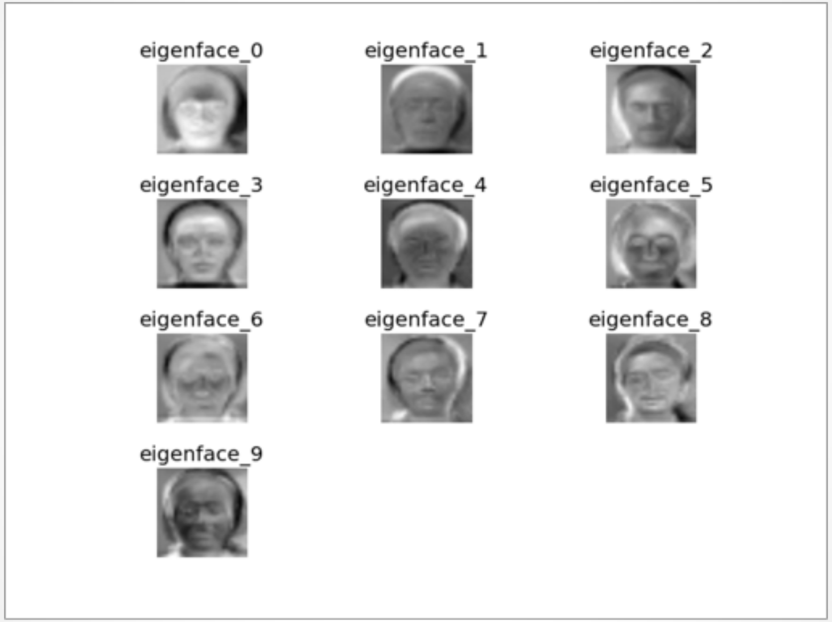
Computer Vision
Fall 2018
Problem Set #6

Junle Lu
junle.lu@gatech.edu

1a: Average face

ps6-1-a-1.png

1b: Eigenvectors



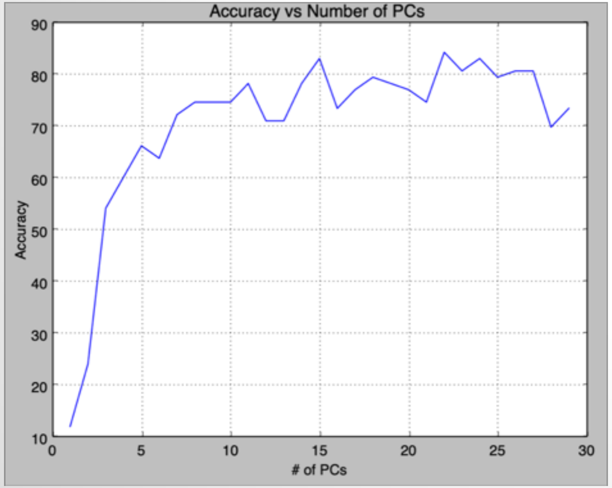
ps6-1-b-1.png

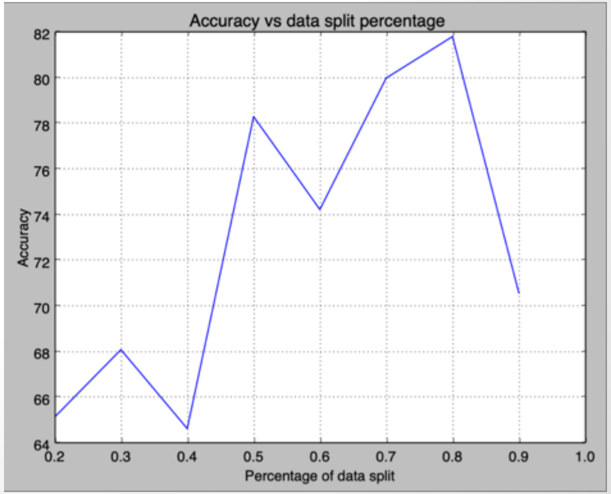
1c: Analysis

I observed the 'predictions' are better than randomly selecting a label between 1 and 15

K value: I ran the experiment for k values from 1-30 and p = 50. From the figure below, it generated the expected curve for the prediction accuracy vs the number of PCs (k values). The first 10 PCs has the most significant contribution to the accuracy. By increasing the PCs furthermore, it won't improve the accuracy much. Therefore, the dimensionality can be reduced to 10 PCs.

Data split percentage: I ran the experiment with fixed k value (10), and iterates through p from 0.2 to 0.9. The result is expected as the 0.7-0.8 yields the best accuracy. This aligns with common Machine Learning practice when splits the data into training and testing sets.





