Project 2 Report

Name: Xujia Qin

Description:

The main goal of this project is to analyze pattern and color matching at the pixel level in images. To compare images, we utilized various feature vectors, including direct pixel values, 2D and 3D histograms incorporating spatial information, combinations of color and texture, and specific image patterns. We explored multiple distance metrics—such as sum of squared differences, L1 distance, histogram intersection, and thresholding—to measure dissimilarities between images based on their histograms. By computing differences between histograms derived from feature vectors, we identify the images most like a given target.

The project is structured into two programs:

1. The first program processes all images, computes feature vectors and stores them in separate CSV files based on feature type for later use.
2. The second program takes a specified feature type, compares the target feature vector against those stored in the corresponding CSV file, and retrieves the top n matching results.

Extensions:

3. Added another feature vector method Laplacian and chi-square distance metric.
4. Integrated with OpenCV DNN API to get read a resNet18 network and get an image embedding for the new matching results.
5. Attempted object recognition for bananas with yellow color recognition.

Required result 1: show the top three matches for the target image pic.1016.jpg.

```
cathyqindembp:Project 2. Content-based Image Retrieval cathyqin$ ./topN square olympus/pic.1016.jpg 5
Reading /Users/cathyqin/Desktop/Pattern_Recognition_Computer_Vision/Project 2. Content-based Image Retrieval/featureVectors.csv
Finished reading CSV file
olympus/pic.1016.jpg:0
olympus/pic.0986.jpg:14049
olympus/pic.0641.jpg:21756
olympus/pic.0547.jpg:49703
olympus/pic.1013.jpg:51539
```

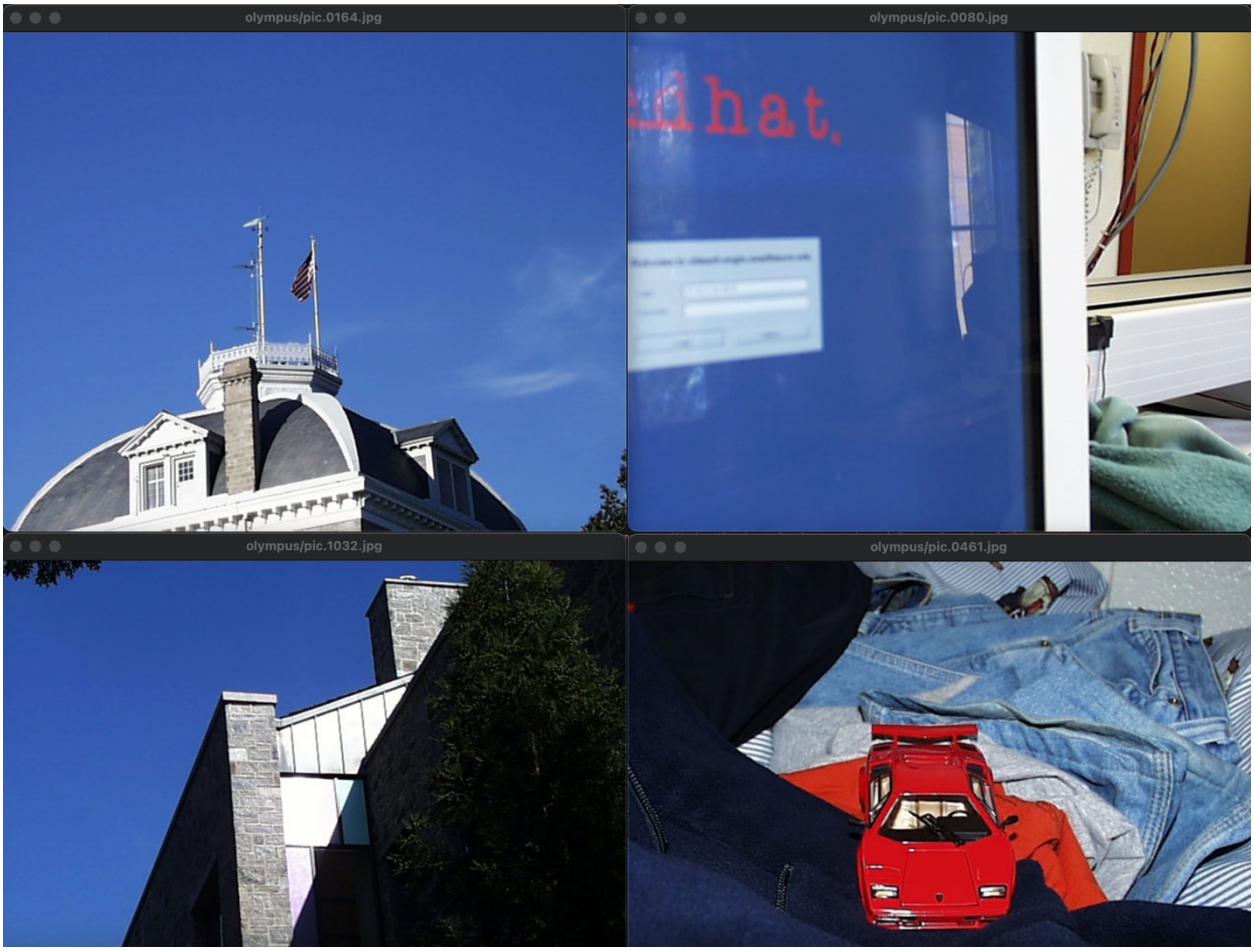
From the terminal output it can be observed that comparing the image with itself results in 0 (in this case comparing pic.1016.jpg with itself). The other images that were listed in the top 5 with ascending order of their MSEs are pic.0095.jpg, pic.0712, pic.0313. We utilized baseline matching for task 1, comparing the 7*7 pixels in the center of the photo. Sum square difference is used as a measure of distance. The matched output from the terminal is displayed above for the target image pic.1016.jpg.


olympus/pic.1016.jpg
olympus/pic.0986.jpg
olympus/pic.0641.jpg
olympus/pic.0547.jpg
olympus/pic.1013.jpg
Project 2 Report
readfiles.cpp
readfiles.o
textureHistogram
topN
topNGeneration
topNGeneration
utils.cpp
utils.h
utils.o
OUTLINE
TIMELINE
topN
bash
bash
bash
Plain Text

Required results 2: show the top three matches for the target image pic.0164.jpg.

