John Hacker
EEL 4660
Object Detection

Procedure

After opening all of the images, my program uses both object detection methods to query the dataset of images. I used COMP_INTERSECT for histogram comparing and TM_CCOEFF for template matching because they yielded the best results for each type of object detection. After calculating the matching score for each image in the dataset, the top four are compared to the actual four images in the same class as the query image. The query gets one point for each image in the top that is in the original classification. After every query, information about the background and results of the query is printed. After all of the queries have occurred for both methods, some statistics about the queries' success is printed. See below pseudo code to describe the procedure of the program in basic steps.

Pseudo Code:

```
readImages()
for query_image in images:
	query_histogram = histogram(query_image)
	for image in images:
		image_histogram = histogram(image)
		hist_score = compareHistograms(image_histogram, query_histogram)
		template_score = matchTemplate(image, query_image)
	if score > min(top4_scores):
		top4_scores.replace(min(top4_scores), score)
	top4_scores.sort()
	number_correct = inQueryClass(top4_scores)
	printQueryReport()
printFinalStatistics()
```

method=CV_TM_CCOEFF

$$R(x,y) = \sum_{x',y'} (T'(x',y') \cdot I'(x+x', y+y'))$$

$$T'(x',y') = T(x',y') - 1/(w \cdot h) \cdot \sum_{x'',y''} T(x'',y'')$$
$$I'(x+x', y+y') = I(x+x', y+y') - 1/(w \cdot h) \cdot \sum_{x'',y''} I(x+x'', y+y'')$$

Intersection (`method=CV_COMP_INTERSECT`)

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

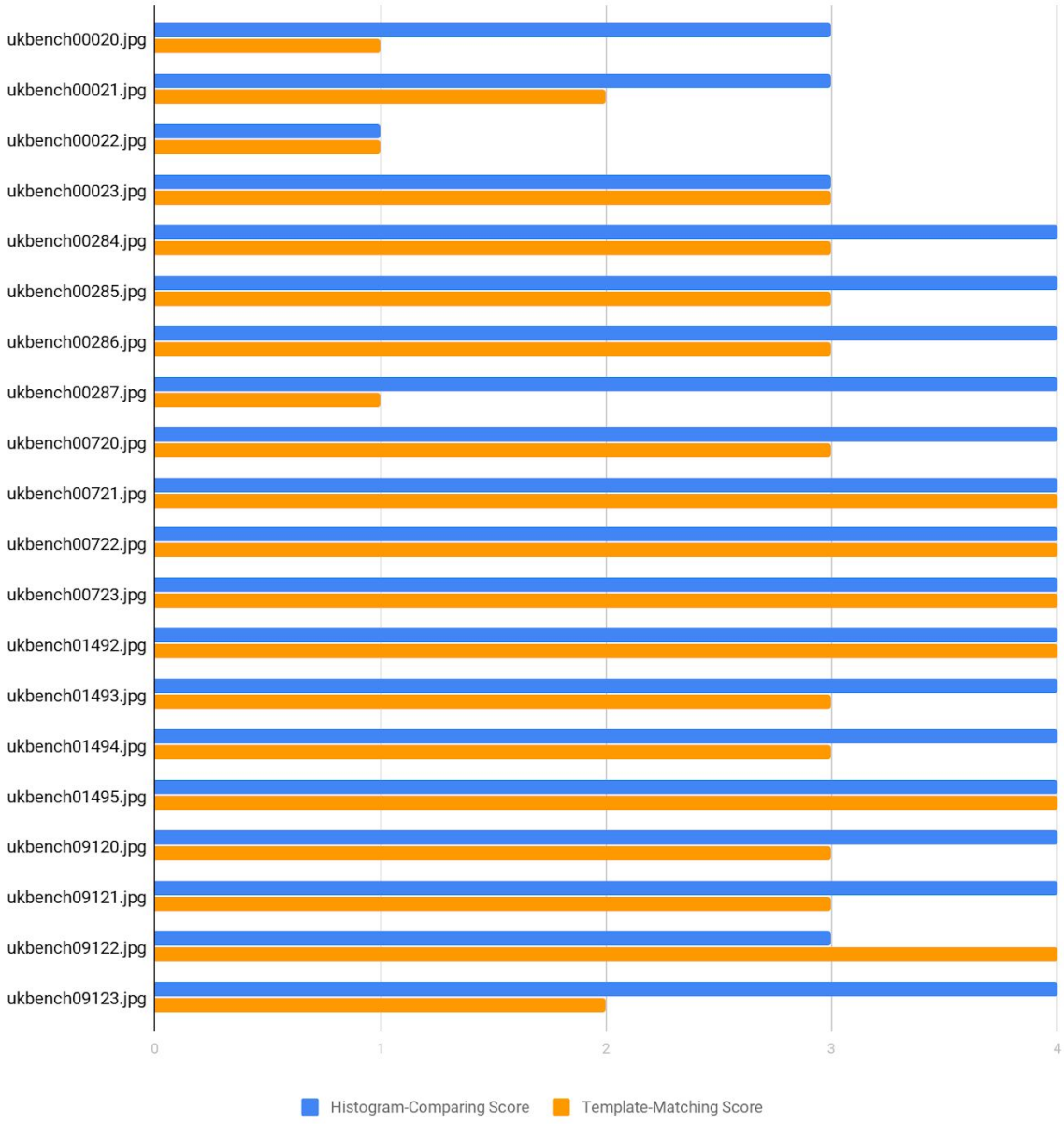
$$\bar{H_k} = \frac{1}{N} \sum_J H_k(J)$$

Statistics

The mean score for histogram comparing is 3.65 with a standard deviation of 0.726. The mean score for template matching is 2.9 with a standard deviation of 0.995. See below for a table and bar graph comparing the successes of the two object detection methods.

File Name	Histogram Comparing Score	Template Matching Score
ukbench00020.jpg	3	1
ukbench00021.jpg	3	2
ukbench00022.jpg	1	1
ukbench00023.jpg	3	3
ukbench00284.jpg	4	3
ukbench00285.jpg	4	3
ukbench00286.jpg	4	3
ukbench00287.jpg	4	1
ukbench00720.jpg	4	3
ukbench00721.jpg	4	4
ukbench00722.jpg	4	4
ukbench00723.jpg	4	4
ukbench01492.jpg	4	4
ukbench01493.jpg	4	3
ukbench01494.jpg	4	3
ukbench01495.jpg	4	4
ukbench09120.jpg	4	3
ukbench09121.jpg	4	3
ukbench09122.jpg	3	4
ukbench09123.jpg	4	2

Object Detection



Bad Detection

Based on this test, histogram comparing is both more accurate and consistent than template matching. I believe this happened because histogram matching is more invariant to warping--such as rotation. Both algorithms could not identify an image besides the original query image for ukbench00022.jpg (bottom-left image below) because the brightness for that image is different than the other three in its class. This made the colors look like they were different than the query, even though they were basically just translated on the histogram.