Project 2: Content-based Image Retrieval

Name: Dev Vaibhav

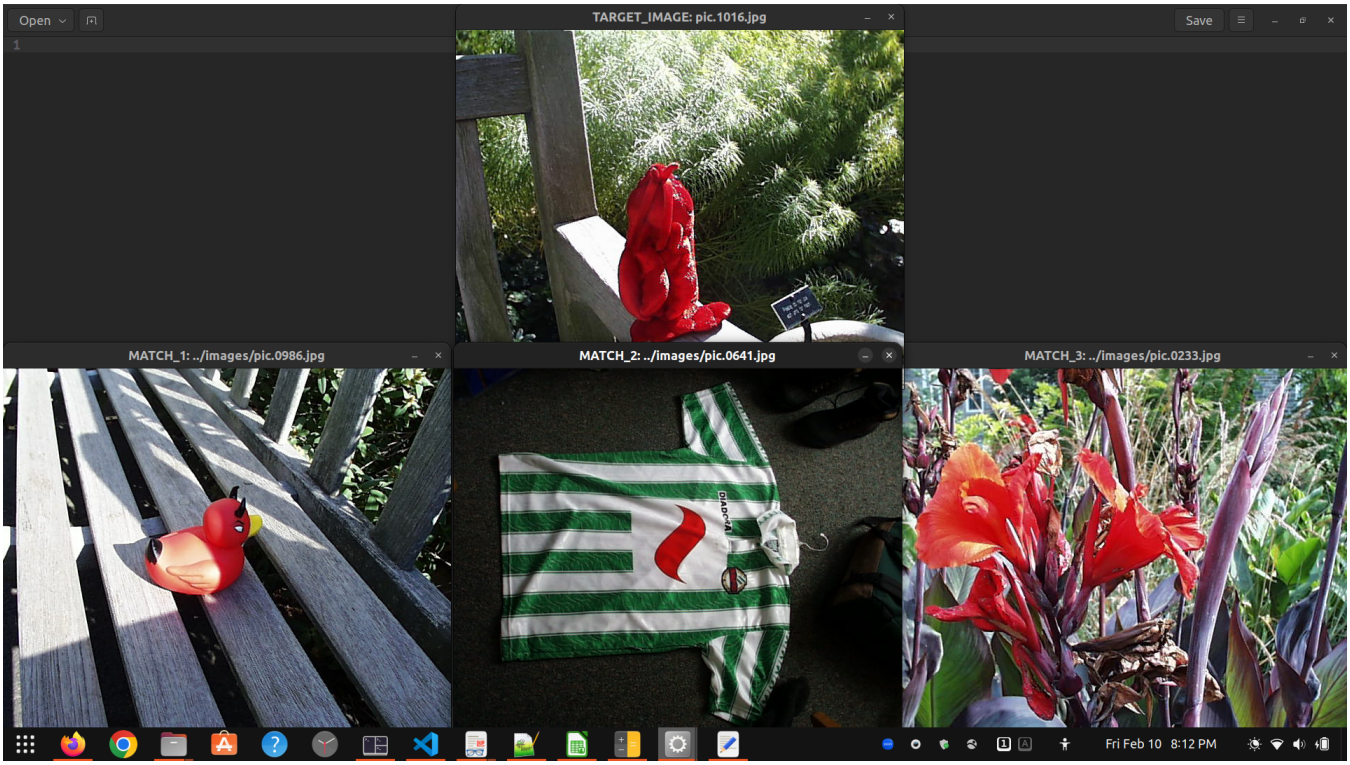
Email: vaibhav.d@northeastern.edu

Class: CS 5330 Pattern Recognition and Computer Vision

Description: The main objective of this project was to gain a deeper understanding of pattern recognition, and subsequently content-based image retrieval. I achieved this by implementing baseline matching, RGB, rg chromaticity histograms, multi-histograms, and Sobel magnitude texture, and experimenting with a few other methods and their combinations like Laws texture (L5E5), and Gabor filters for texture extraction. I tried to compare different distance metrics viz. sum of squared difference, histogram intersection, Chi-square, and Correlation distances. A set of training images were also chosen to test the various mix and match of feature calculation and matching metrics.

Task 1) Baseline Matching

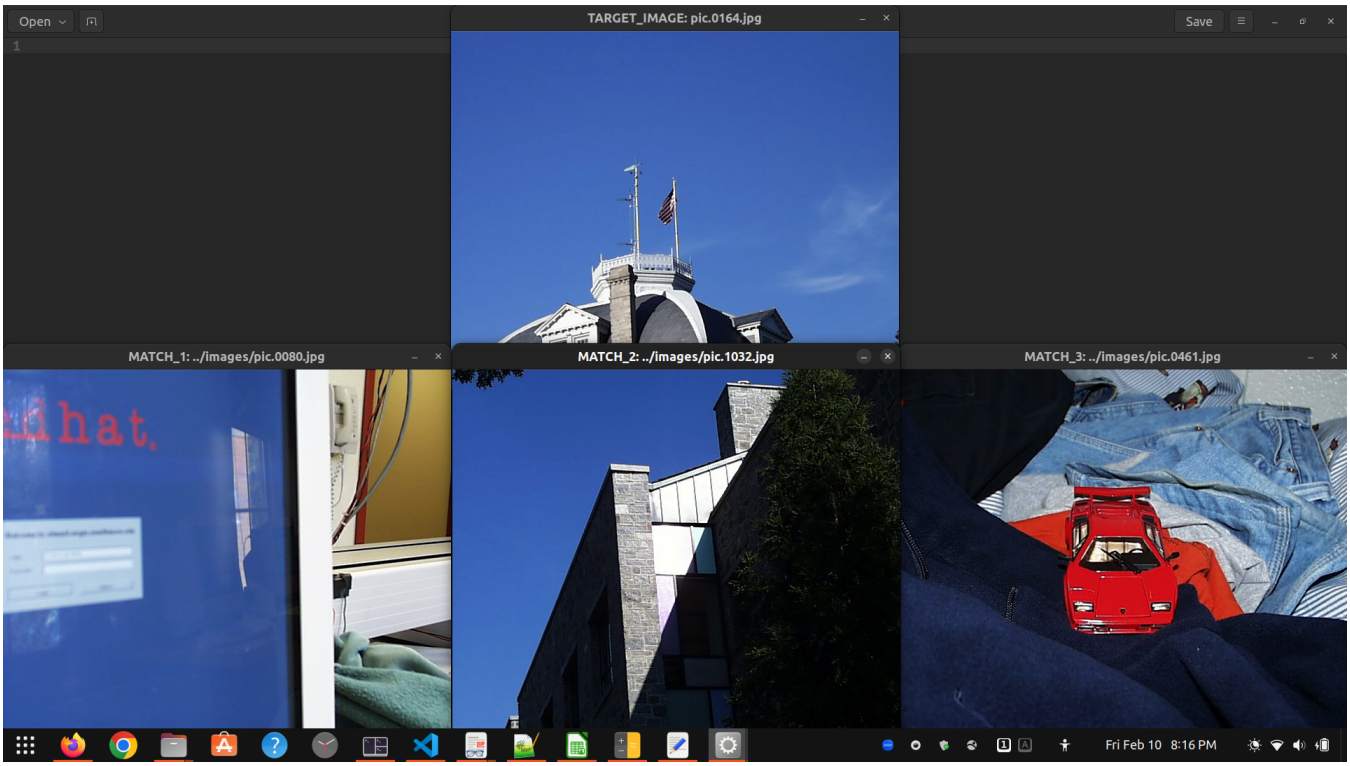
```
9.72303e+06 , 1007
The 3 closest matches to pic.1016.jpg are
../images/pic.0986.jpg | Distance: 28731.000000
../images/pic.0641.jpg | Distance: 41959.000000
../images/pic.0233.jpg | Distance: 86092.000000
```