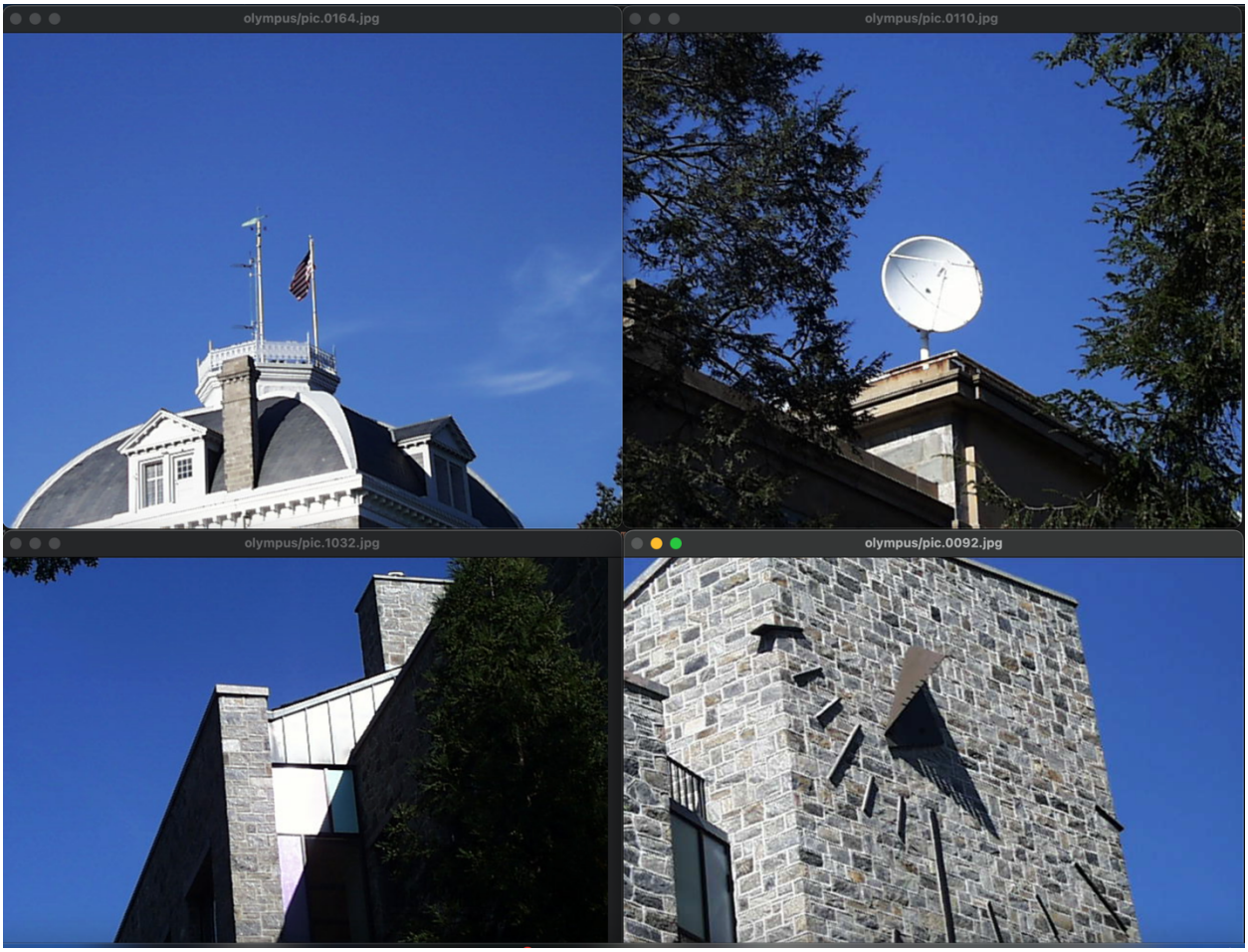
The following results for query pic.0164 were obtained using a whole-image RG chromaticity histogram with 16 bins for both R and G, and histogram intersection as the distance metric.

```
cathyqindeMacBook-Pro:Project 2. Content-based Image Retrieval cathyqin$ ./topN hist2D olympus/pic.0164.jpg 4
Reading /Users/cathyqin/Desktop/Pattern_Recognition_Computer_Vision/Project 2. Content-based Image Retrieval/hist2DfeatureVector.csv
Finished reading CSV file
Matching hist2d1106olympus/pic.0164.jpg:0.00148946
olympus/pic.0080.jpg:0.30935
olympus/pic.1032.jpg:0.373279
olympus/pic.0461.jpg:0.439427
```



The following results for query pic.0164 were obtained using a whole-image RGB histogram with 8 bins for each of R, G, and B, utilizing histogram intersection as the distance metric.

