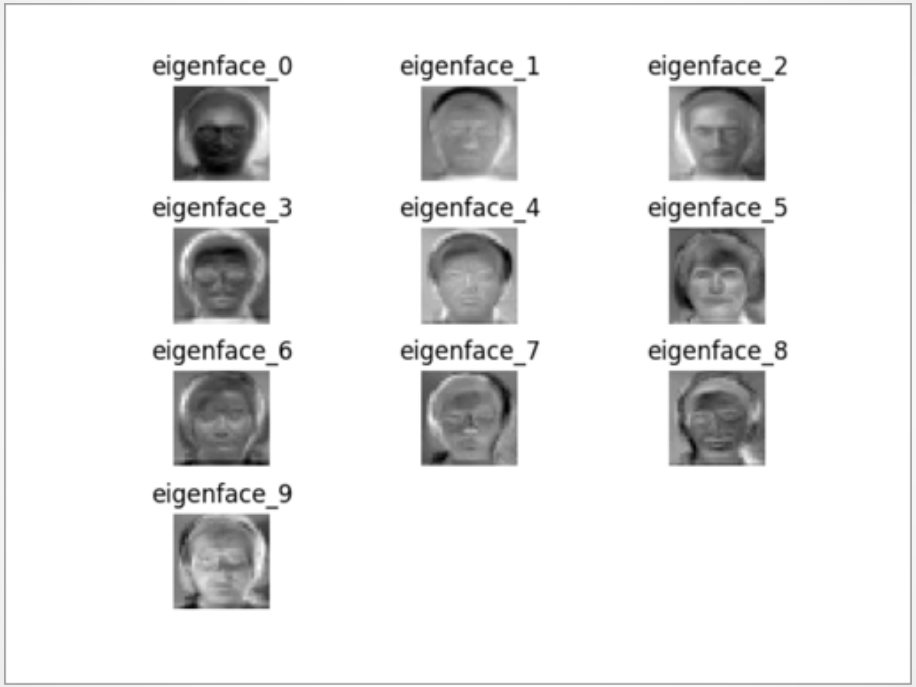
Computer Vision
Fall 2018
Problem Set #6

James Peruggia
jperuggia@gatech.edu

1a: Average face

ps6-1-a-1.png

1b: Eigenvectors



ps6-1-b-1.png

1c: Analysis

The predictions here perform much better than randomly selecting a value between 1 and 15. To validate this I randomly selected a value of 1 to 15 and compared the results to the PCA selector. The results obtained were as follows

Type	Good	Bad	Percent Correct
Random Value	5	78	6.02%
PCA (Eigenvectors)	57	26	68.67%

1c: Analysis

K	Percent Accuracy
1	14.46%
3	45.78%
5	63.86%
7	74.70%
9	75.90%
11	72.29%
13	77.11%
15	79.52%
17	75.90%
19	56.63%
21	75.90%
23	80.72%

While changing the values of k we can see that the resulting accuracy is impacted. At low K values we see very low accuracy in our predictions. As K increases our accuracy increases, but tends to stabilize around 75 – 80% range for accuracy.

1c: Analysis

P	Accuracy %
0.1	38.93%
0.3	75.86%
0.5	80.72%
0.7	84.00%
0.9	88.24%

As we change the percent of data that is allowed to be trained on, we can also see that our accuracy increases. For this method, I left k=15 (Based on previous results), and enumerated over new values of P. Our most accurate result was when p = 0.9. It is important to note the huge increase in accuracy when we went from 0.1 to 0.3 as well. This might indicate that although we increase our training set size, the increase in accuracy falls off to not be as drastic as the size of our training data increases.

2a: Average accuracy

Training Result Set

Type	1	2	3	4	5	Average
Random	48.51%	48.36%	51.33%	49.92%	52.74%	**50.172%**
Weak	87.48%	87.32%	86.70%	87.48%	86.23%	**87.042%**
Boosting	92.49%	91.55%	93.74%	94.21%	92.64%	**92.926%**

Testing Result Set

Type	1	2	3	4	5	Average
Random	52.50%	50.62%	50%	48.75%	51.88%	**50.75%**
Weak	86.25%	88.75%	89.38%	86.25%	90.00%	**88.126%**
Boosting	92.50%	89.38%	94.38%	92.50%	95.00%	**92.752%**

2a: Analysis

We can see that our random classifier is by far the worst method achieving an accuracy of 50.75% on average on our testing set. Our boosting model is the most accurate achieving an average accuracy of over 92%. This isn't vastly superior to our Weak classifier which obtains an average accuracy of 88%.

2a: Analysis

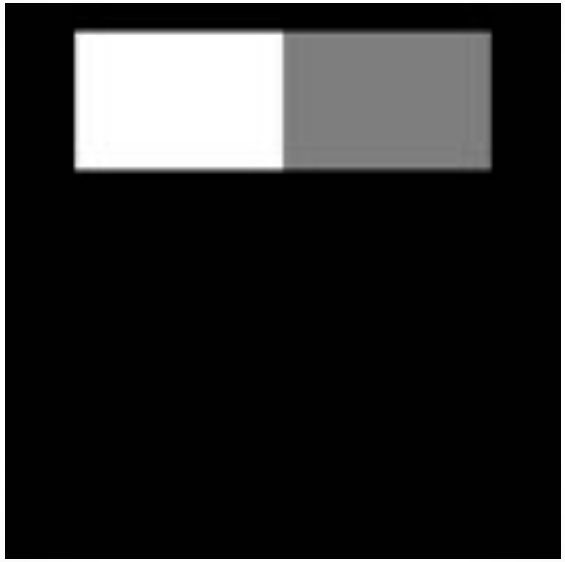
Iterations	Training	Testing
1	86.85%	88.75%
3	90.45%	90.00%
5	93.58%	94.38%
7	97.50%	96.88%
9	99.37%	97.50%

Different values of our *num_iterations* causes the boosting method to increase in accuracy as expected. The large number of Iterations we allow to happen we can see that our accuracy increases vastly. I iterated for up to 10 iterations and found that the best case was using a value of 9, achieving an accuracy of over 99% on the training set and an accuracy of 97.5% on the testing data.