Task 2) Histogram Matching

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

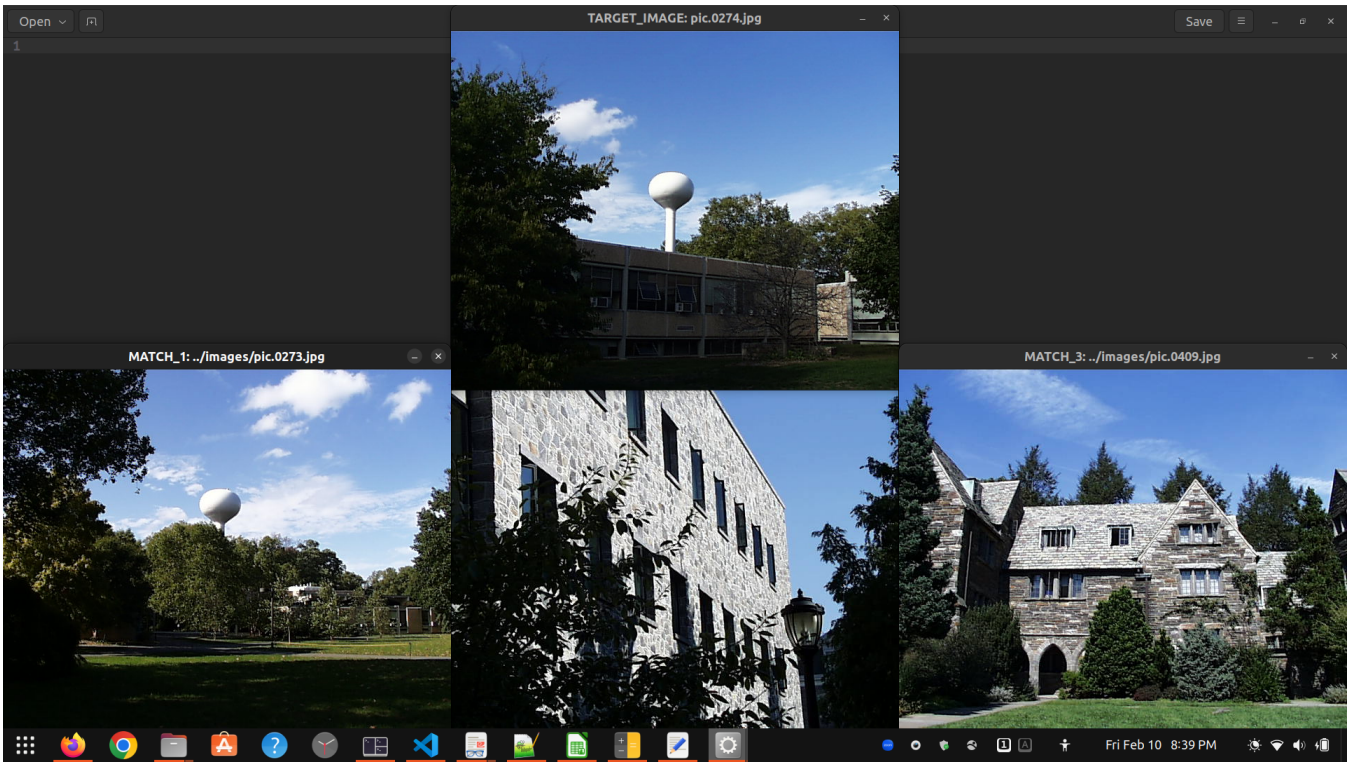
```
The 3 closest matches to pic.0164.jpg are
../images/pic.0080.jpg | Distance: 0.309700
../images/pic.1032.jpg | Distance: 0.374100
../images/pic.0461.jpg | Distance: 0.440100
```



Task 3) Multi-histogram Matching:

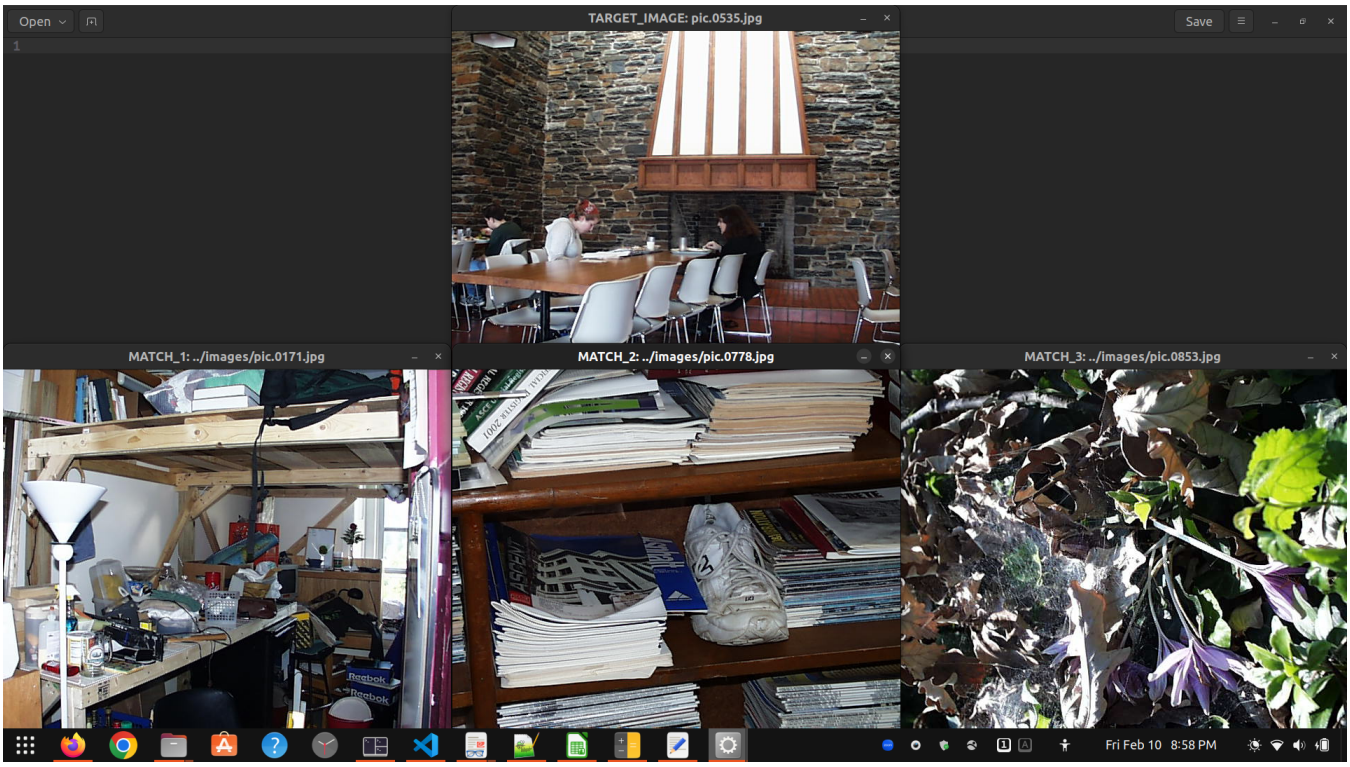
The results were obtained with two 3D RGB histograms by dividing the image into top and bottom halves using 8 bins and distance metric being weighted histogram intersection. The weights of both histograms are set to equal i.e. 0.5

```
The 3 closest matches to pic.0274.jpg are
../images/pic.0273.jpg | Distance: 0.347450
../images/pic.1031.jpg | Distance: 0.375300
../images/pic.0409.jpg | Distance: 0.379700
```

