Project 2: Content Based Image Retrieval

Samara Holmes
Northeastern University

I. Introduction

The objective is to work with a large database of images and to find images in within the database that are similar to the target image. The purpose of this project is to learn how to manipulate and analyze images at the pixel level. We will be using classic features and deep network embeddings to extract features and recognize patterns.

II. Methodology

For this project we have the following tasks:

1) Baseline matching
2) Histogram matching
3) Multi-histogram matching
4) Texture and color
5) Deep network embeddings
6) Compare DNN embeddings and classic features
7) Custom content-based image retrieval (CBIR) design

A 7x7 square in the middle of the image was used as a feature vector for the baseline matching. With that, a sum-of-squared-difference could be calculated as the distance metric. Once the distance metric is calculated for every image in the database, we can look for the smallest distances for the closest matches.

Histogram matching is a technique used to identify images with similar color distributions. For this approach I created an RG chromaticity histogram, then used the histogram intersection between the target image and the comparing image as the distance metric. Then, I adapted the RG chromaticity histogram, and used a whole image RGB histogram for a comparison later on.

Next, adapting the histogram matching, we move to multi-histogram matching. For this I used two color histograms as the feature vector. I use a whole-image histogram for the first histogram, then a histogram that looks only at the center of the image as the second.

To do the texture feature matching, we adapted the histogram matching and added in a whole image texture histogram as the feature vector. For the texture, a magnitude gradient of the sobel filter in the X and Y direction is used. These features are weighed equally.

For the deep network embeddings, we needed to use a given CSV file that contained the output of the final average pooling layer of a ResNet18 deep network pre-trained on ImageNet. From there, we selected a distance metric to do that matching. Since we use sum-of-squared-difference in the baseline matching, we used it here as well. The deep network embeddings and the histogram matching were used to run the algorithms on two different images and compare.

Lastly, we implemented a few custom algorithms for the content-based image retrieval including, face recognition and SIFT.

III. Results

1016 0996 0641 0547

Figure 1. Top 3 matches for target image 1016 using 7x7 square as a feature vector and sum-of-squared-differences (baseline matching).

0164 0110 0080 1032

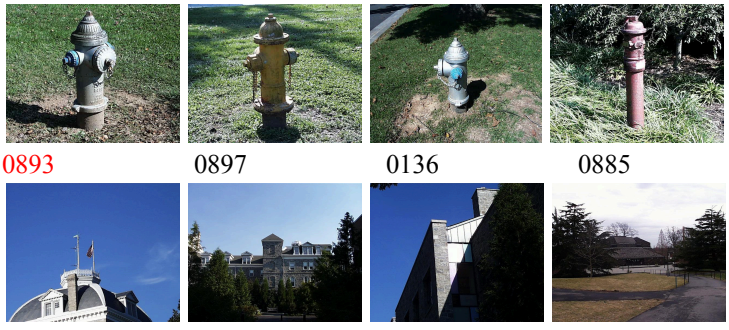
Figure 2. Top 3 matches for target image 0164 using histogram matching (RG chromaticity).

0274 0275 0294 0825

Figure 3. Top 3 matches for target image 0274 using multi-histogram matching (RG chromaticity).

0535 0534 0338 1059

Figure 4. Top 3 matches for target image 0535 using texture feature matching.



0893 0897 0136 0885

0164 0213 1032 0543

Figure 5. Top 3 matches for target images 0893 and 0164 using deep network embeddings and sum-of-squared-difference metric.

1072 (DNN) 0143 0329 0940

0948 (DNN) 0930 0928 0960

1072 (classic) 0936 0937 0233

0948 (classic) 1019 1059 0233

Figure 6. Comparison of the DNN embedding and classic feature (RGB chromaticity histogram) results with target images 1072 and 0948.

0343 0330 0335 0329

Figure 7. Top 3 matches with bananas given target image 0343 (which contains a banana) using SIFT and sum-of-squared-differences.

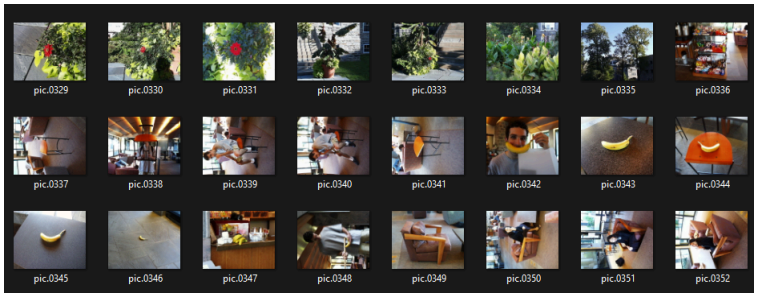


Figure 8. Database of images used for Figure 7.

0343 0282 0345 0289

Figure 9. Top 3 matches with bananas given target image 0343 (which contains a banana) using SIFT and sum-of-squared-differences with a different small database of images.

