Aayush Hitesh Sanghvi Yogi Hetal Shah

Project 2 - PRCV

Abstract

Project was designed and programmed to achieve the seven required tasks with added custom design and a sophisticated command line GUI which gives the user flexibility in interchanging and choosing how the feature vectors are calculated and how the distance metric is computed. With respect to Code flow the structure is organized and optimized in way that the repeatability of the code is reduced and the int main () is straightforward to understand. The code starts by looping over all the images in the data set and adding them to csv files for different feature extraction techniques like baseline, histogram, ... After this the user is asked to choose from what type of feature extraction, he wants go ahead for the target image. After this the user will be prompted to choose from the distance metric and then compute and show results.

Results

Task 1: Baseline Matching Results

```
yogi@vostro:~/PRCV/Project-2/build$ ./project-2 /home/yogi/PRCV/Project-2/olympu
s /home/yogi/PRCV/Project-2/olympus/pic.1016.jpg 3
Processing directory /home/yogi/PRCV/Project-2/olympus

Please enter the number for what kind of feature vector you want to generate.

1: 7x7 matrix in the center of the image.
2: 2-D RG chromaticity histogram of the image.
3: 3-D RGB hustogram of the image.
4: Texture features of the image.
5: Deep network embeddings of the image.
1
Which distance metrics do you want to choose?
1: Euclidean Distance
2: Histogram Intersection
3: Split Histogram Intersection
4: Texture Distance Metrics
5: Custom designed Distance Metrics
1
Top 3 matches for Baseline matching are
118: /home/yogi/PRCV/Project-2/olympus/pic.0986.jpg
147: /home/yogi/PRCV/Project-2/olympus/pic.0641.jpg
222: /home/yogi/PRCV/Project-2/olympus/pic.0547.jpg
```

Target image

Task 2: Histogram Matching Results

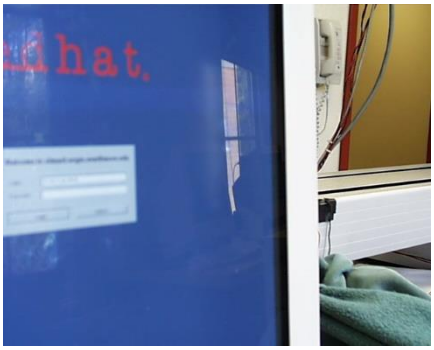
```
yogi@vostro:~/PRCV/Project-2/build$ ./project-2 /home/yogi/PRCV/Project-2/olympu
s /home/yogi/PRCV/Project-2/olympus/pic.0164.jpg 3
Processing directory /home/yogi/PRCV/Project-2/olympus

Please enter the number for what kind of feature vector you want to generate.

1: 7x7 matrix in the center of the image.
2: 2-D RG chromaticity histogram of the image.
3: 3-D RGB hustogram of the image.
4: Texture features of the image.
5: Deep network embeddings of the image.
2
Euclidean Distance cannot be calculated for this feature type.
Which distance metrics do you want to choose?
1: Histogram Intersection
2: Split Histogram Intersection
3: Texture Distance Metrics
4: Custom designed Distance Metrics
1
entery
Top 3 matches for Histogram matching are
0.6941: /home/yogi/PRCV/Project-2/olympus/pic.0080.jpg
0.6571: /home/yogi/PRCV/Project-2/olympus/pic.0489.jpg
0.647: /home/yogi/PRCV/Project-2/olympus/pic.0461.jpg
```

Target image





Task 3: Multi-histogram Matching Results

```
yogi@vostro:~/PRCV/Project-2/build$ ./project-2 /home/yogi/PRCV/Project-2/olympu
s /home/yogi/PRCV/Project-2/olympus/pic.0274.jpg 3
Processing directory /home/yogi/PRCV/Project-2/olympus

Please enter the number for what kind of feature vector you want to generate.

1: 7x7 matrix in the center of the image.
2: 2-D RG chromaticity histogram of the image.
3: 3-D RGB hustogram of the image.
4: Texture features of the image.
5: Deep network embeddings of the image.
3
Euclidean Distance cannot be calculated for this feature type.
Which distance metrics do you want to choose?
1: Histogram Intersection
2: Split Histogram Intersection
3: Texture Distance Metrics
4: Custom designed Distance Metrics
1
entery
Top 3 matches for Histogram matching are
0.7588: /home/yogi/PRCV/Project-2/olympus/pic.0409.jpg
0.7373: /home/yogi/PRCV/Project-2/olympus/pic.0825.jpg
0.7095: /home/yogi/PRCV/Project-2/olympus/pic.0335.jpg
```