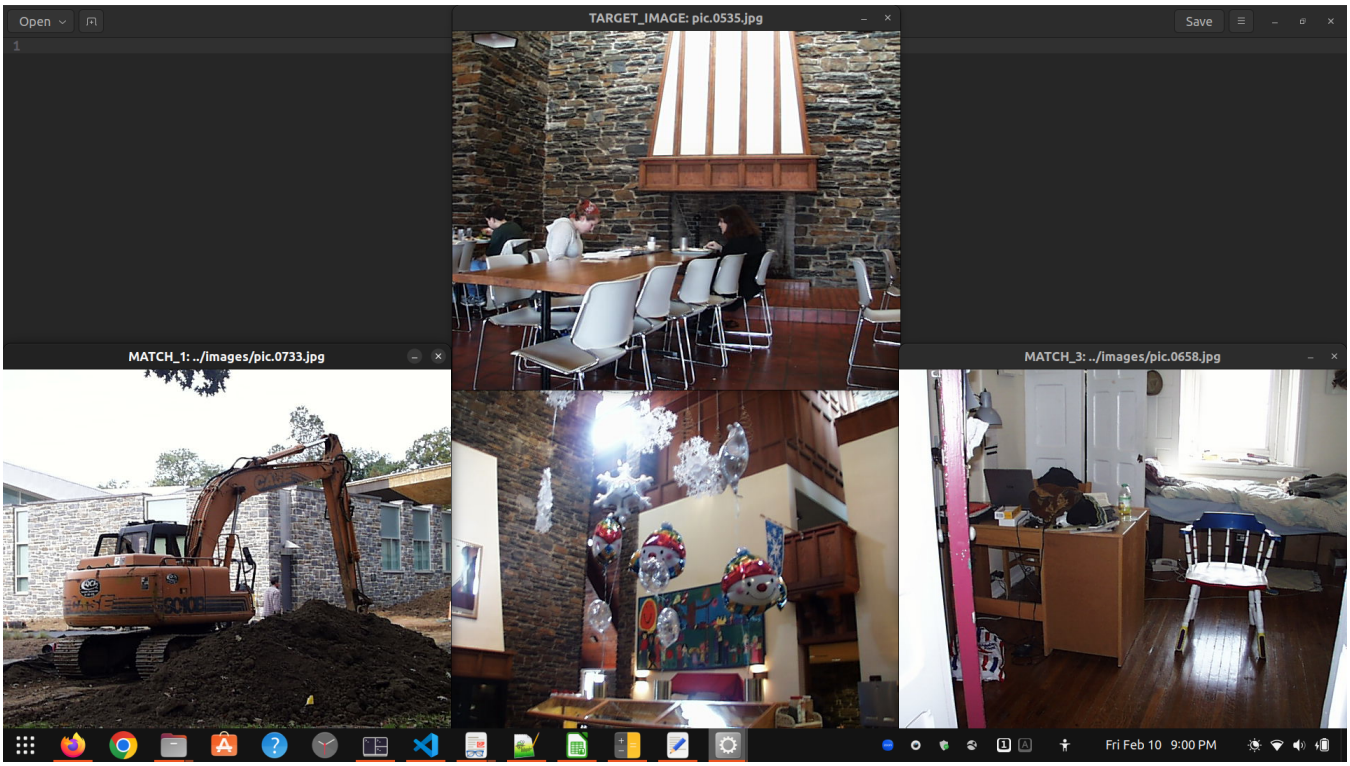


Task 4) Texture and Color:

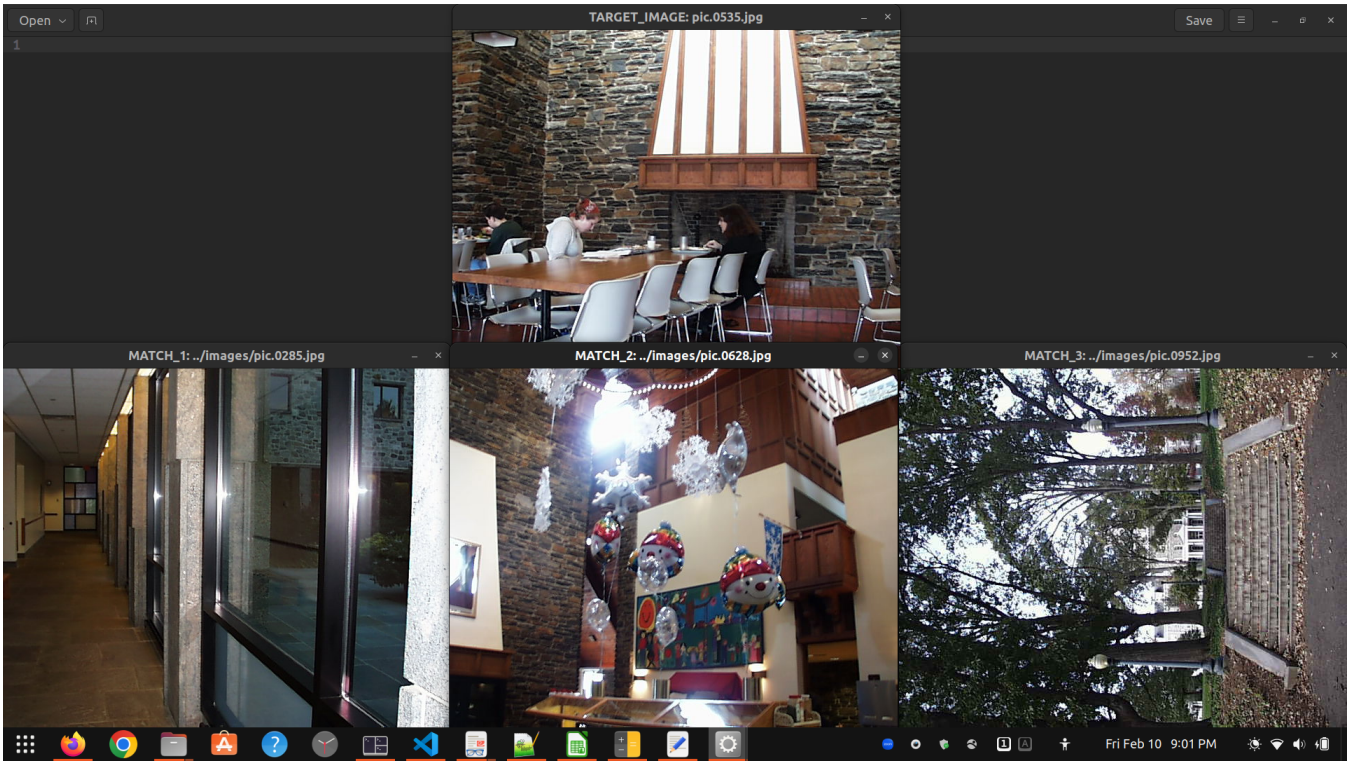
Calculated Sobel Gradient Magnitude 3D (implemented in Project 1) and whole image 3D RGB color histograms using 8 bins for each axis. Weight for the gradient magnitude image histogram is 0.6 whereas for original image's color histogram is 0.4



pic.0535.jpg results for Task 2 (Histogram matching)



pic.0535.jpg results for Task 3 (Multi Histogram matching)