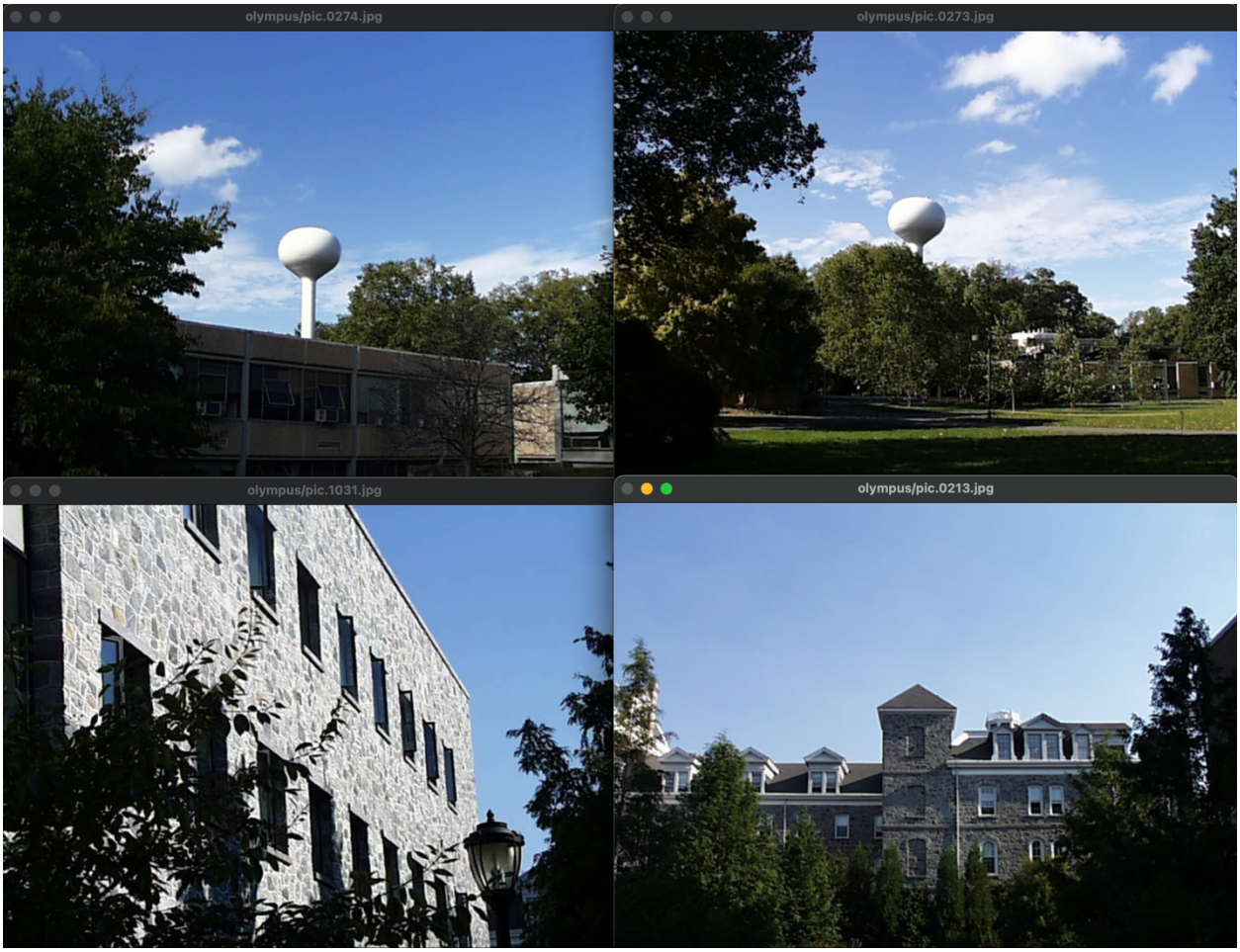
```
cathyqindeMacBook-Pro:Project 2. Content-based Image Retrieval cathyqin$ ./topN hist3D olympus/pic.0164.jpg 4
Reading /Users/cathyqin/Desktop/Pattern_Recognition_Computer_Vision/Project 2. Content-based Image Retrieval/hist3DfeatureVector.csv
Finished reading CSV file
Matching hist3d1106olympus/pic.0164.jpg:0.000990093
olympus/pic.0110.jpg:0.57558
olympus/pic.1032.jpg:0.576317
olympus/pic.0092.jpg:0.610934
```



Required results 3: show the top three matches for the target image pic.0274.jpg.

```
cathyqindeMacBook-Pro:Project 2. Content-based Image Retrieval cathyqin$ ./topN multiHist olympus/pic.0274.jpg 4
Reading /Users/cathyqin/Desktop/Pattern_Recognition_Computer_Vision/Project 2. Content-based Image Retrieval/multiHistFeaturevectors.csv
Finished reading CSV file
Matching top bottom hist1106olympus/pic.0274.jpg:1.00258
olympus/pic.0273.jpg:1.35513
olympus/pic.1031.jpg:1.38819
olympus/pic.0213.jpg:1.40987
```

These results for the query image pic.0274 were obtained using two RGB histograms, representing the top and bottom halves of the image. Each histogram utilized 8 bins per RGB channel, with histogram intersection serving as the distance metric.