2a: Average accuracy

20 iterations, data split = 0.8:

('(Random) Training accuracy', '48.97 %')

('(Weak) Training accuracy', '87.61 %')

('(Boosting) Training accuracy', '91.49 %')

('(Random) Testing accuracy:', '47.13 %')

('(Weak) Testing accuracy', '86.35 %')

('(Boosting) Testing accuracy', '91.11 %')

20 iterations, data split = 0.5:

('(Random) Training accuracy', '48.81 %')

('(Weak) Training accuracy', '88.04 %')

('(Boosting) Training accuracy', '94.21 %')

('(Random) Testing accuracy:', '52.11 %')

('(Weak) Testing accuracy', '86.32 %')

('(Boosting) Testing accuracy', '89.22 %')

5 iterations, data split = 0.8:

('(Random) Training accuracy', '47.15 %')
('(Weak) Training accuracy', '82.38 %')
('(Boosting) Training accuracy', '86.83 %')
('(Random) Testing accuracy:', '47.42 %')
('(Weak) Testing accuracy', '79.96 %')
('(Boosting) Testing accuracy', '81.99 %')

2a: Analysis

As expected, the random classifier has the worst accuracy. Weak classifier is better than random, and the Boosting classifier has slightly better accuracy than Weak classifier. The change in data split percentage does not seem to affect much in this case. Random classifier produces consistent result regardless the number of iteration and data split percentage. For the other two classifier, the accuracy improves with increasing number of iterations.

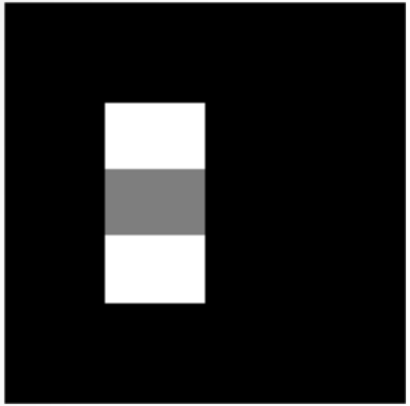
3a: Haar Features

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

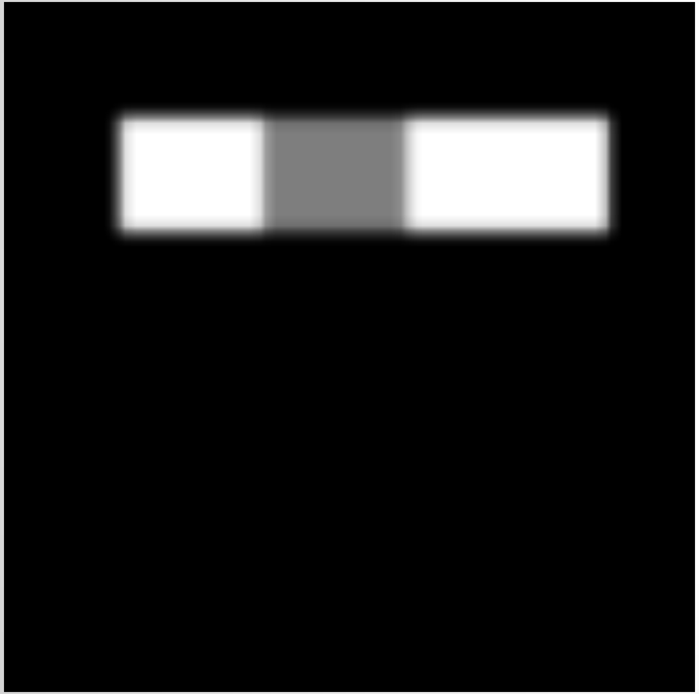
3a: Haar Features

ps6-3-a-5.png

3c: Analysis

With integral images, the computation time is dramatically reduced. With np.sum, the value of each pixel is added each window. The integral image approach allows us to use the previously computed sum to find the sum for the new region of the image. It is much more efficient this way, especially with large of pixels.