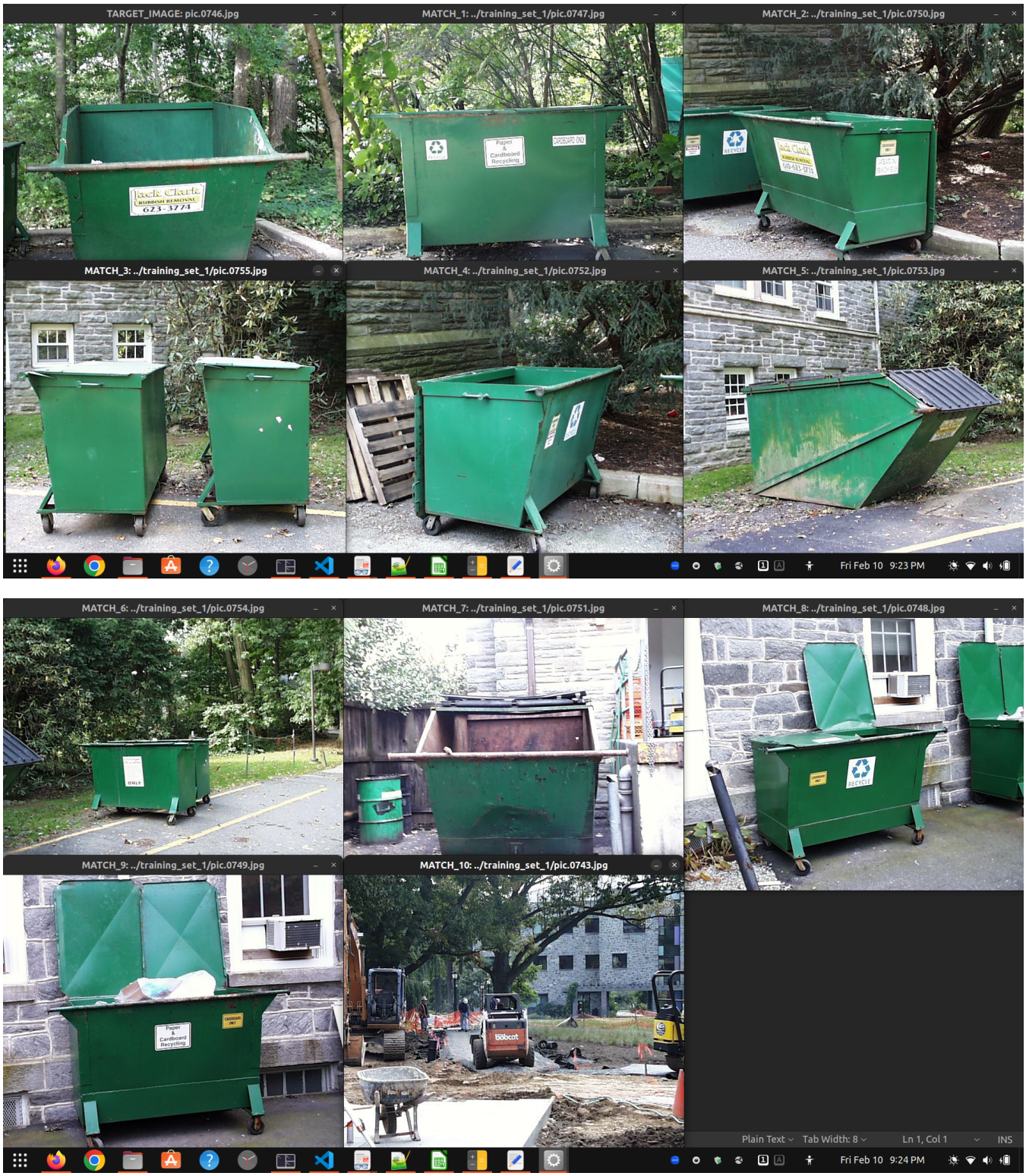


Task 5) Custom Design:

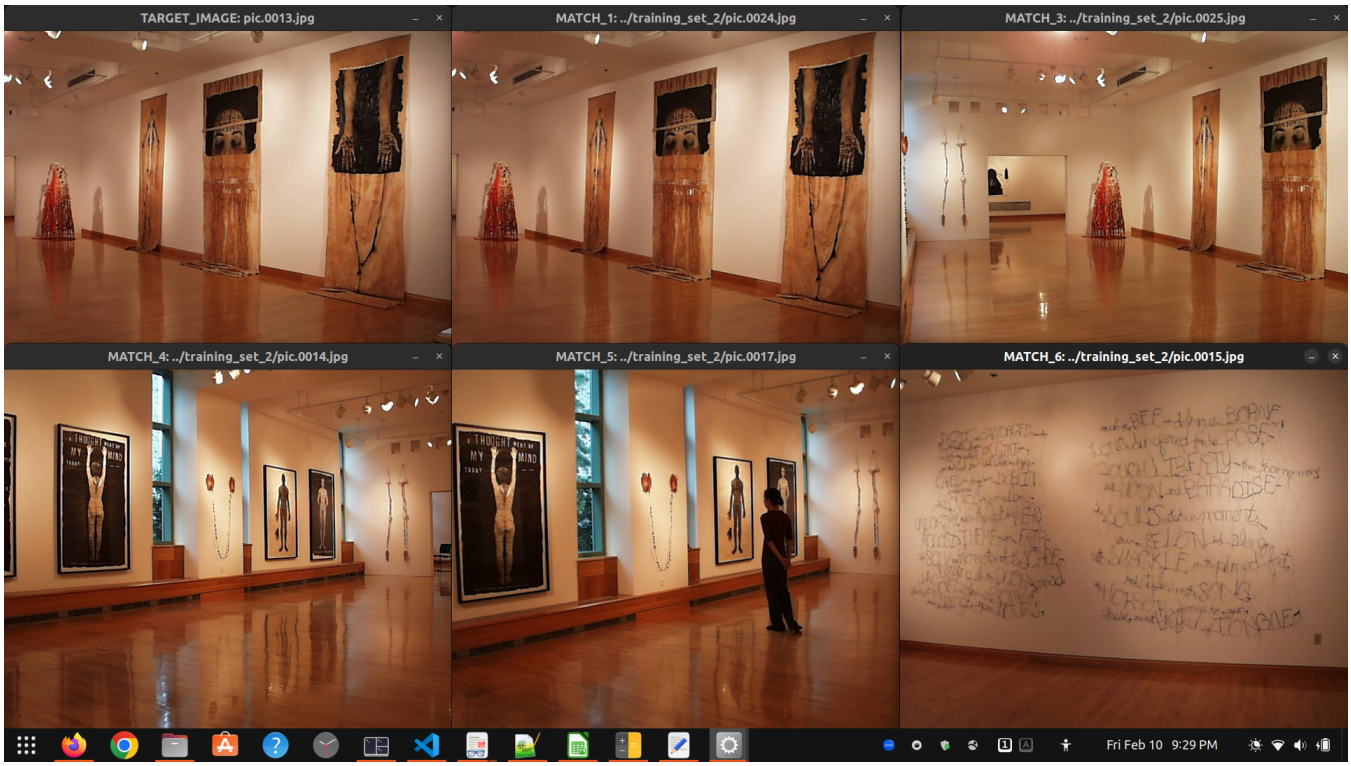
I have computed rg chromaticity using Laws L5E5 filter (weight = 0.6) output and 3D RGB histogram (weight = 0.4) of the original image. Top 10 results are shown in this case.

