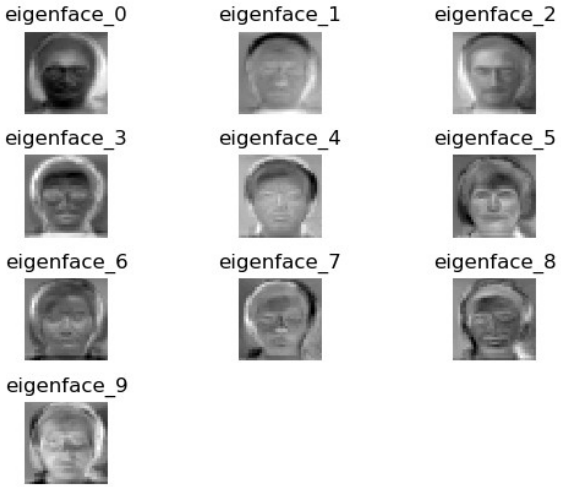
Computer Vision (Spring 2019) Problem Set #6

Jijun HU
jijun.hu.0930@gatech.edu

1a: Average face

ps6-1-a-1

1b: Eigenvectors



ps6-1-b-1

1c: Analysis

Analyze the accuracy results over multiple iterations. Do these “predictions” perform better than randomly selecting a label between 1 and 15? Are there any changes in accuracy if you try low values of k? How about high values? Does this algorithm improve changing the split percentage p?

First, Yes, the predictions perm better than randomly selecting, In my tests, my random selecting only gives around 5% accuracy while prediction with p = 0.5 and K = 9 gives around 72.3%

Second, k does impact the prediction accuracy, by controlling p = 0.5, here is the result, seems accuracy increases when k increases, but the marginal incremental becomes smaller and smaller

k=5	k=9	k=11	k=15	k=25
63.46%	73.55%	75.17%	76.43%	78.42%

Third, p also does impact the prediction accuracy, by controlling k = 11, here is the result, the accuracy will increase when p increases. But when training sample is enough, accuracy won’t increase even we have more training data.

p=0.5	p=0.6	p=0.7	p=0.8	p=0.9
75.17%	75.74%	76.24%	77.85%	76.35%

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.

Report the average accuracy over 5 iterations. In each iteration, load and split the data sets, instantiate a Boosting object and obtain its accuracy. Below result using the P as 0.8, and num_iterations as 15

		Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Average
Training	Random Choice	49.30%	49.92%	46.95%	49.14%	49.61%	48.99%
	Weak Learner	87.48%	86.85%	87.95%	86.70%	87.32%	87.26%
	Boosting	95.15%	97.50%	94.84%	98.44%	96.56%	96.50%
Testing	Random Choice	51.88%	45.00%	48.12%	48.12%	51.88%	49.00%
	Weak Learner	86.25%	88.75%	84.38%	89.38%	89.38%	87.63%
	Boosting	98.12%	95.00%	93.12%	99.38%	95.00%	96.12%

2a: Analysis

Analyze your results. How do the Random, Weak Classifier, and Boosting perform? Is there any improvement when using Boosting? How do your results change when selecting different values for num_iterations? Does it matter the percentage of data you select for training and testing (explain your answers showing how each accuracy changes).

As shown in the previous table, by assigning num_iter = 15 and p = 0.8, we can conclude -

Boosting perform better than weak learner, which is also better than random assignment.

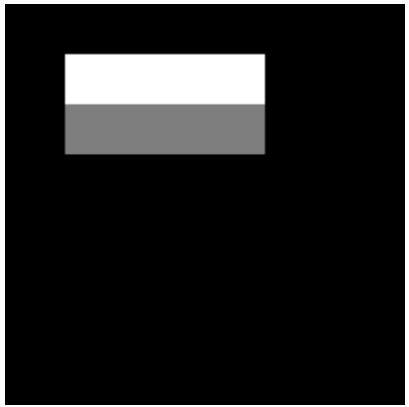
In terms of values of num_iterations, repeating the same work, we can see from the table below, accuracy will improve when num_iteration increases, by since the accuracy for boosting is already pretty good, the marginal increase is very limit.

num_Iterations	Random Testing Accuracy	Weak Testing Accuracy	Boosting Testing Accuracy
8	41.25%	90.62%	95.00%
10	48.38%	86.38%	95.89%
15	49.00%	87.63%	96.12%

Similar Analysis for different p, but controlling num_iterations, p helps improve accuracy but impact is very limited given already high accuracy