3a: Haar Features

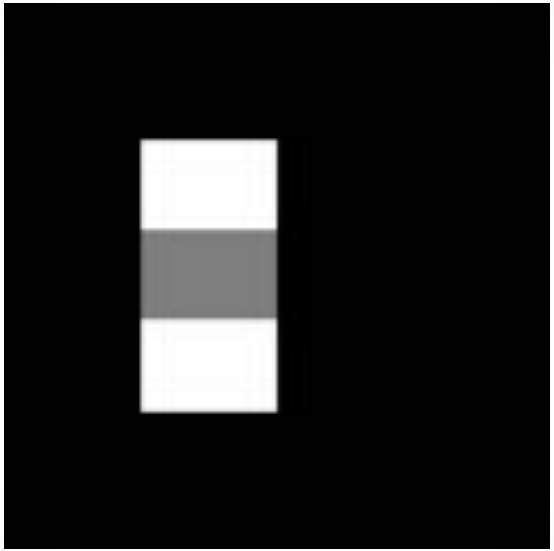
ps6-3-a-1.png

3a: Haar Features



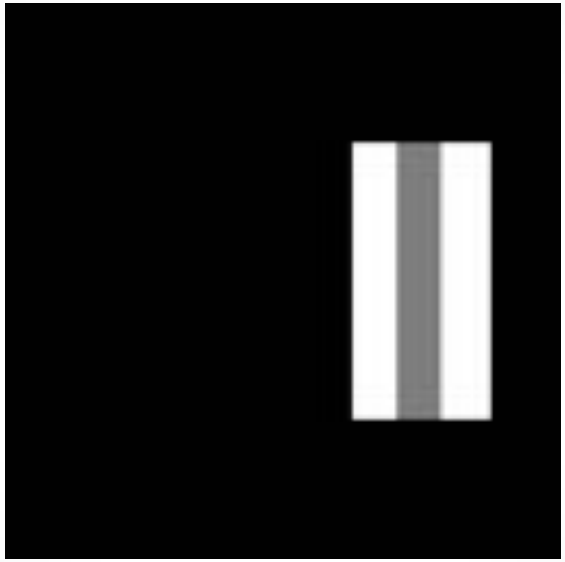
ps6-3-a-2.png

3a: Haar Features



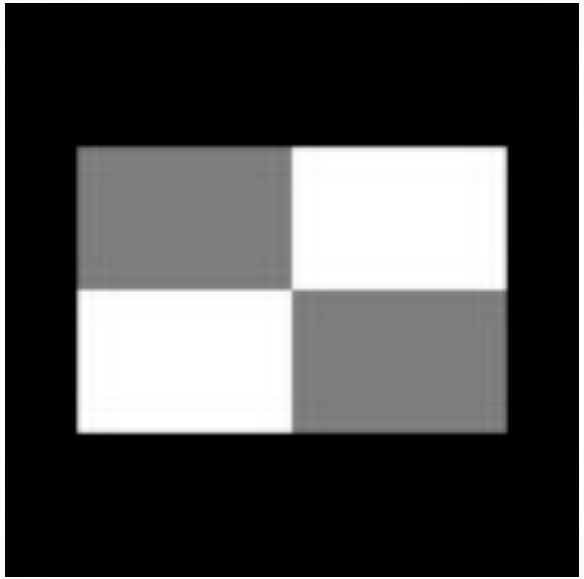
ps6-3-a-3.png

3a: Haar Features



ps6-3-a-4.png

3a: Haar Features



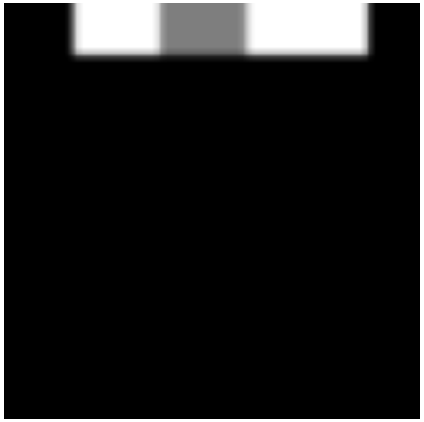
ps6-3-a-5.png

3c: Analysis

Working with integral images improves our computation time by an order of magnitude so it is worth using. The difference in computation is negligible when dealing with small sets, but once we get to a size of over 10,000 iterations , a noticable change can be observed. When using np.sum(), the total time was roughly 2.116 seconds, but when using integral images, we can see the average time decrease to roughly 1 second.

This can be attributed to the fact that when using np.sum() the values need to be recalculated every time whereas the use of integral images allows the values to be calculated once and reused for all iterations.

4b: Viola Jones Features



ps6-4-b-1.png

4b: Viola Jones Features

ps6-4-b-2.png

4b: Analysis

Type	Accuracy
Training	80.00%
Testing	82.86%

As indicated by the table above, there is a training accuracy of 80% and testing accuracy of 82.86%. I attempted to achieve higher training accuracy, but in doing so I would loose testing accuracy due to possible over fitting.

4b: Analysis

The selected haar features represent the best haar features that were used to find the properties of a face. They have the lowest error rate for classification and are deemed the best to classify face and non-face images. By using these selected features, a lot of irrelevant properties of a face are eliminated thus increasing the accuracy of our classifiers.

4c: Viola Jones Face Recognition

ps6-4-c-1.png