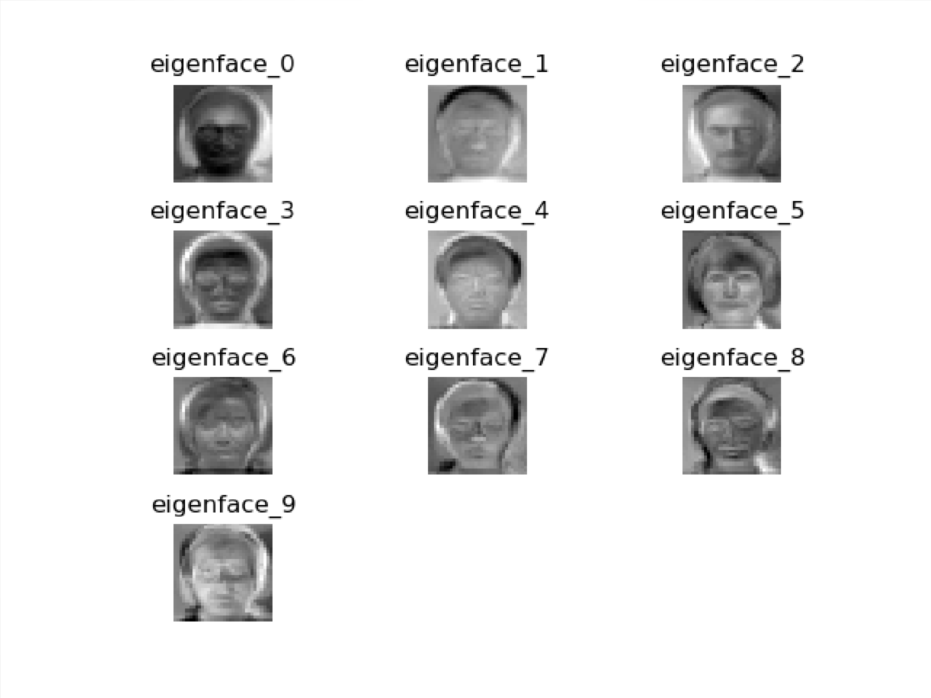
# Computer Vision
# Fall 2017
# Problem Set #6

Sergiy Palguyev
sergiy.palguyev@gatech.edu

# 1a: Average face

**ps6-1-a-1.png**

# 1b: Eigenvectors




**ps6-1-b-1.png**

# 1c: Analysis

The predictions perform better than randomly selecting a label provided a good data set is given.
Lowering k value significantly decreases accuracy.
Increasing k value slight increases accuracy.
Lowering or Increasing p value to extremes will decrease accuracy in both cases.

# 1c: Analysis

Example used for analysis:

Command Prompt

```
Good predictions =  54 Bad predictions =  29
65.06% accuracy p= 0.5 k= 5
Good predictions =  56 Bad predictions =  93
37.58% accuracy p= 0.1 k= 5
Good predictions =  9 Bad predictions =  8
52.94% accuracy p= 0.9 k= 5
Good predictions =  12 Bad predictions =  71
14.46% accuracy p= 0.5 k= 1
Good predictions =  62 Bad predictions =  21
74.70% accuracy p= 0.5 k= 25
```

# 2a: Average accuracy

Approximately 88% for training and 82% for testing:

Command Prompt

```
(Random) Training accuracy: 51.33%
(Weak) Training accuracy 88.26%
(Boosting) Training accuracy 88.26%
(Random) Testing accuracy: 46.88%
(Weak) Testing accuracy 82.50%
(Boosting) Testing accuracy 82.50%
```

# 2a: Analysis

Between Random, Weak, and Boosting, it appears there is not much difference between Weak and Boosting methods.

Slightly lowering p value, increases testing accuracy and slightly the training accuracy.

Dropping p value lower, increases training accuracy and slightly the testing accuracy.

Decreasing number of iterations slightly lowers accuracy for all.

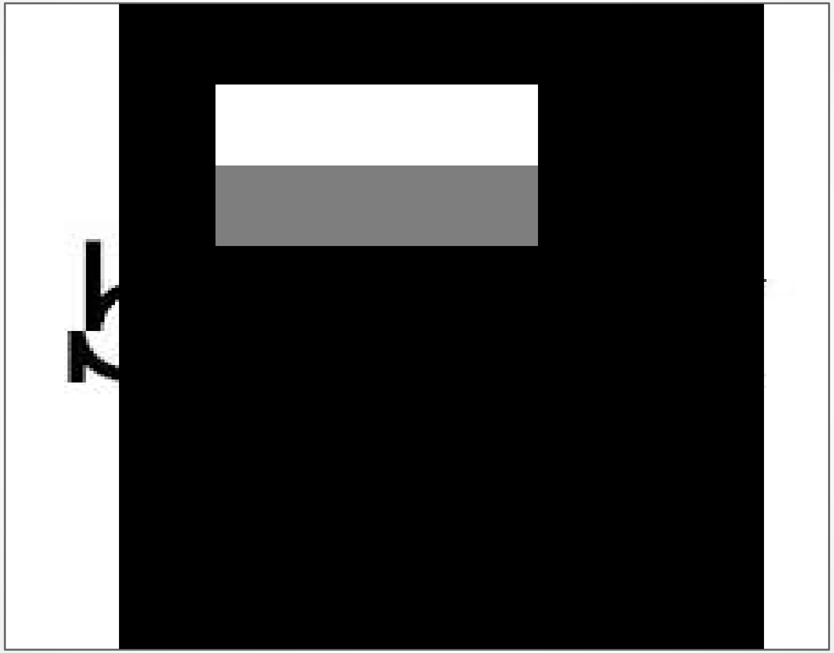
Increasing number of iterations slightly increases accuracy for all.