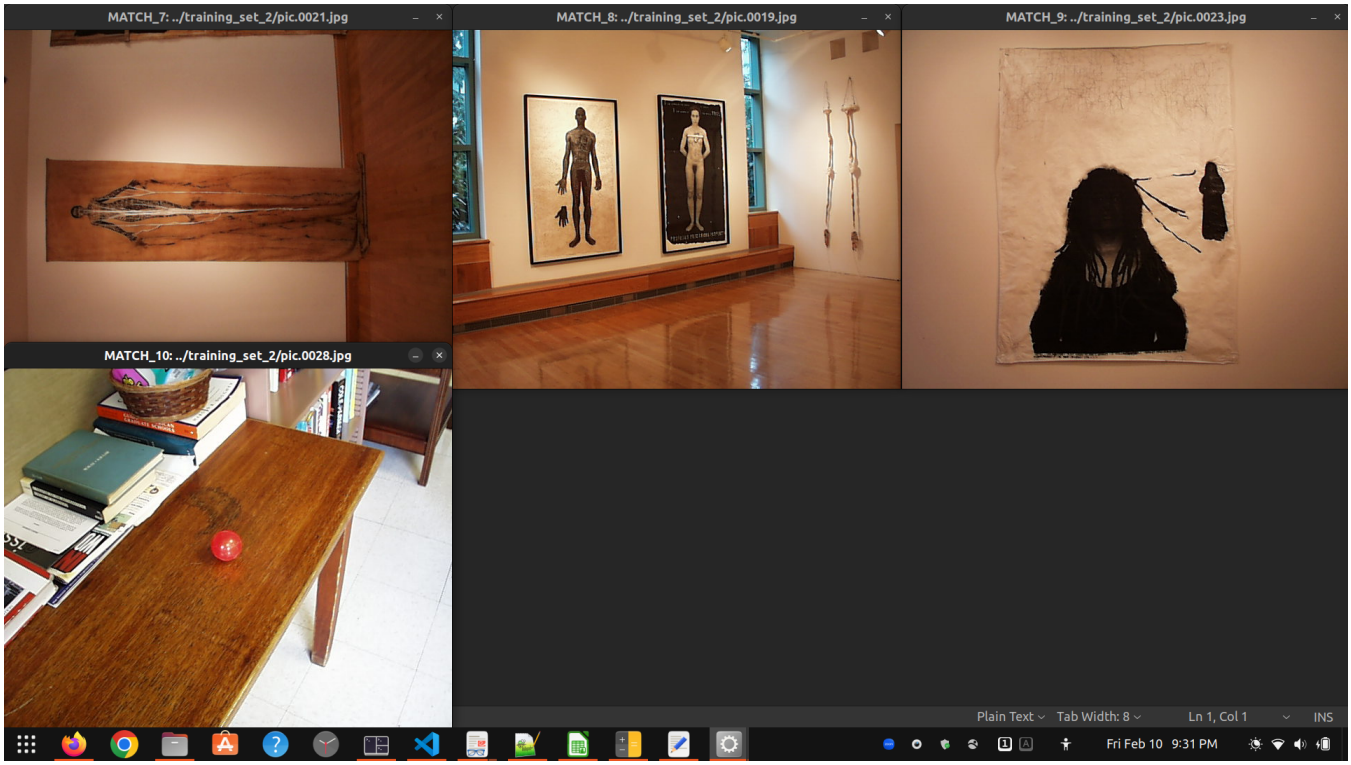
Two training sets which are subsets of the *olympus* database are chosen to calculate the feature vector and distance metric for this task.

Training Set 1 consists of 31 images which contains images of a green dustbin along with some images of a shoe and construction site. The algorithm is successfully able to extract most of the images which are similar to that of a green dustbin.



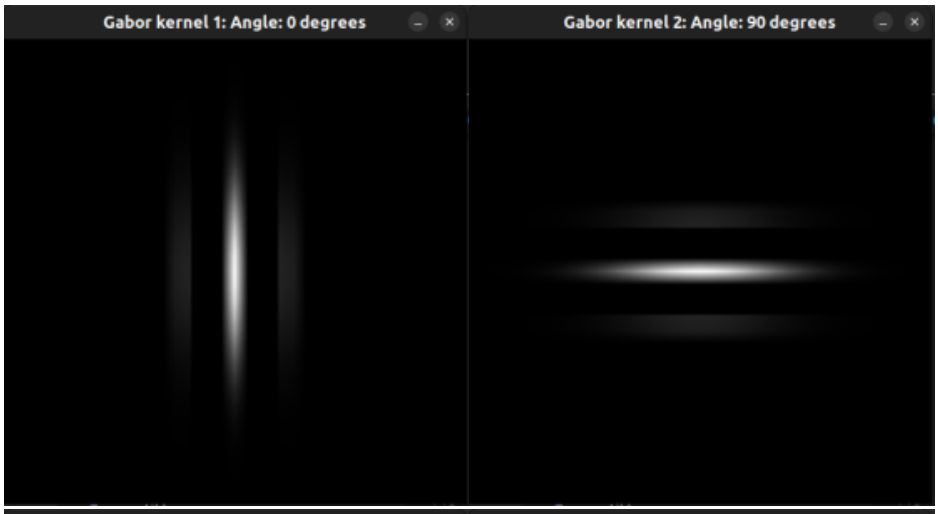
Training Set 2 consists of 28 images which contains images of museum, ball, pen, room setting. The algorithm is successfully able to extract most of the images which are similar to that of a museum.