Required results 4: show the top three matches for the target image pic.0535.jpg and show how they differ when compared to tasks 2 and 3

```
cathyqindeMacBook-Pro:Project 2. Content-based Image Retrieval cathyqin$ ./topN textureHist "olympus/pic.0535.jpg" 4
Reading /Users/cathyqin/Desktop/Pattern_Recognition_Computer_Vision/Project 2. Content-based Image Retrieval/textureHistogram.csv
Finished reading CSV file
Matching texture and color hist1106olympus/pic.0535.jpg:0.00549769
olympus/pic.0171.jpg:0.339247
olympus/pic.0004.jpg:0.354226
olympus/pic.0829.jpg:0.35879
```

Based on the terminal output and the images above, it is evident that the top three matches are pic.0171.jpg, pic.0004.jpg, and pic.0605.jpg. Initially, we applied a 3D histogram to the RGB image. Additionally, we used a 3D histogram based on the gradient magnitude of the image. For both histograms, we employed histogram intersection as the distance metric.

The difference when compared to task 2 (3D histogram):

```
cathyqindeMacBook-Pro:Project 2. Content-based Image Retrieval cathyqin$ ./topN hist3D "olympus/pic.0535.jpg" 4
Reading /Users/cathyqin/Desktop/Pattern_Recognition_Computer_Vision/Project 2. Content-based Image Retrieval/hist3DfeatureVector.csv
Finished reading CSV file
Matching hist3d1106olympus/pic.0535.jpg:0.00160718
olympus/pic.0285.jpg:0.221018
olympus/pic.0628.jpg:0.226091
olympus/pic.0952.jpg:0.262761
```

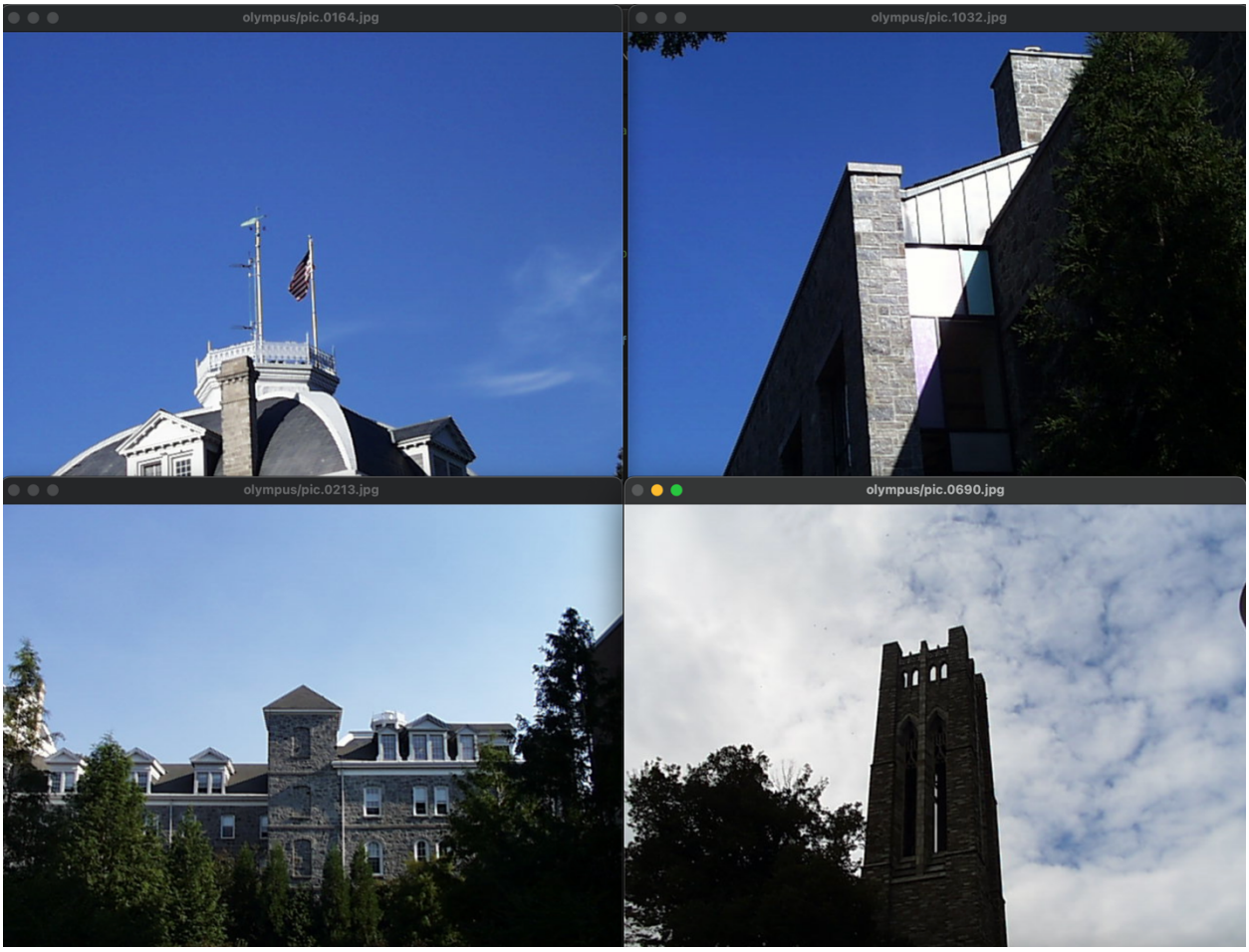
The difference when compared to task 3 (multi histogram):

```
cathyqindeMacBook-Pro:Project 2. Content-based Image Retrieval cathyqin$ ./topN multiHist "olympus/pic.0535.jpg" 4
Reading /Users/cathyqin/Desktop/Pattern Recognition Computer Vision/Project 2. Content-based Image Retrieval/multiHistFeaturevectors.csv
Finished reading CSV file
Matching top bottom hist1106olympus/pic.0535.jpg:1.00279
olympus/pic.0285.jpg:1.30242
olympus/pic.1105.jpg:1.34271
olympus/pic.0972.jpg:1.34274
```