# 2a: Analysis

Example used for analysis:

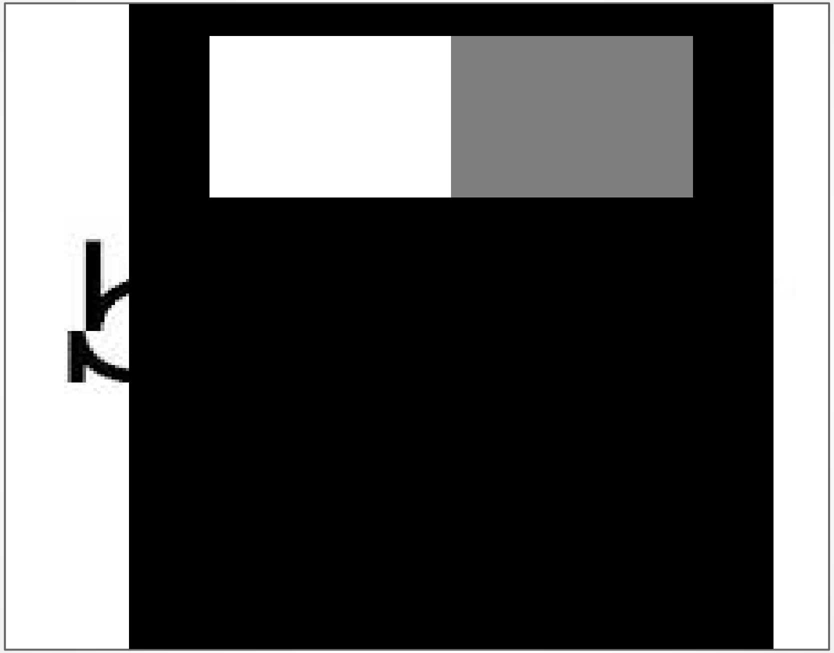
```
(Random) Training accuracy: 48.20%
(Weak) Training accuracy 87.79%
(Boosting) Training accuracy 88.42%
(Random) Testing accuracy: 46.88%
(Weak) Testing accuracy 83.75%
(Boosting) Testing accuracy 83.12%
p= 0.8  num_iter= 5
(Random) Training accuracy: 50.13%
(Weak) Training accuracy 88.22%
(Boosting) Training accuracy 88.22%
(Random) Testing accuracy: 54.00%
(Weak) Testing accuracy 87.75%
(Boosting) Testing accuracy 87.75%
p= 0.5  num_iter= 5
(Random) Training accuracy: 52.20%
(Weak) Training accuracy 91.82%
(Boosting) Training accuracy 91.82%
(Random) Testing accuracy: 50.00%
(Weak) Testing accuracy 85.62%
(Boosting) Testing accuracy 85.62%
p= 0.2  num_iter= 5
(Random) Training accuracy: 47.89%
(Weak) Training accuracy 87.95%
(Boosting) Training accuracy 87.95%
(Random) Testing accuracy: 47.50%
(Weak) Testing accuracy 83.12%
(Boosting) Testing accuracy 83.12%
p= 0.8  num_iter= 1
(Random) Training accuracy: 46.48%
(Weak) Training accuracy 88.26%
(Boosting) Training accuracy 88.26%
(Random) Testing accuracy: 53.12%
(Weak) Testing accuracy 85.62%
(Boosting) Testing accuracy 85.62%
p= 0.8  num_iter= 25
```