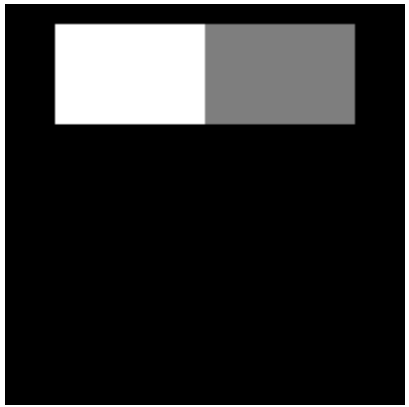
P	Random Testing Accuracy	Weak Testing Accuracy	Boosting Testing Accuracy
0.35	46.35%	84.81%	97.50%
0.5	49.00%	87.63%	96.12%
0.7	53.33%	90.00%	98.33%

3a: Haar Features



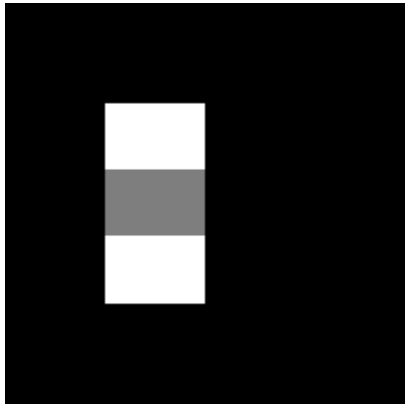
ps6-3-a-1

3a: Haar Features



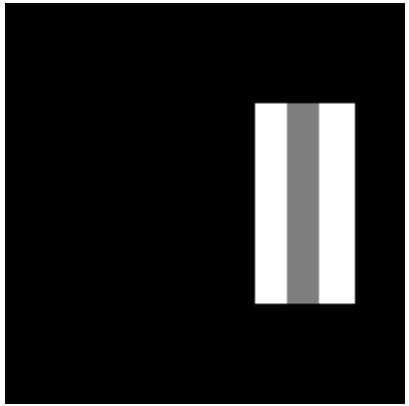
ps6-3-a-2

3a: Haar Features



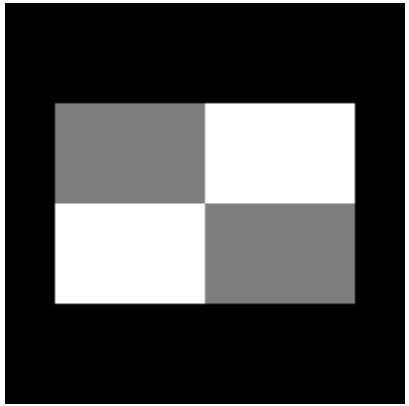
ps6-3-a-3

3a: Haar Features



ps6-3-a-4

3a: Haar Features



ps6-3-a-5

3c: Analysis

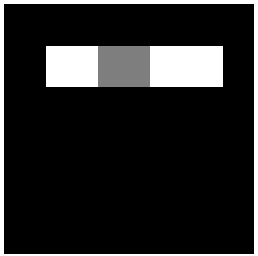
How does working with integral images help with computation time? Give some examples comparing this method and np.sum.

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.

You do only have to calculate the whole image once to get the accumulated sum value, later no matter how may area you want to calculate, each area only takes 4 * 4 single calculation (+/-) to get the number.

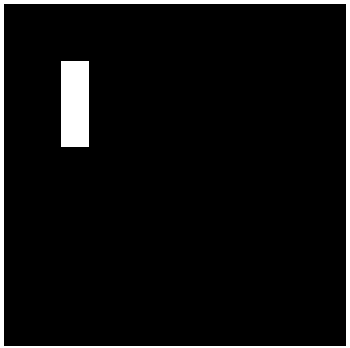
When using np.sum, basically every time you want to calculate an area, you have to subset that area and aggregate all pixel numbers in that area, suppose our area is a * b, then we need a*b calculation the derive the total pixel numbers for just only one area. If we have n similar multiple pictures to calculate, we have to compute n * a * b times, while integral images calculation only takes much less operations to get the exact same results.

4b: Viola Jones Features



ps6-4-b-1

4b: Viola Jones Features



ps6-4-b-2

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?

Num. Of Classifier	Trainning Accuracy	Testing Accuracy
5	100.00%	48.57%
10	98.57%	68.57%
15	98.57%	82.86%

The top 2 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 1st selected feature relies on the property that the bridge of the nose is darker than the eyes, while the 2nd selected feature is more like an indicator of when the face(edge of the face) usually comes into the image.
By using these selected features, we can discard lots of irrelevant information and noise, thus to improve the computational speed and accuracy.

4c: Viola Jones Face Recognition

ps6-4-c-1