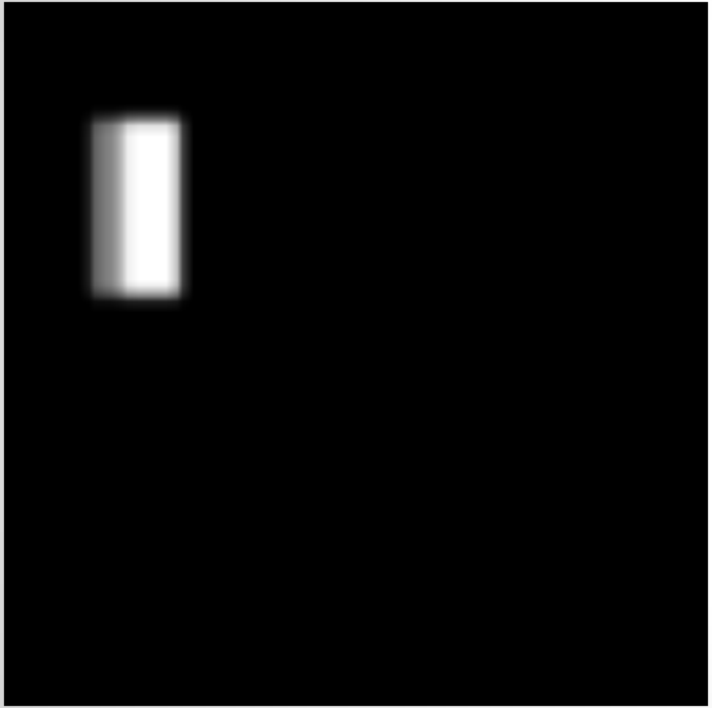
4b: Viola Jones Features



```
Prediction accuracy on training: 95.71%
Prediction accuracy on testing: 85.71%
```

ps6-4-b-1.png

4b: Viola Jones Features



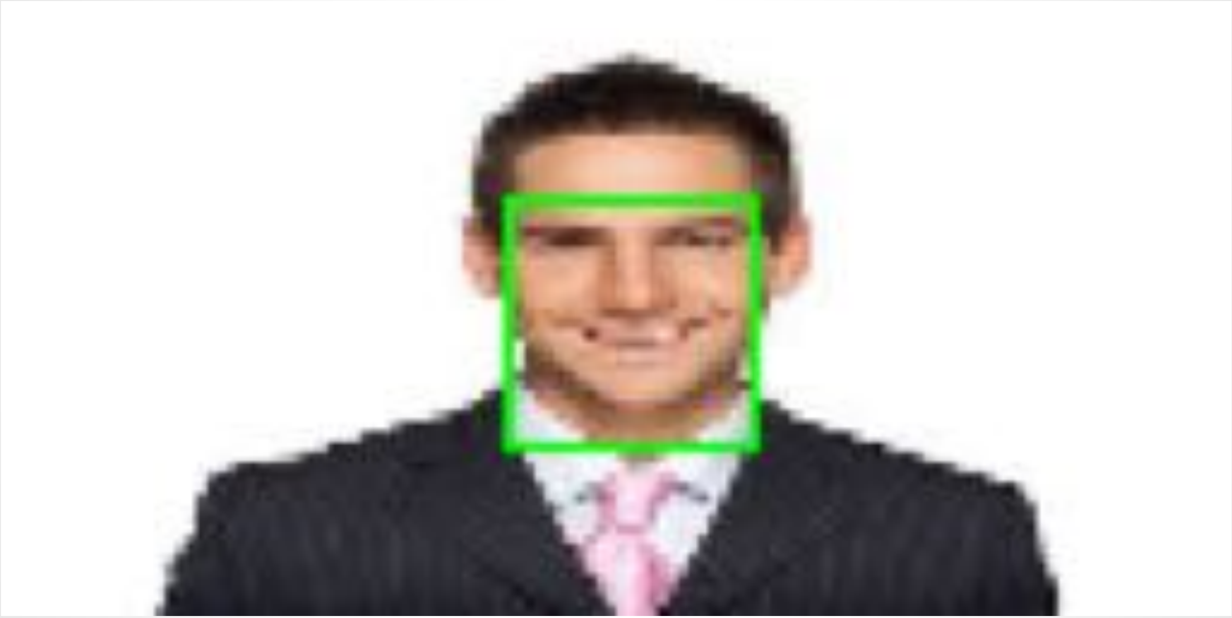
```
Prediction accuracy on training: 95.71%
Prediction accuracy on testing: 85.71%
```

ps6-4-b-2.png

4b: Analysis

The feature ps6-4-b-1 is good for matching eyes and nose. For example, our eyes with dark skin region in between. The second feature is hard to explain how it is used on our facial. It may be due to the training images we used that somehow it thinks that it is a significant feature. Normally, I would expect feature that aligns with our forehead, nose, and eye to be the top features.

4c: Viola Jones Face Recognition



ps6-4-c-1.png