# 3a: Haar Features



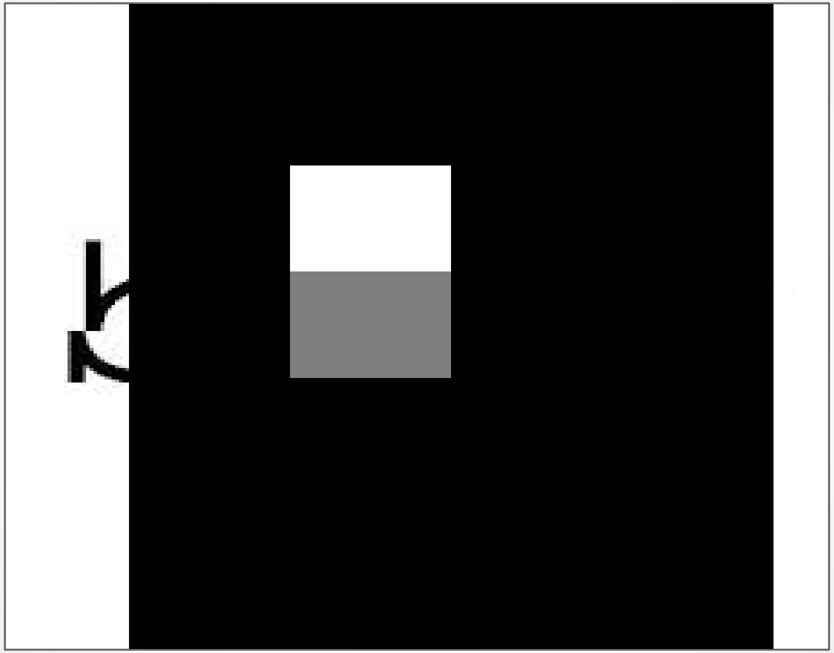
**ps6-3-a-1.png**

# 3a: Haar Features



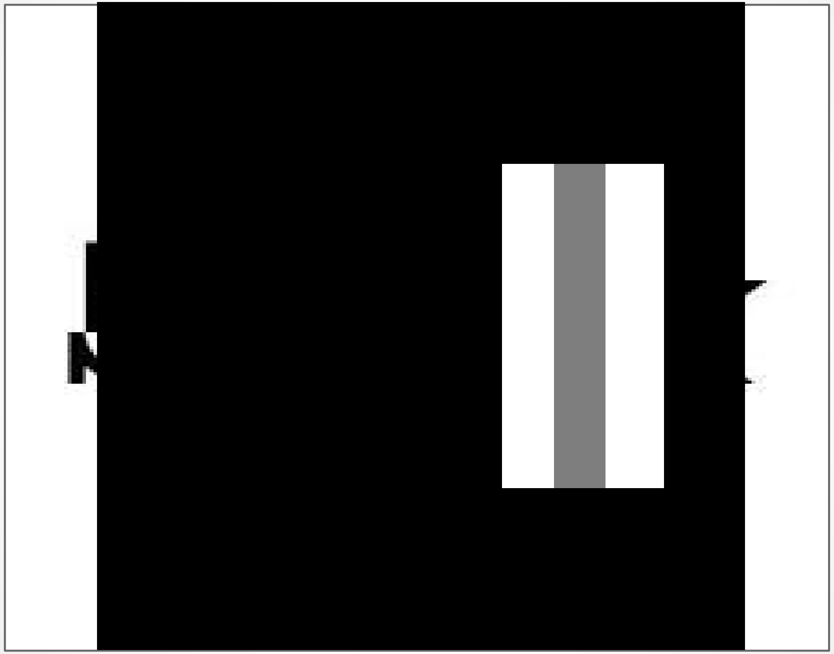
**ps6-3-a-2.png**

# 3a: Haar Features



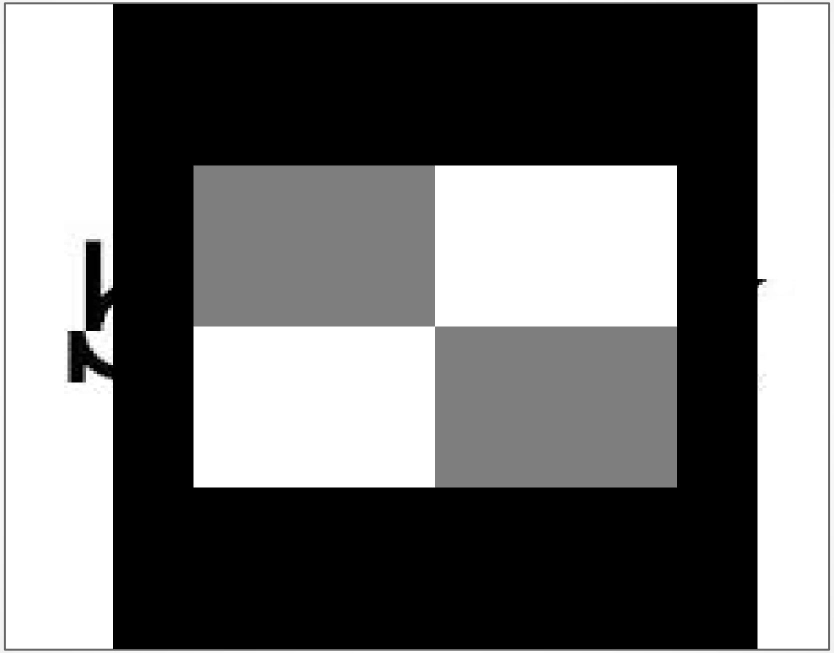
**ps6-3-a-3.png**

# 3a: Haar Features



**ps6-3-a-4.png**

# 3a: Haar Features



**ps6-3-a-5.png**

# 3c: Analysis

Using integral imaging helps by allowing estimations to be made in almost constant time irrelavant of image size, whereas a np.sum function would require quadratic compute time with the growth of image size.

# 4b: Viola Jones Features


**ps6-4-b-1.png**

# 4b: Viola Jones Features


**ps6-4-b-2.png**

# 4b: Analysis

Add more slides if needed.

# 4c: Viola Jones Face Recognition


**ps6-4-c-1.png**