Target image



For 3D histogram matching, first generate an empty 3D 8x8x8 histogram, then calculate r index, b index, and g index from each pixel of the picture, increase the position in the histogram matrix at rindex, gindex, and bindex, normalise the histogram, and save the histogram as a feature vector. Next, we created two distinct histogram intersection "distance metrics" for both the target image and images in the database, one for the upper half of the image and the other for the lower half of the image, compared the results, and computed the distance metrics. Depending on the distance measurements, returned top N best matches for the target picture.

Aayush Hitesh Sanghvi
Yogi Hetal Shah

Task 4: Texture and Colour Matching Results

```
aayush@aayush-Dell-G16-7620:~/PRCV/Project-2/build$ ./project-2 /home/aayush/PRCV/Project-2/o
Processing directory /home/aayush/PRCV/Project-2/olympus

Please enter the number for what kind of feature vector you want to generate.

1: 7x7 matrix in the center of the image.
2: 2-D RG chromaticity histogram of the image.
3: 3-D RGB hustogram of the image.
4: Texture features of the image.
5: Deep network embeddings of the image.
5
For this feature, either Euclidean Distance metrics can be calculated or Custom designed dista
Which distance metrics do you want to choose?
1: Euclidean Distance
2: Custom designed Distance Metrics
1
Top 3 matches for Baseline matching are
15: /home/aayush/PRCV/Project-2/olympus/pic.1032.jpg
16: /home/aayush/PRCV/Project-2/olympus/pic.1101.jpg
17: /home/aayush/PRCV/Project-2/olympus/pic.1104.jpg
```

Target image

For texture matching, we calculated the orientation and magnitude of each image in the database and generated a 2D 16x16 histogram of orientation vs magnitude, resulting in 256 bins. The top N best matches were produced after calculating texture distance metrics between the target image and the image in the database.

Task 5: Deep Neural Network Embedding Matching Results

```
yogi@vostro:~/PRCV/Project-2/build$ ./project-2 /home/yogi/PRCV/Project-2/olympu
s /home/yogi/PRCV/Project-2/olympus/pic.0893.jpg 3
Processing directory /home/yogi/PRCV/Project-2/olympus

Please enter the number for what kind of feature vector you want to generate.

1: 7x7 matrix in the center of the image.
2: 2-D RG chromaticity histogram of the image.
3: 3-D RGB hustogram of the image.
4: Texture features of the image.
5: Deep network embeddings of the image.
5
For this feature, either Euclidean Distance metrics can be calculated or Custom
designed distance metric can be calculated.
Which distance metrics do you want to choose?
1: Euclidean Distance
2: Custom designed Distance Metrics
2
Top 3 matches for Custom designed distance metrics are
1.4019: /home/yogi/PRCV/Project-2/olympus/pic.0136.jpg
1.17596: /home/yogi/PRCV/Project-2/olympus/pic.0897.jpg
-0.383506: /home/yogi/PRCV/Project-2/olympus/pic.0135.jpg
```