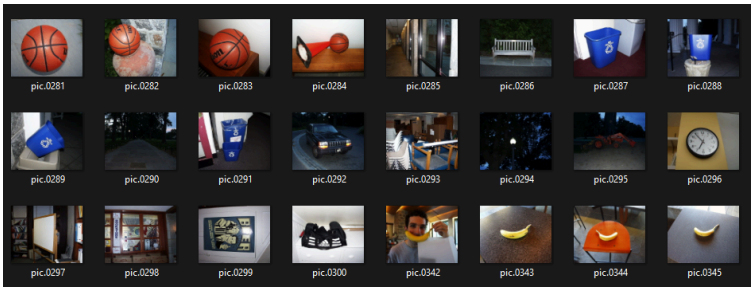


Figure 10. Database of images used for Figure 9.

0287 0289 0291 0288

Figure 11. Top 3 matches with recycling bins given target image 0287 (which contains a recycling bin) using SIFT and sum-of-squared-differences.

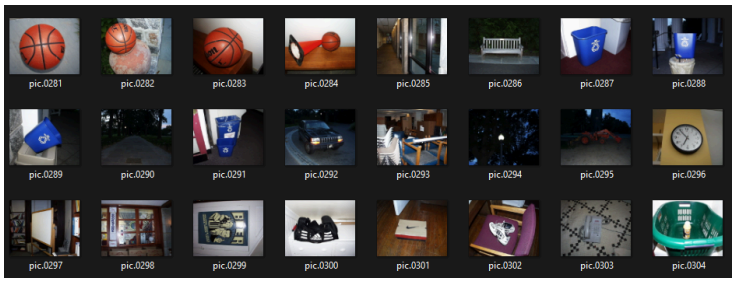


Figure 12. Database of images used for Figure 11.

0006 0075 0299 0872

Figure 13. Samples retrieved with faces given target image 0006 (which contains a face).

IV. Discussion

Figure 1 is a simple run of the baseline matching using image 1016 as the target. Since it was run using a 7x7 square region in the center as the feature vector, it returned the top results of images showing a red color in the center.

Figure 2 displays the top 3 image matches when using an RG chromaticity histogram as the feature vector and the intersection as the distance metric. The histogram intersection reflects the similarity between two histograms.

Figure 3 displays the top 3 matches when using a multi-histogram match where the first histogram is the whole-image RG chromaticity and the second is a smaller center of the image histogram. Ensuring that the weight for the whole image histogram was greater than the center of the image histogram weight for the distance metric was very important. This is because it matters more what the whole image looks like, as opposed to just the center section.

Figure 4 shows top 3 matches using a whole image color histogram and a whole image texture histogram as the feature vector. In contrast to what was done in Figure 3, the two histograms were weighed equally.

Figure 5 displays the deep network embedding's ability to determine the similarity between images in the database to the target image. For the fire hydrant image, image 0893, we were able to find matches containing fire hydrants. However, for image 0164, which is an image with a building and blue sky, it didn't perform as well.The feature values from the ResNet18 model allowed for the algorithm to look for singular objects better than larger scenes.

Figure 6 shows the comparison of results when using DNN embeddings versus classic features, in this case, RGB chromaticity histograms. Figure 6 shows that DNN embeddings isn't always better. For image 0948, DNN embeddings definitely perform better than RGB chromaticity, but this is because it is using a pre-trained model specific to the database. However, for image 1072, DNN embeddings and the RGB chromaticity seemed to have performed similarly. This is likely due to DNN embeddings performing better with complex features, and RGB chromaticity responding well to images where color similarity is more important than the features.

As a custom design algorithm, I decided to use Scale-Invariant Feature Transform (SIFT) which retrieves keypoints from the image, then computes descriptors from the keypoints.This took awhile to compute the sum-of-squared-difference score, so I used a small subsection of the image database. See Figure 7, 9, and 11 for results.

Figure 7 took significantly longer to compute as opposed to Figure 11. Figure 11 also provided better results. This may be due to the complexity of the images provided with the banana database samples (see Figure 8). With the recycling bins, the images were darker and didn't contain as many features (see Figure 12). Since I noticed this, I replaced the recycling bin images in the database in Figure 12, with the 4 banana images, resulting in the database in Figure 10. When running the algorithm again, I got the results in Figure 9. Not a perfect identification of all the bananas, but better than the first attempt using SIFT.

As an extension, I used a face detection algorithm to be able to return images if a face is detected (see Figure 13). When I initially ran this, I received a list of images that had faces, but also a lot that didn't upon visual inspection. To mitigate this error, I added a threshold on the detected face width and height. Initially, I used only an upper threshold, but I realized that an upper and lower threshold would work better. There we're still a couple of false positives, however, this could be improved with more tuning.

V. Conclusion

The analysis compared various image matching techniques, finding different strengths and limitations in each. Baseline matching and RG chromaticity histograms did well in specific color detection, while multi-histogram and deep network embeddings offered improved accuracy for more complex features. The SIFT algorithm showed varied performance based on image complexity, and face detection required additional tuning for better accuracy. Overall, the project enhanced the understanding of content based image retrieval and matching performance with different methods and features.

VI. References

https://docs.opencv.org/3.4/da/df5/tutorial_py_sift_intro.html

https://towardsai.net/p/machine-learning/chromaticity-segmentation