Required results 5: include the top 3 results for images pic.0893.jpg and pic.0164.jpg and compare the results with the prior methods.

```
cathyqindembp:Project 2. Content-based Image Retrieval cathyqin$ ./topN ResNet18 olympus/pic.0893.jpg 4
Reading /Users/cathyqin/Desktop/Pattern_Recognition_Computer_Vision/Project 2. Content-based Image Retrieval/ResNet18_olym.csv
Finished reading CSV file
Matching ResNet181106olympus/pic.0893.jpg:1.19209e-07
olympus/pic.0897.jpg:0.151768
olympus/pic.0136.jpg:0.176157
olympus/pic.0146.jpg:0.224857
```

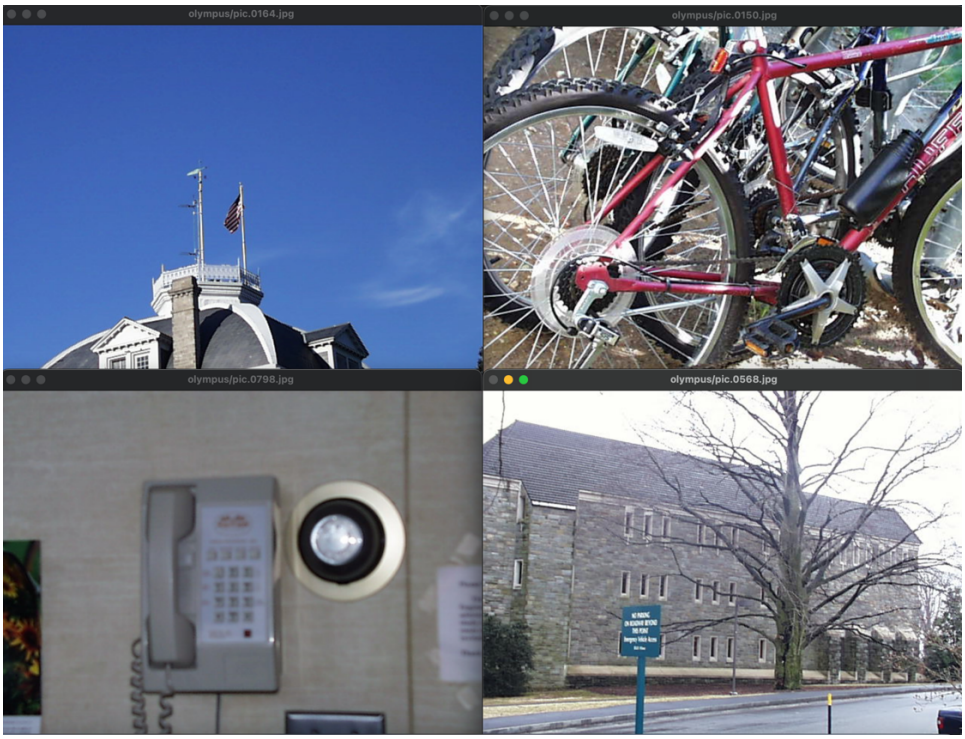
```
cathyqindembp:Project 2. Content-based Image Retrieval cathyqin$ ./topN ResNet18 olympus/pic.0164.jpg 4
Reading /Users/cathyqin/Desktop/Pattern_Recognition_Computer_Vision/Project 2. Content-based Image Retrieval/ResNet18_olym.csv
Finished reading CSV file
Matching ResNet181106olympus/pic.0164.jpg:0
olympus/pic.1032.jpg:0.21219
olympus/pic.0213.jpg:0.212836
olympus/pic.0690.jpg:0.235137
```



Comparison with the prior methods:

Take the result above of pic.0164.jpg as example, cosine distance focuses on comparing if the feature vectors are close in direction. The angular difference between vectors rather than magnitude where relative orientation matters more than scale.

SSD focus on the absolute differences between corresponding values, so it’s used for pixel-wise image comparisons where precise intensity differences matter.



Histogram Intersection is for comparing distributions, such as color histograms, where the goal is to measure similarity in frequency rather than exact values.

Required result 6: compare the DNN embedding and classic features results for 2-3 images of your choice.

Pic.1072 result

DNN:

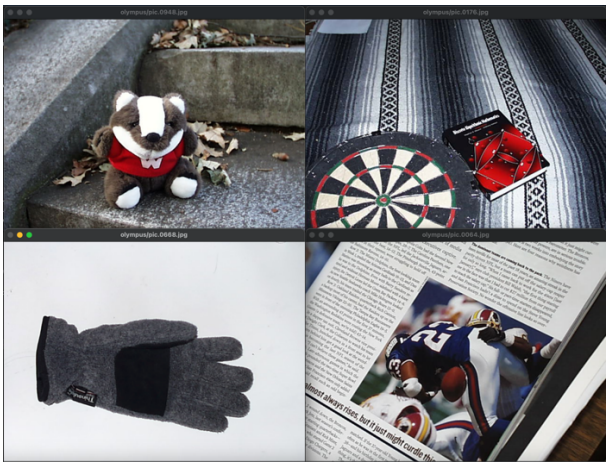
SSD:

Histogram:

Pic.948 result

DNN:

SSD:



Histogram:

Pic.734 result

DNN:

SSD:

Histogram:

No, DNN embeddings are not always better.