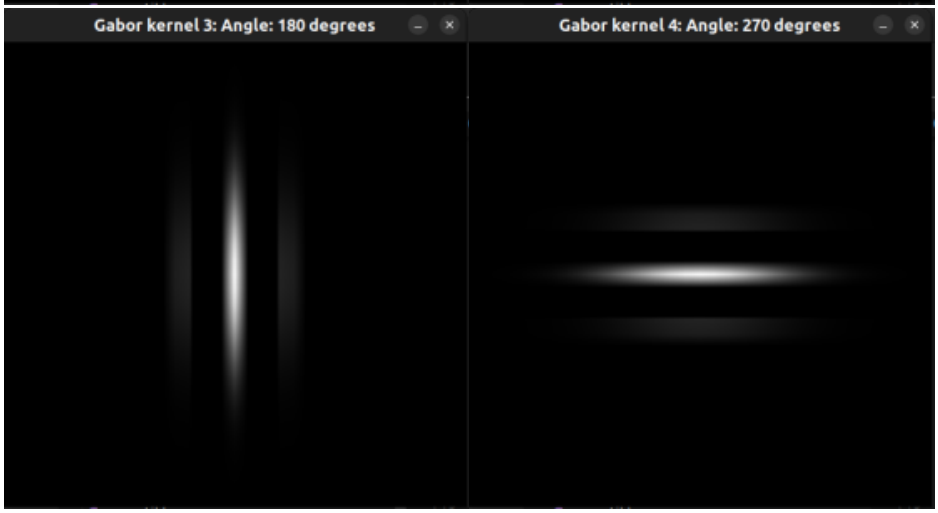


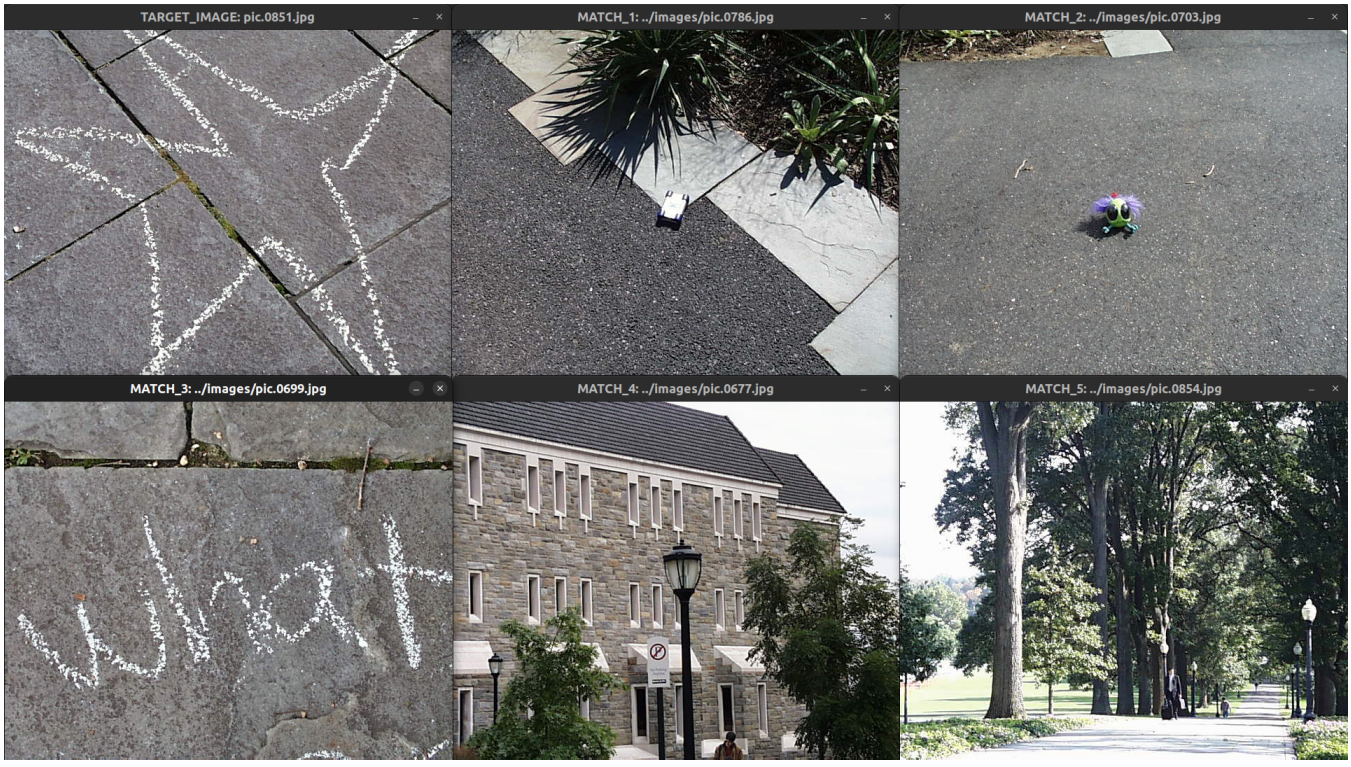


EXTENSION 1) Gabor filter: I implemented four Gabor filters (theta = 0, 90, 180, and 270 degrees) and made rg chromaticity histograms for each Gabor filter's output. I used the histogram intersection metric with a weight of 0.25 to each to find the closest five results against a target image. Gabor filters are selected as an extension as they represent the human visual cortex. As can be seen that the algorithm gives good results. However. the filter requires a bunch of parameters that might not be applicable to all of the images in the dataset. Moreover, it required a lot of tuning to get the acceptable results.

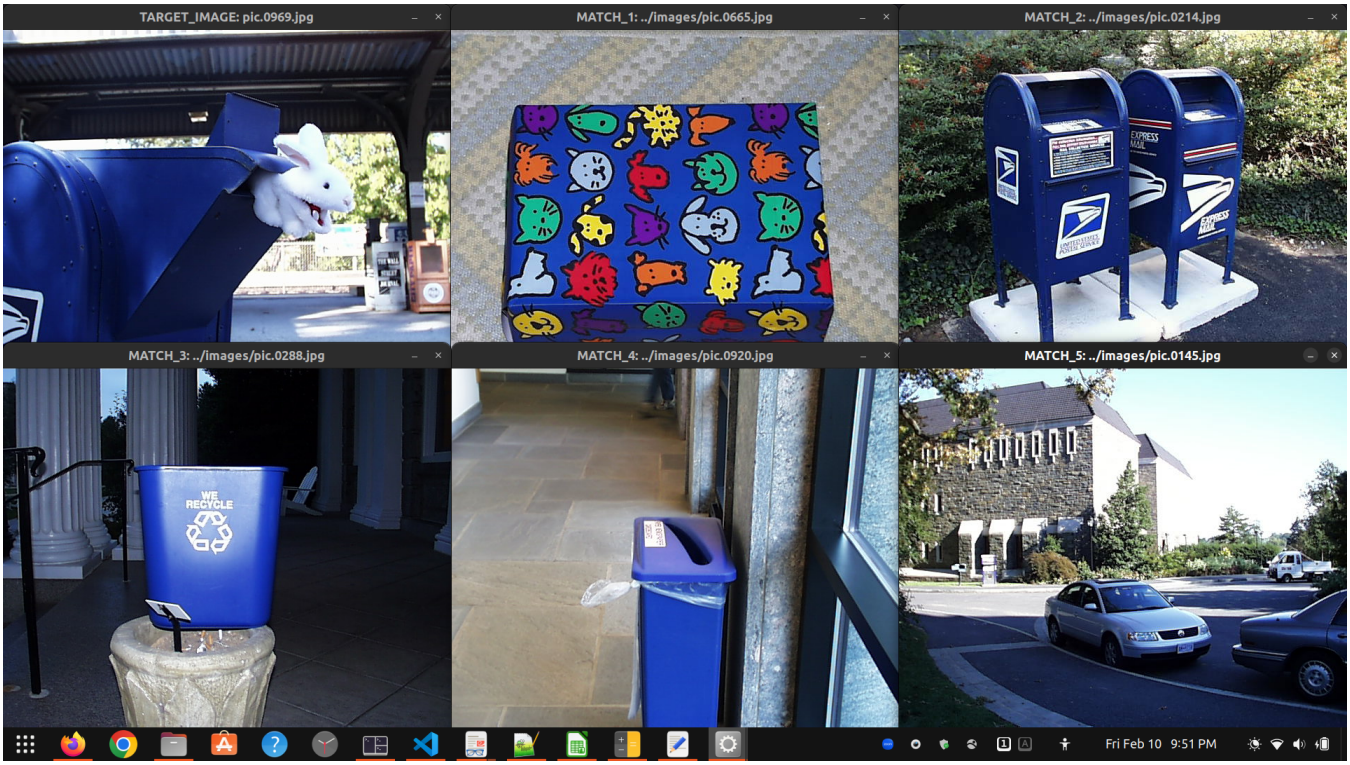
Gabor filter kernels (resized):

