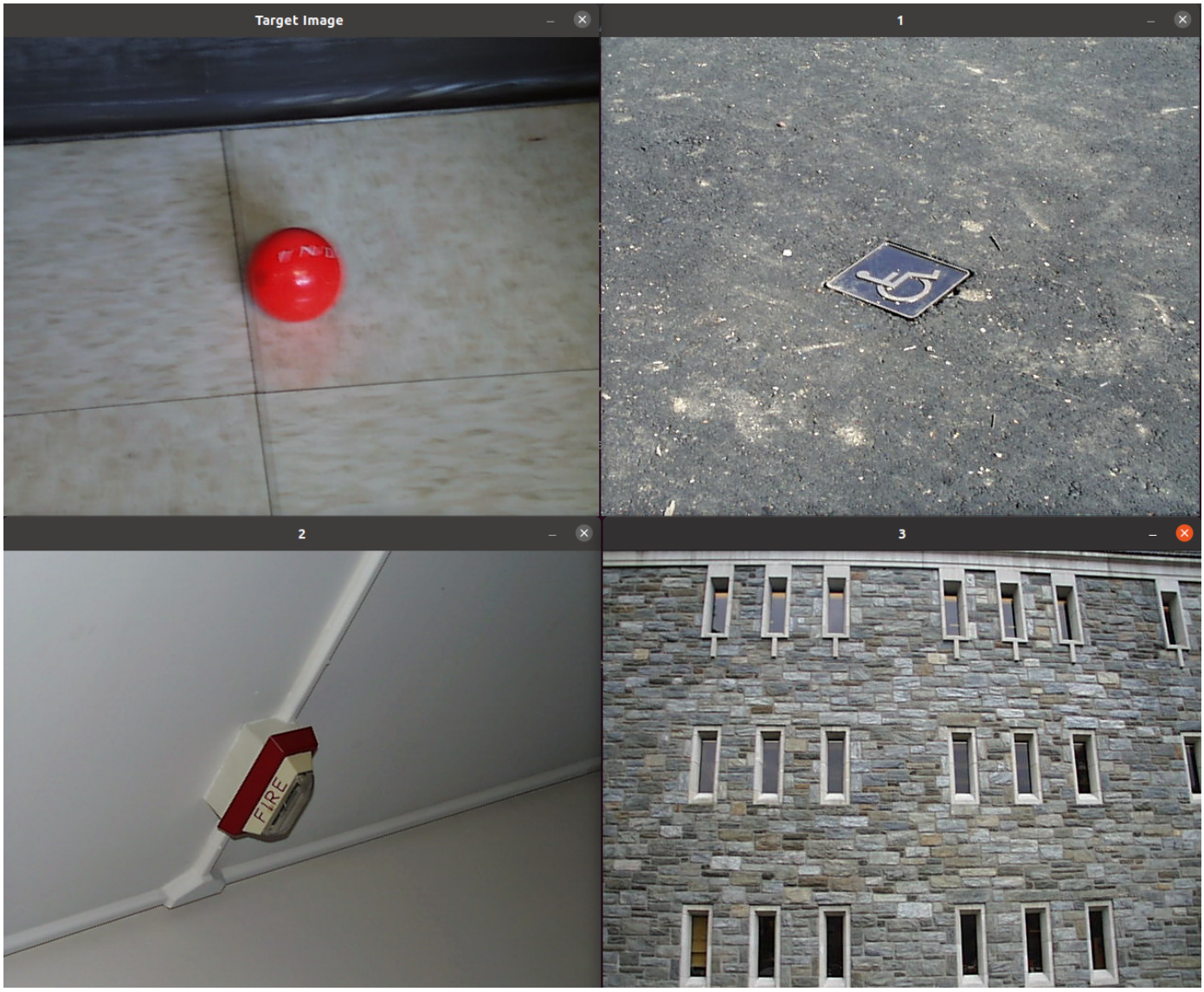
A. Chi-Square Result:

pic.1070.jpg pic.0440.jpg pic.1076.jpg



B. Correlation Result:

pic.0446.jpg pic.0440.jpg pic.1070.jpg



C. Histogram intersection Result:

pic.1070.jpg pic.0440.jpg pic.0799.jpg

Acknowledgments

https://en.wikipedia.org/wiki/Gabor_filter#:~:text=In%20image%20processing%2C%20a%20Gabor,point%20or%20region%20of%20analysis.
https://medium.com/@anuj_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac9
https://www.geeksforgeeks.org/sorting-vector-of-pairs-in-c-set-1-sort-by-first-and-second/

https://en.wikipedia.org/wiki/Structural_similarity
https://docs.opencv.org/3.4/d8/dc8/tutorial_histogram_comparison.html

Reflection

This project helped me to understand how to extract different features using histograms, texture analysis and also to calculate the metric distances using methods like intersection and sum of squared differences, chi-square, Correlation and also how to normalize the histograms. I have also considered to manage memory by using simple arrays and not many STL c++ functionalities. Also, how to handle cases like divide by zero and beyond matrix index situations in the code.

No labels