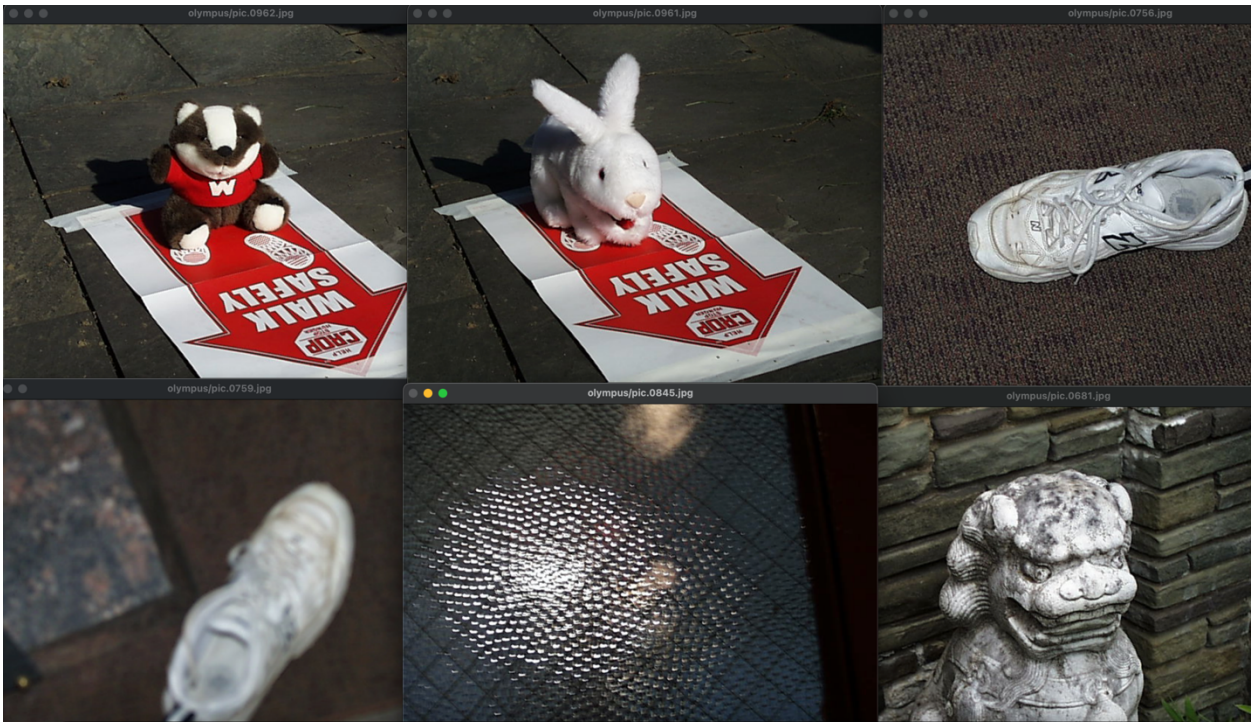
- DNN embeddings are good at capturing or recognizing objects regardless of slight color or texture differences, pic 948 is the good example.
- Classic features can be better for, low-level detail matching, when precise texture, edges, or color details are crucial, pic 0164 is a good example.

Required results 7: Customized design for two target images of your choice, show the top five results. It's also helpful to show some of the least similar results.

With the new method that processes the image by dividing it into left and right halves, focusing on horizontal divisions, it looks for differences or similarities between the left and right sides, while vertical matching looks for patterns between the top and bottom regions of the image.

