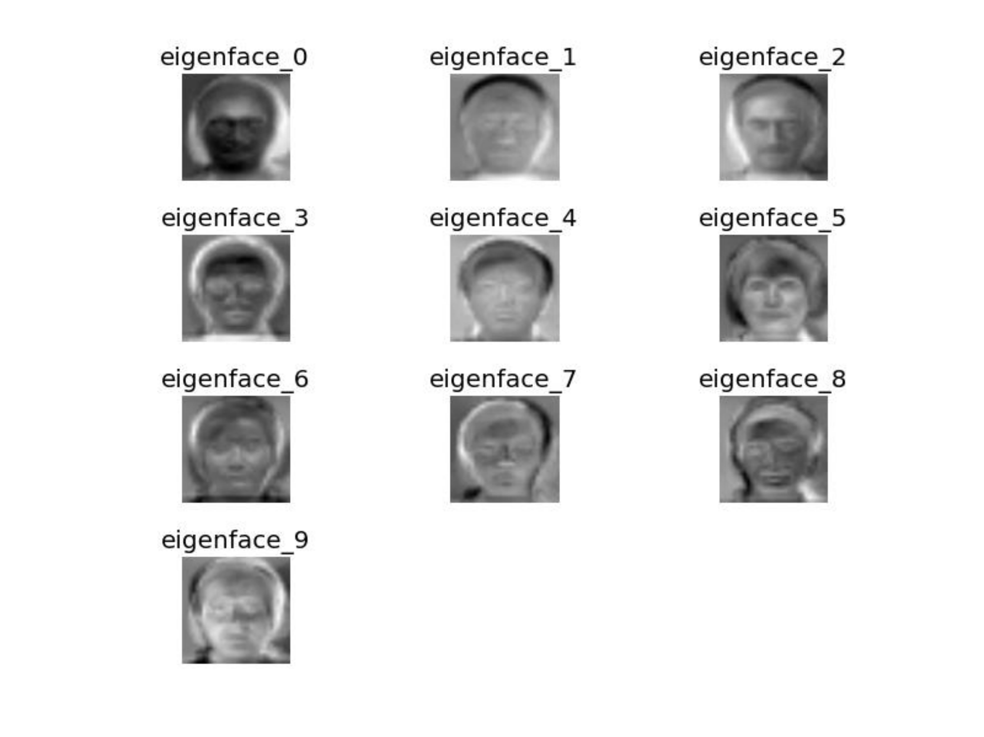
Computer Vision
Fall 2017
Problem Set #6

Zhi Zhang
zhizhang@gatech.edu

1a: Average face

ps6-1-a-1.png

1b: Eigenvectors



ps6-1-b-1.png

1c: Analysis

Do these “predictions” perform better than randomly selecting a label between 1 and 15?

Yes, they perform better than randomly selecting. Below is the result using “rand_y = np.random.randint(low=1, high=16, size=len(ytest))” compared to using engivectors. The p is 0.5, and k = 8.

Random:(Random) Testing accuracy: 4.82%

PCA: Good predictions = 65 Bad predictions = 18

78.31% accuracy

1c: Analysis

K	Accuracy
1	27.71%
3	33.73%
5	67.47%
8	78.31%
10	71.08%
15	73.49%

Are there any changes in accuracy if you try low values of k? How about high values? Yes, it changes. Left is the results for different k holding P as 0.5. As shown, lower k gives worse accuracy, and larger k gives better until to a point where it will converge.

1c: Analysis

K	P	Accuracy
8	0.1	47.65%
8	0.2	56.82%
8	0.3	78.45%
8	0.5	78.31%
8	0.7	70.00%
8	0.8	75.76%
8	0.9	82.35%

Does this algorithm improve changing the split percentage p?

Yes, it changes. Left is the results for different P at same K. As shown, we don't want P to be too small because too little training samples, but, at certain point, the training sample is enough, then the accuracy will not change too much.

2a: Average accuracy

Report the average accuracy over 5 iterations. In each iteration, load and split the dataset, instantiate a Boosting object and obtain its accuracy. Below result using the P as 0.8, and num_iterations as 10

		1	2	3	4	5	Average
Training	Random	48.98%	48.67%	49.45%	51.96%	48.83%	49.58%
	Weak	87.32%	87.95%	87.95%	89.05%	87.32%	87.92%
	Boosting	98.28%	97.81%	98.90%	99.06%	99.22%	98.65%
Testing	Random	48.75%	51.88%	55.00%	56.25%	43.12%	51.00%
	Weak	89.38%	84.38%	86.88%	80.00%	86.88%	85.50%
	Boosting	99.38%	98.12%	98.75%	95.62%	97.50%	97.87%

2a: Analysis

Analyze your results. How do the Random, Weak Classifier, and Boosting perform? Is there any improvement when using Boosting?

As shown below, using P as 0.8, and num_iterations as 10, the random just guess half percent correctly, the weak classifier is doing better, but the boosting can achieve even better.

Methods	Testing Accuracy
Random	51.00%
Weak Classifier	85.50%
Boosting	97.87%

2a: Analysis

How do your results change when selecting different values for num_iterations?

As shown below, holding the P as 0.8, as the number of iterations increase, the random and weak classifier results keep relative stable, but the boosting can get better results.