Target image

Aayush Hitesh Sanghvi Yogi Hetal Shah

Task 6: DNN vs Classic Matching Results

```
yogi@vostro:~/PRCV/Project-2/build$ ./project-2 /home/yogi/PRCV/Project-2/olympu
s /home/yogi/PRCV/Project-2/olympus/pic.1072.jpg 3
Processing directory /home/yogi/PRCV/Project-2/olympus

Please enter the number for what kind of feature vector you want to generate.

1: 7x7 matrix in the center of the image.
2: 2-D RG chromaticity histogram of the image.
3: 3-D RGB hustogram of the image.
4: Texture features of the image.
5: Deep network embeddings of the image.
5
For this feature, either Euclidean Distance metrics can be calculated or Custom
designed distance metric can be calculated.
Which distance metrics do you want to choose?
1: Euclidean Distance
2: Custom designed Distance Metrics
1
Top 3 matches for Baseline matching are
17: /home/yogi/PRCV/Project-2/olympus/pic.0863.jpg
18: /home/yogi/PRCV/Project-2/olympus/pic.0937.jpg
19: /home/yogi/PRCV/Project-2/olympus/pic.1012.jpg

yogi@vostro:~/PRCV/Project-2/build$ ./project-2 /home/yogi/PRCV/Project-2/olympu
s /home/yogi/PRCV/Project-2/olympus/pic.1072.jpg 3
Processing directory /home/yogi/PRCV/Project-2/olympus

Please enter the number for what kind of feature vector you want to generate.

1: 7x7 matrix in the center of the image.
2: 2-D RG chromaticity histogram of the image.
3: 3-D RGB hustogram of the image.
4: Texture features of the image.
5: Deep network embeddings of the image.
1
Which distance metrics do you want to choose?
1: Euclidean Distance
2: Histogram Intersection
3: Split Histogram Intersection
4: Texture Distance Metrics
5: Custom designed Distance Metrics
1
Top 3 matches for Baseline matching are
280: /home/yogi/PRCV/Project-2/olympus/pic.0768.jpg
281: /home/yogi/PRCV/Project-2/olympus/pic.0138.jpg
287: /home/yogi/PRCV/Project-2/olympus/pic.0303.jpg
```

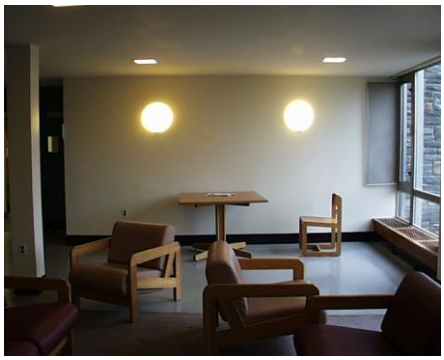
Target images

Figures above are the Results of 1072 using DNN matching.

Figures above are the Results of 1072 using Classical matching.

Figures above are the Results of 0734 using DNN matching

Aayush Hitesh Sanghvi Yogi Hetal Shah



Figures above are the Results of 0734 using Classical matching

Observation: While comparing the two types of features matching it very evident that DNN is a better choice as it not only extracts and recognizes the features but gives a distance metric of the complete image. If the purpose is checking the feature in a small area then it would be a be easier for the classical matching to perform easily.

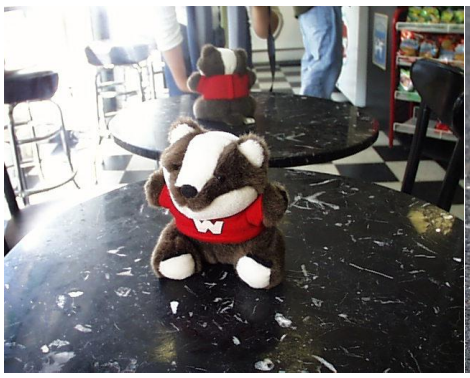
Task 7: Custom Matching Results

```
yogi@vostro:~/PRCV/Project-2/build$ ./project-2 /home/yogi/PRCV/Project-2/olympu
s /home/yogi/PRCV/Project-2/olympus/pic.0948.jpg 5
Processing directory /home/yogi/PRCV/Project-2/olympus

Please enter the number for what kind of feature vector you want to generate.

1: 7x7 matrix in the center of the image.
2: 2-D RG chromaticity histogram of the image.
3: 3-D RGB hustogram of the image.
4: Texture features of the image.
5: Deep network embeddings of the image.
4
Euclidean Distance cannot be calculated for this feature type.
Which distance metrics do you want to choose?
1: Histogram Intersection
2: Split Histogram Intersection
3: Texture Distance Metrics
4: Custom designed Distance Metrics
4
Top 5 matches for Custom designed distance metrics are
1.07544: /home/yogi/PRCV/Project-2/olympus/pic.0930.jpg
-0.687883: /home/yogi/PRCV/Project-2/olympus/pic.0962.jpg
-0.913792: /home/yogi/PRCV/Project-2/olympus/pic.0972.jpg
-1.6789: /home/yogi/PRCV/Project-2/olympus/pic.0060.jpg
-1.73759: /home/yogi/PRCV/Project-2/olympus/pic.0120.jpg
```

Target Image




In this code we have performed weighted average of the DNN matching and 2D histogram. And computed the results.