Target: pic.0679.jpg

Target: pic.0962.jpg



Target: pic.0375

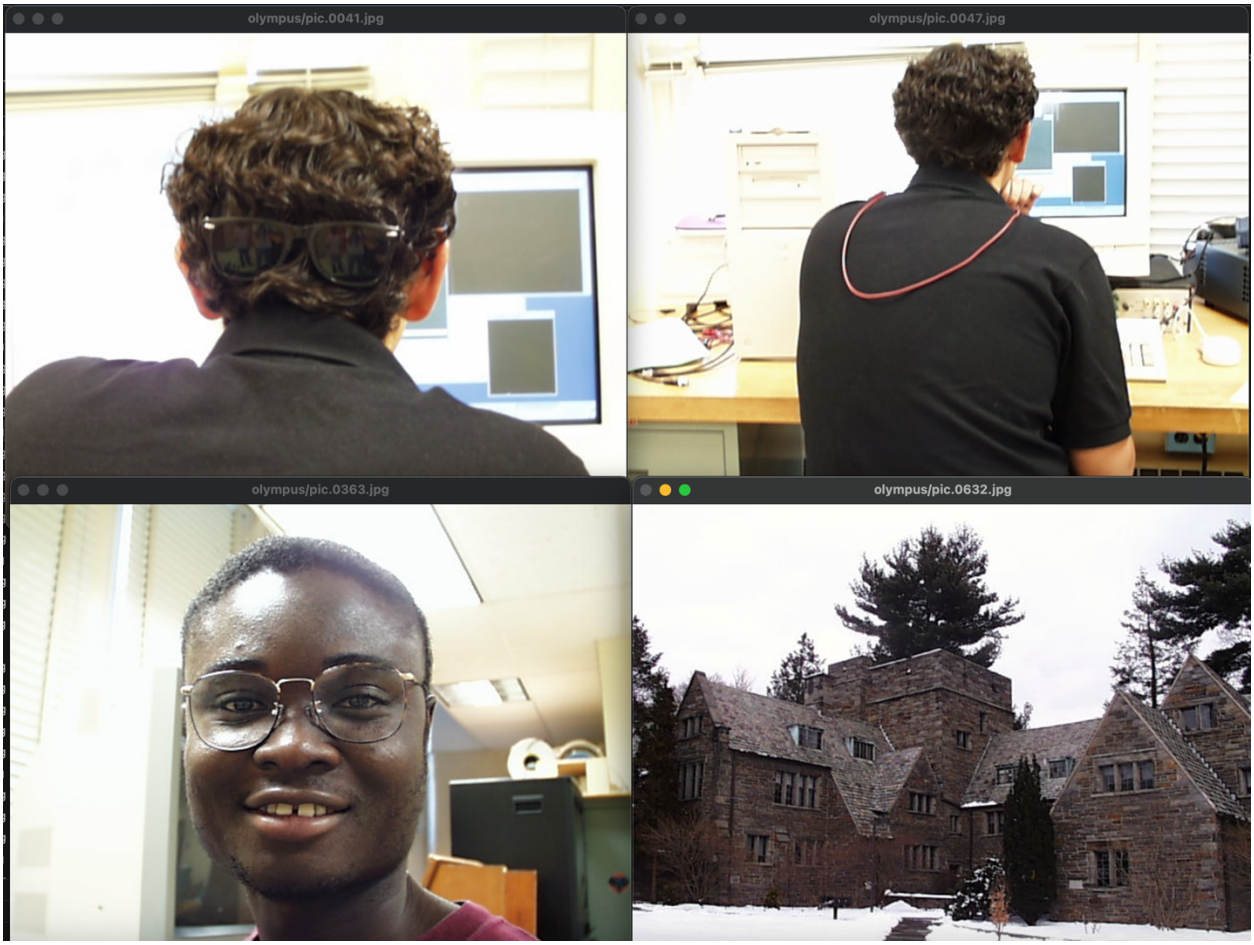
Extensions:

1. Additional features and matching method

Laplacian matching method

Chi-square distance metrics

This is the result of Laplacian feature vector with cos similarity distance metrics:



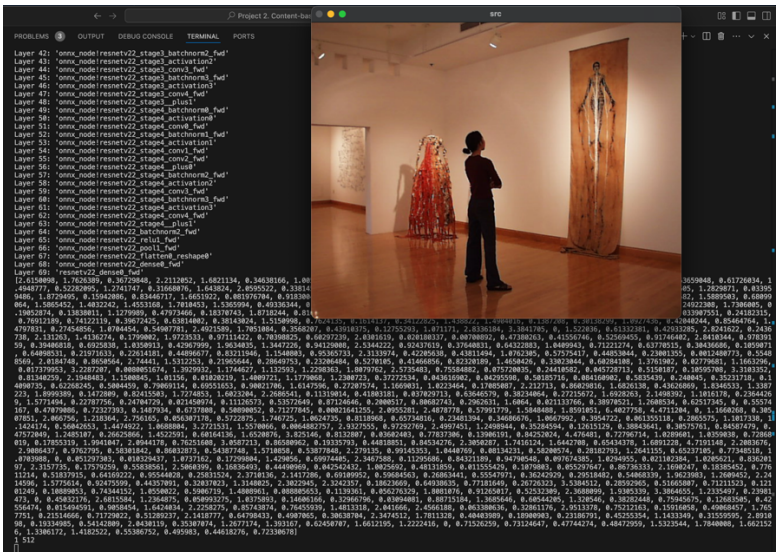
Obviously, Laplacian with cosine similarity distance metric is better for face recognition the Laplacian filter can detect the contours and edges of facial features (eyes, nose, mouth) which are important for identifying individuals.

This is the result of multiHist feature vector with chi-square distance metrics

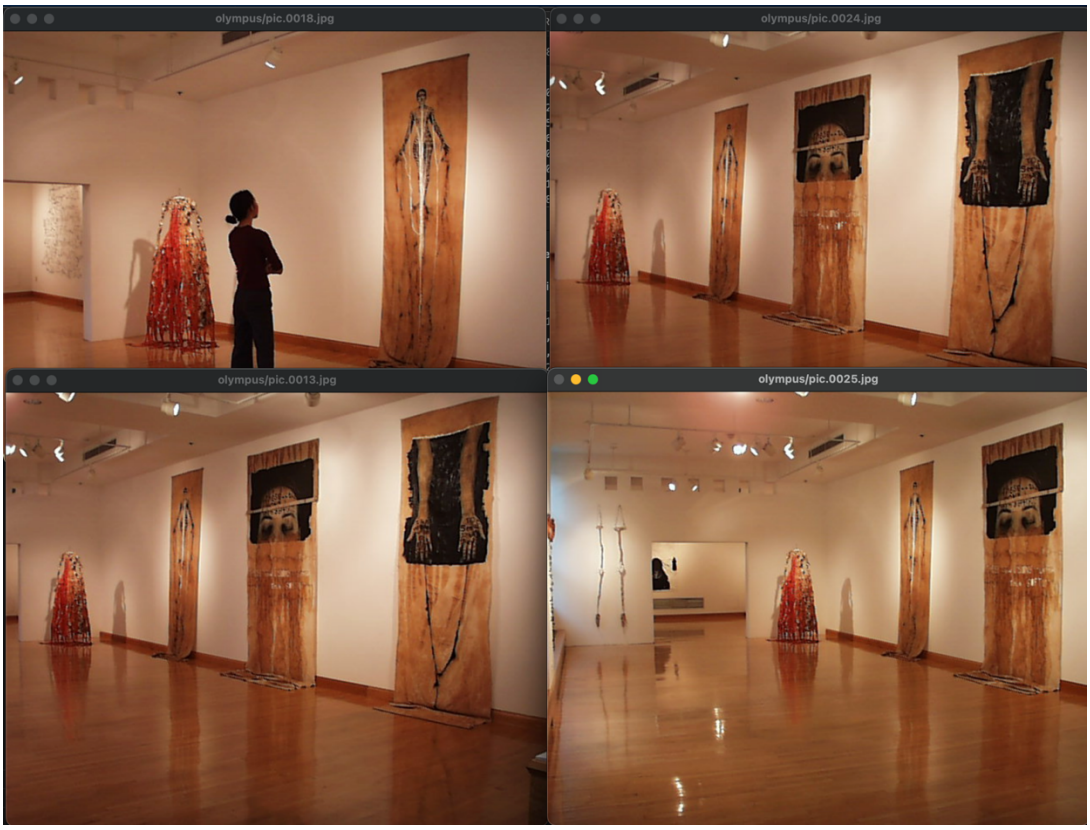
Chi-square is much better for Histogram-based image retrieval where the goal is to find visually similar images. A low Chi-Square value indicates that the images are visually similar, based on color distribution, as you can see from the result.