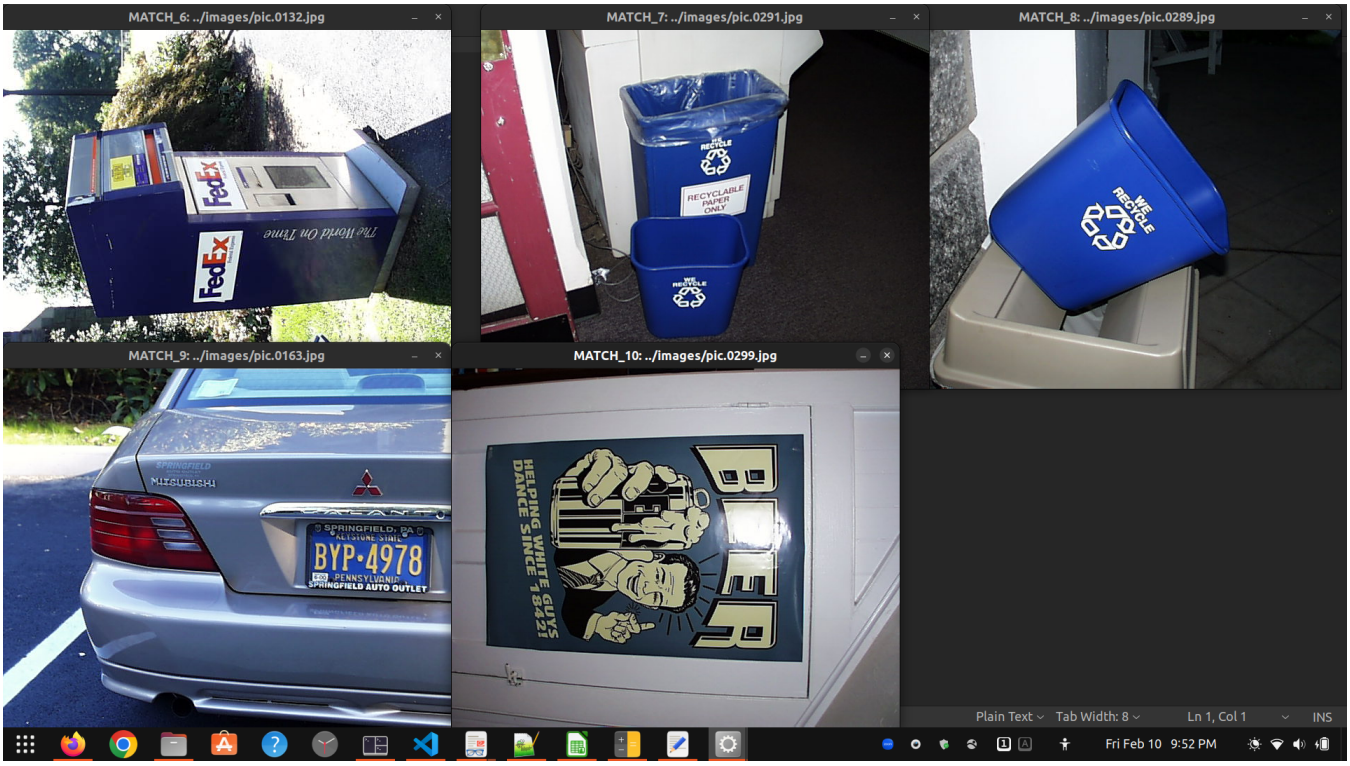

Gabor kernel 1: Angle: 0 degrees
Gabor kernel 2: Angle: 90 degrees


Gabor kernel 3: Angle: 180 degrees
Gabor kernel 4: Angle: 270 degrees


TARGET_IMAGE: pic.0851.jpg
MATCH_1: ../images/pic.0786.jpg
MATCH_2: ../images/pic.0703.jpg
MATCH_3: ../images/pic.0699.jpg
MATCH_4: ../images/pic.0677.jpg
MATCH_5: ../images/pic.0854.jpg

EXTENSION 2) Collect all blue trash bins: I have implemented 2D rg histogram of the Laws L5E5 filter along with + 2D rg chromaticity histogram for the original image. As can be seen, the algorithm is able to give good enough results.





EXTENSION 3) Comparing 4 distance metrics: Correlation, Chi-Square, Histogram Intersection, and Bhattacharyya distance.

Distance is calculated as per the documentation provided by OpenCV. After comparison, Chi-Square and Bhattacharya distance seems to be good distance metric for the selected target image.

1. **Correlation (CV_COMP_CORREL)**

$$d(H_1,H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I(H_1(I) - \bar{H_1})^2 \sum_I(H_2(I) - \bar{H_2})^2}}$$

where

$$\bar{H_k} = \frac{1}{N}\sum_J H_k(J)$$

and N is the total number of histogram bins.

2. **Chi-Square (CV_COMP_CHISQR)**

$$d(H_1,H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)}$$

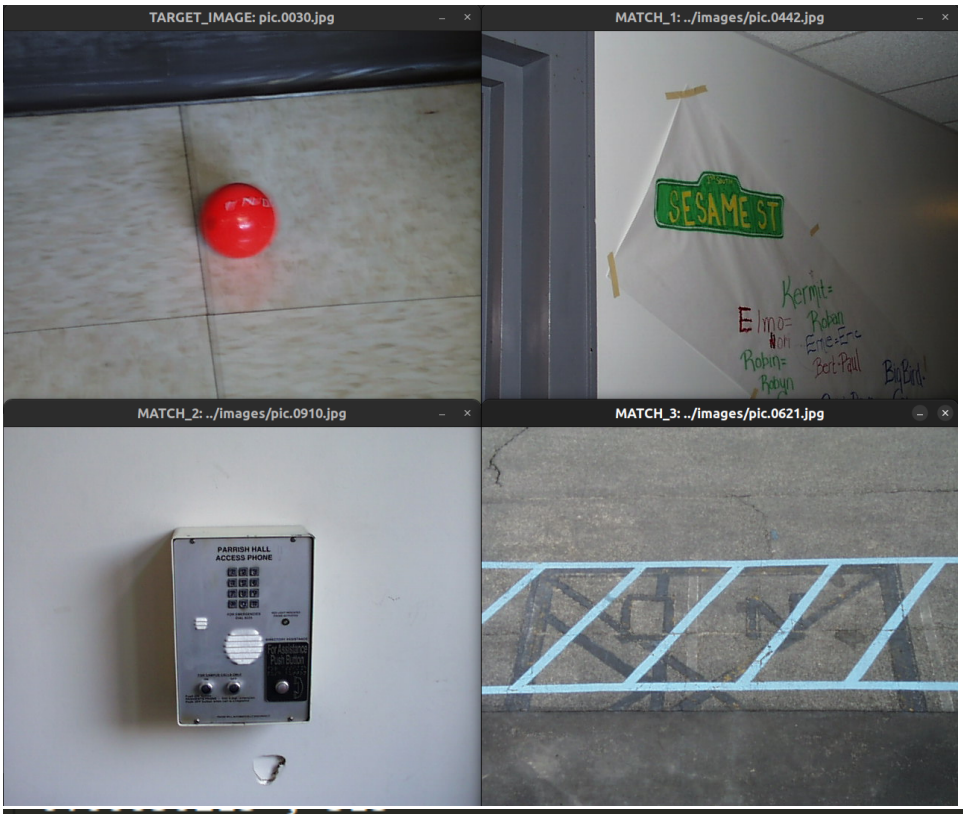
3. **Intersection (method=CV_COMP_INTERSECT)**

$$d(H_1,H_2) = \sum_I \min(H_1(I), H_2(I))$$

4. **Bhattacharyya distance (CV_COMP_BHATTACHARYYA)**

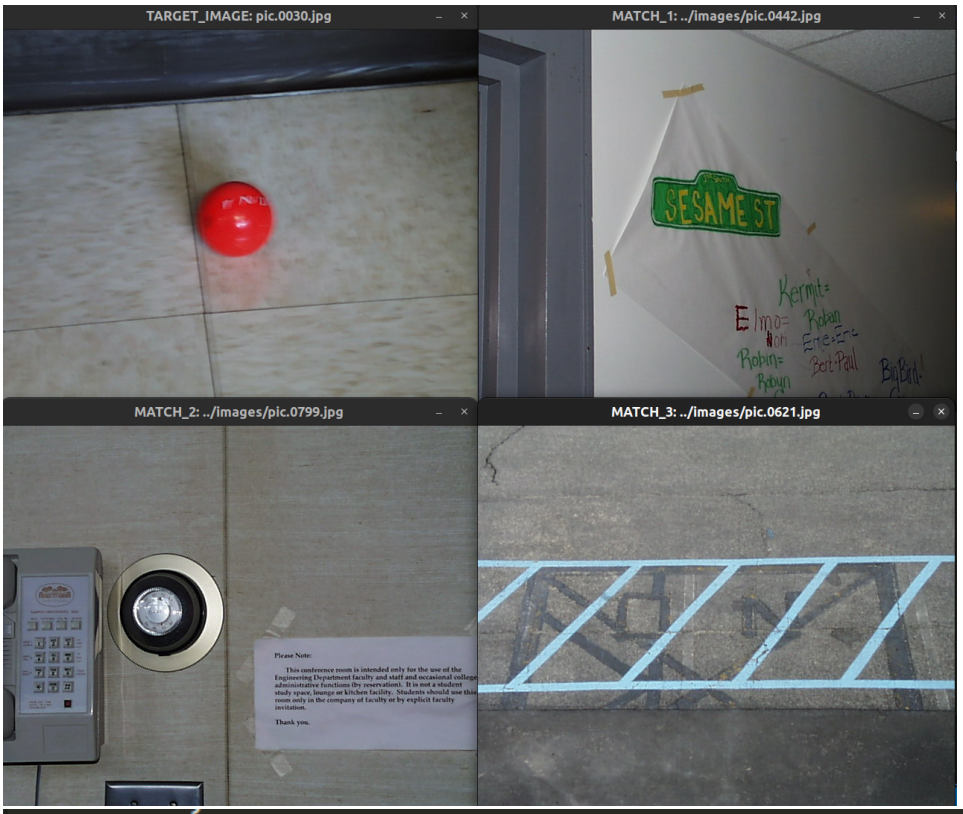
$$d(H_1,H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H_1}\bar{H_2}N^2}}\sum_I \sqrt{H_1(I) \cdot H_2(I)}}$$