Aayush Hitesh Sanghvi Yogi Hetal Shah

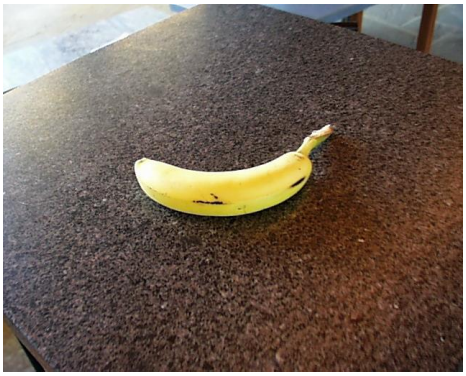
Extensions : Checking for bananas



```
yogi@vostro:~/PRCV/Project-2/build$ ./project-2 /home/yogi/PRCV/Project-2/olympu
s /home/yogi/PRCV/Project-2/olympus/pic.0343.jpg 3
Processing directory /home/yogi/PRCV/Project-2/olympus

Please enter the number for what kind of feature vector you want to generate.

1: 7x7 matrix in the center of the image.
2: 2-D RG chromaticity histogram of the image.
3: 3-D RGB hustogram of the image.
4: Texture features of the image.
5: Deep network embeddings of the image.
5
For this feature, either Euclidean Distance metrics can be calculated or Custom
designed distance metric can be calculated.
Which distance metrics do you want to choose?
1: Euclidean Distance
2: Custom designed Distance Metrics
2
Top 3 matches for Custom designed distance metrics are
-2.7462: /home/yogi/PRCV/Project-2/olympus/pic.0650.jpg
-2.96644: /home/yogi/PRCV/Project-2/olympus/pic.0342.jpg
-3.0576: /home/yogi/PRCV/Project-2/olympus/pic.0348.jpg
```



Extensions: checking how many trash cans can the system detect and

```
Top 10 matches for Custom designed distance metrics are
-1.00235: /home/yogi/PRCV/Project-2/olympus/pic.0291.jpg
-3.43555: /home/yogi/PRCV/Project-2/olympus/pic.0289.jpg
-4.54472: /home/yogi/PRCV/Project-2/olympus/pic.0177.jpg
-4.72616: /home/yogi/PRCV/Project-2/olympus/pic.0920.jpg
-4.78615: /home/yogi/PRCV/Project-2/olympus/pic.0969.jpg
-4.84341: /home/yogi/PRCV/Project-2/olympus/pic.0318.jpg
-4.97111: /home/yogi/PRCV/Project-2/olympus/pic.0214.jpg
-5.1049: /home/yogi/PRCV/Project-2/olympus/pic.0665.jpg
-5.13658: /home/yogi/PRCV/Project-2/olympus/pic.0201.jpg
-5.14194: /home/yogi/PRCV/Project-2/olympus/pic.0152.jpg
```






we could detect 6 out of top 10 images using the DNN + histogram matching.

Extensions: As a part of extension we have also made a sophisticated command line version for integrating a user-friendly experience in interacting with the functions and methods.

Aayush Hitesh Sanghvi Yogi Hetal Shah

Reflection

While performing the various tasked and understand how feature mapping is done, we realized how the computing and selection of the features is done, how the textures, colours and filters can be used to detect similarities in images. With more testing and tuning the code we understood what change gives different images giving an intuitive sense on how the filters and detection is thinking.

Major takeaway is how by increasing the number of filters gets to better feature detection and a sense of how the DNN processing might have been done and how it gets such accurate results. While making the Custom design task we implemented a different colour scheme and initially it gave results with random image of dark objects and the target image was 0948, and later we realized that we not only took a small feature but also a position that was the nose of the bear which is black in colour and hence the matches we got were accurate.

Acknowledgements

1. OpenCV Docs : https://docs.opencv.org/4.x/index.html
2. Stack overflow: https://stackoverflow.com/questions/19641597/what-is-segmentation-fault-core-dumped
3. https://www.sciencedirect.com/science/article/pii/S1319157812000444
4. Professor and TA : Arun Madhusudhanan, Tejaswini Dilip Deore, Erica Sheperd, Poorna Chandra Vemula