2. Try other embeddings by using the OpenCV DNN API to read a resNet18 network and get an image embedding.

This is the embedding result for pic.0018.jpg

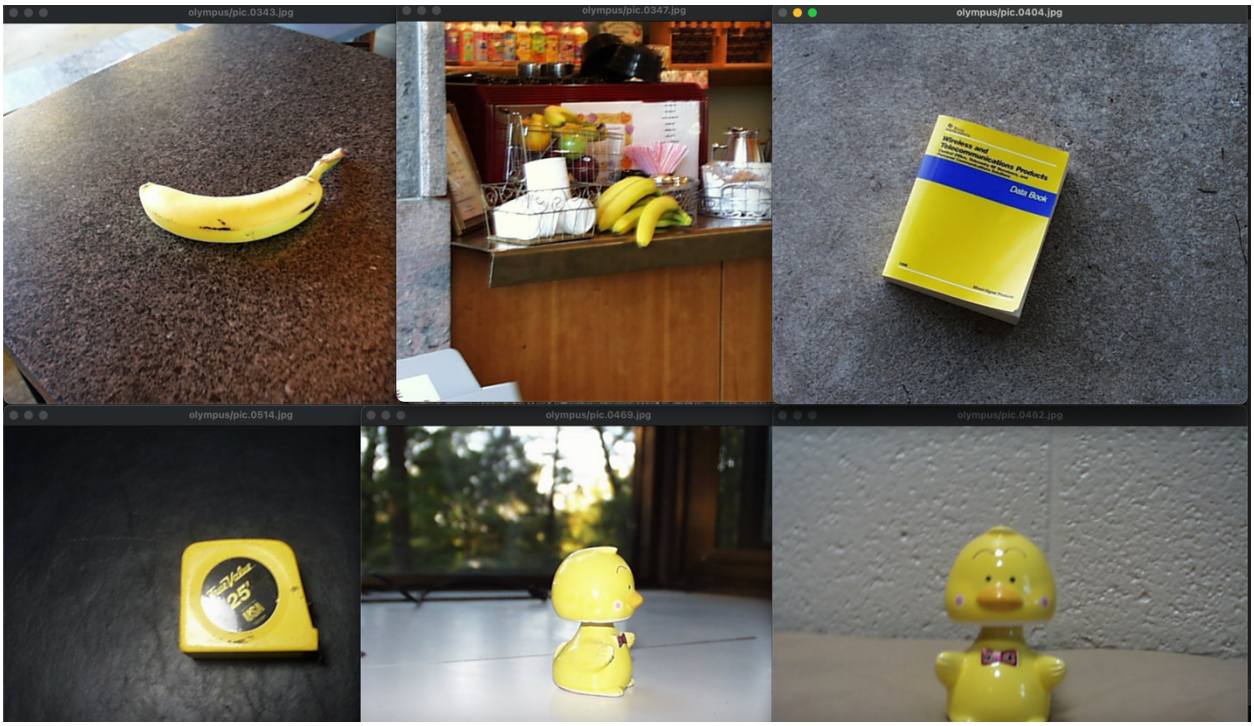


I converted the embeddings as vector<float> and save all the embedding vectors into csv file. The result of histogram intersection distance matching for pic.0018.jpg is below:



3. See if you can find as many pictures with bananas as possible given an input image containing banana.

First, I've tried to crop the image to 150x150 to ensure it captures the center. The image was then converted to HSV color space, followed by thresholding based on yellow. In the example below, the function successfully identifies some yellow matches for a banana image. While not all bananas appear in the top five results, it's observed that yellow objects all included within the top 10.



Reflections:

I've learned the pipeline and PCA logic to perform the CIBR. This assignment has provided valuable insights into image matching using color histograms, multi-histograms, and thresholding techniques. We also learned the importance of considering texture in image analysis. And different distance metric method, e.g: cosine, ssd, chi-square. is suitable for different cases.

Acknowledgements:

Although I didn't choose DAV7, instead, I use horizontal histogram to customize the feature vector extraction. I would like to thank Professor Bruce, TA for their guidance and clarifications on tasks 7, which focused on Depth Anything V2 map feature vector extraction. Their suggestions were invaluable in helping us implement extensions to filter pixels in the image dataset.