Iterations	Random Testing Accuracy	Weak Testing Accuracy	Boosting Testing Accuracy
5	49.38%	87.50%	90.62%
8	41.25%	90.62%	95.00%
10	49.38%	89.38%	96.25%
15	49.38%	88.75%	97.50%
20	55.62%	89.38%	100.00%

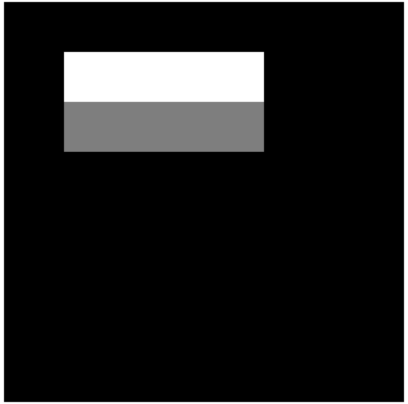
2a: Analysis

Does it matter the percentage of data you select for training and testing (explain your answers showing how each accuracy changes).?

As shown below, holding the num_iterations as 20, as the P increases, the random and weak classifier results are up and down, but the boosting can get better results, and will converge eventually.

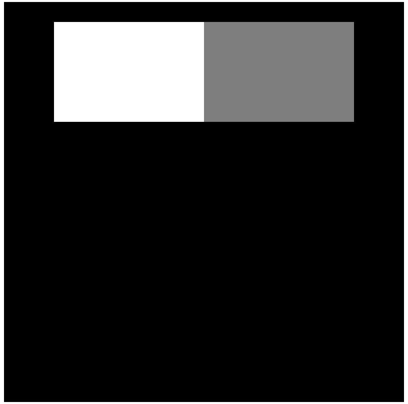
P	Random Testing Accuracy	Weak Testing Accuracy	Boosting Testing Accuracy
0.2	53.12%	87.19%	97.03%
0.35	46.35%	84.81%	97.50%
0.5	50.25%	87.75%	97.75%
0.7	53.33%	90.00%	98.33%
0.9	55.00%	88.75%	100.00%

3a: Haar Features



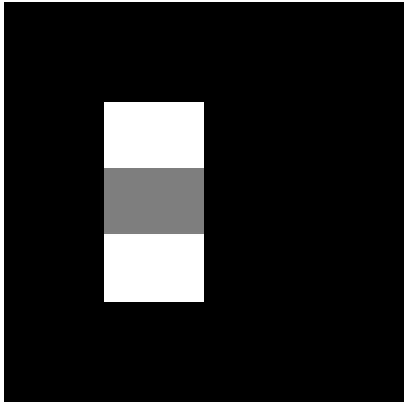
ps6-3-a-1.png

3a: Haar Features



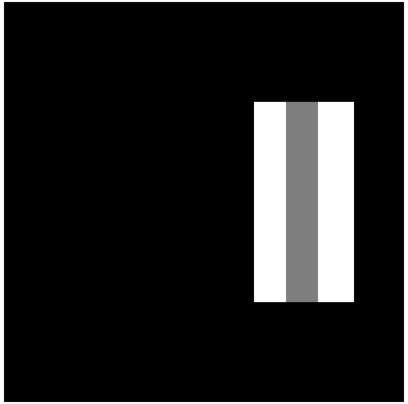
ps6-3-a-2.png

3a: Haar Features



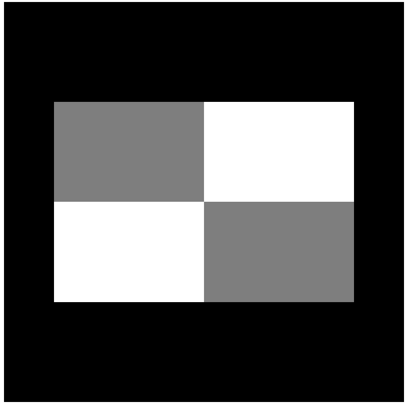
ps6-3-a-3.png

3a: Haar Features



ps6-3-a-4.png

3a: Haar Features



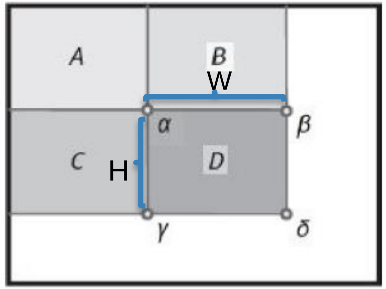
ps6-3-a-5.png

3c: Analysis

How does working with integral images help with computation time? Give some examples comparing this method and np.sum.

Integral image can improve the computing efficiency because once the summed-area table has been computed, any sum of intensities over any rectangular area can be calculated by means of exactly four array references regardless of the area size. In the following example, the sum of rectangle D is $D_{sum} = I(\delta) + I(\alpha) - [I(\beta) + I(\gamma)]$

If using np.sum, it needs sum $W \times H$ times, also, in the feature calculation, each feature will need a sum of pixels under white and black rectangles, the number of features is very big, so the computation time will significantly incresed if without working with integral images.



4b: Viola Jones Features

ps6-4-b-1.png

4b: Viola Jones Features

ps6-4-b-2.png

4b: Analysis

Report the classifier accuracy both the training and test sets with a number of classifiers set to 5. What do the selected Haar features mean? How do they contribute in identifying faces in an image?

Prediction accuracy on training: 95.71%

Prediction accuracy on testing: 65.71%

The selected Haar features mean the most relevant features that can represent the properties of the face. They have the minimum error rate for classification, and they are the features that best classify the face and non-face images. The two features selected relies on the property that the bridge of the nose is darker than the eyes. By using these selected features, a lot of irrelevant features are eliminated, which increased the prediction accuracy and computation efficiency.

4c: Viola Jones Face Recognition

ps6-4-c-1.png