A. Correlation metric result



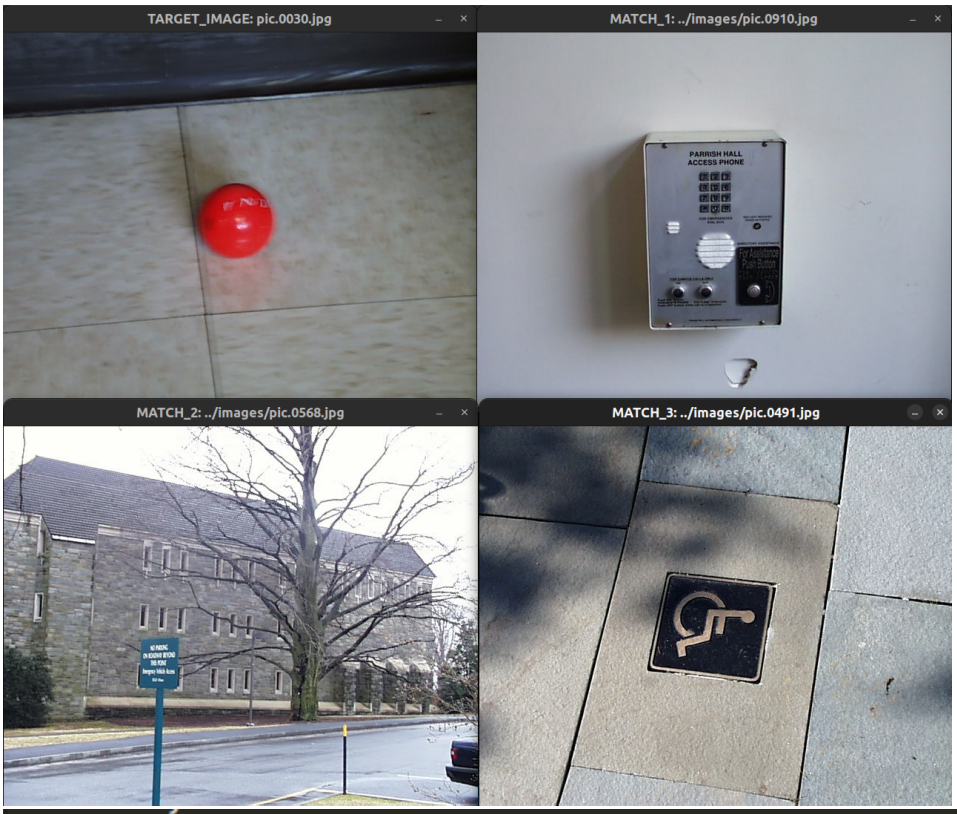

```
The 3 closest matches to pic.0030.jpg are
../images/pic.0442.jpg | Distance: 0.999553
../images/pic.0910.jpg | Distance: 0.999481
../images/pic.0621.jpg | Distance: 0.999433
```

B. Chi-Square metric result



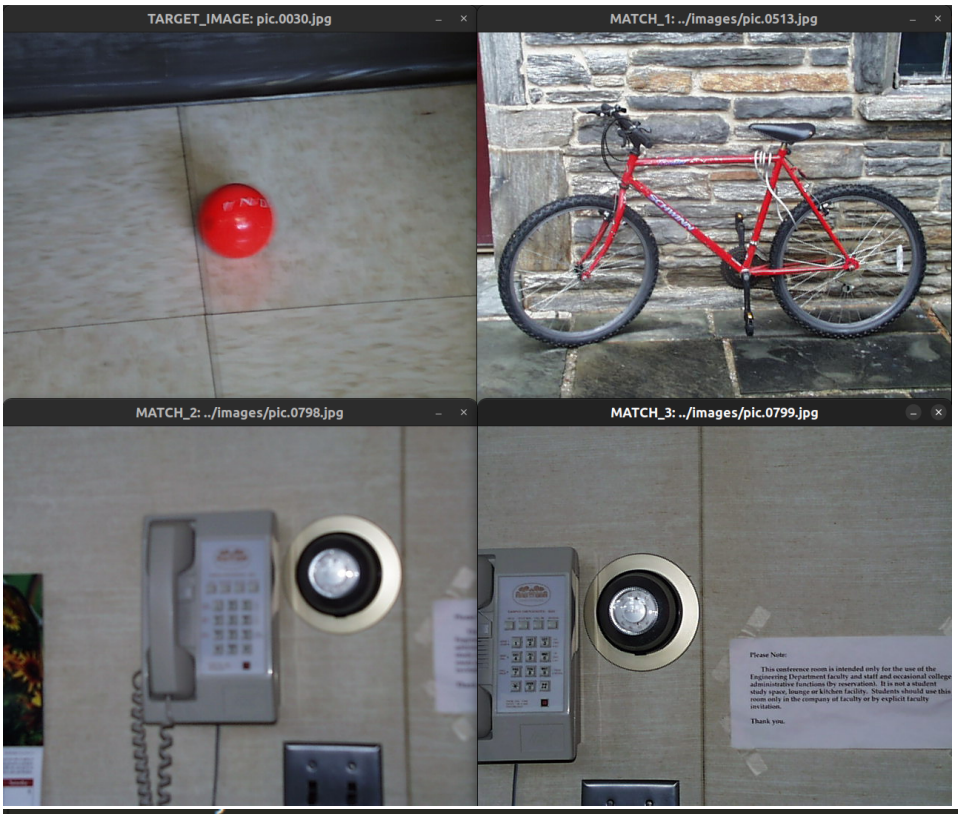

```
The 3 closest matches to pic.0030.jpg are
../images/pic.0442.jpg | Distance: 0.057290
../images/pic.0799.jpg | Distance: 0.065368
../images/pic.0621.jpg | Distance: 0.075554
```

C. Intersection metric result




```
The 3 closest matches to pic.0030.jpg are
../images/pic.0910.jpg | Distance: 0.047600
../images/pic.0568.jpg | Distance: 0.051800
../images/pic.0491.jpg | Distance: 0.053200
```

D. Bhattacharyya distance




```
The 3 closest matches to pic.0030.jpg are
../images/pic.0513.jpg | Distance: 0.128315
../images/pic.0798.jpg | Distance: 0.141709
../images/pic.0799.jpg | Distance: 0.148595
```

REFLECTION

This project gave me a good understanding of the various factors that have to be taken into consideration for pattern matching and image retrieval. It gave me a good idea about how various combinations of color and texture feature vectors can correspond to developing accurate matching algorithms. I faced some difficulty in understanding the 2d rg chromaticity histogram initially in the lectures but when I tried to implement them, I found them fun and easy. However, for Gabor and Laws filters, the concepts were easy to understand but a little complicated to implement.

Additionally, the custom design task, or rather the freedom of choice in the task, left me in a bind and I went down various paths that did not seem to give the desired results. Nonetheless, it was very interesting and a great learning experience to explore all the different ways in which color and texture could be accounted for while developing feature vectors. Building upon the things we learned in the last project such as pixel-wise manipulation, various data types, and memory allocation was a good way to solidify these basic concepts. I learned to handle edge cases like dividing by zero and debugging C++ code.

Acknowledgments:

https://stackoverflow.com/questions/7571326/why-does-dividing-two-int-not-yield-the-right-value-when-assigned-to-double

https://stackoverflow.com/questions/26948110/gabor-kernel-parameters-in-opencv

https://cvtuts.wordpress.com/2014/04/27/gabor-filters-a-practical-overview/

https://www.geeksforgeeks.org/strstr-in-ccpp/

https://www.geeksforgeeks.org/keep-track-of-previous-indexes-after-sorting-a-vector-in-c-stl/

https://docs.opencv.org/3.4/d8/dc8/tutorial_histogram_comparison.html