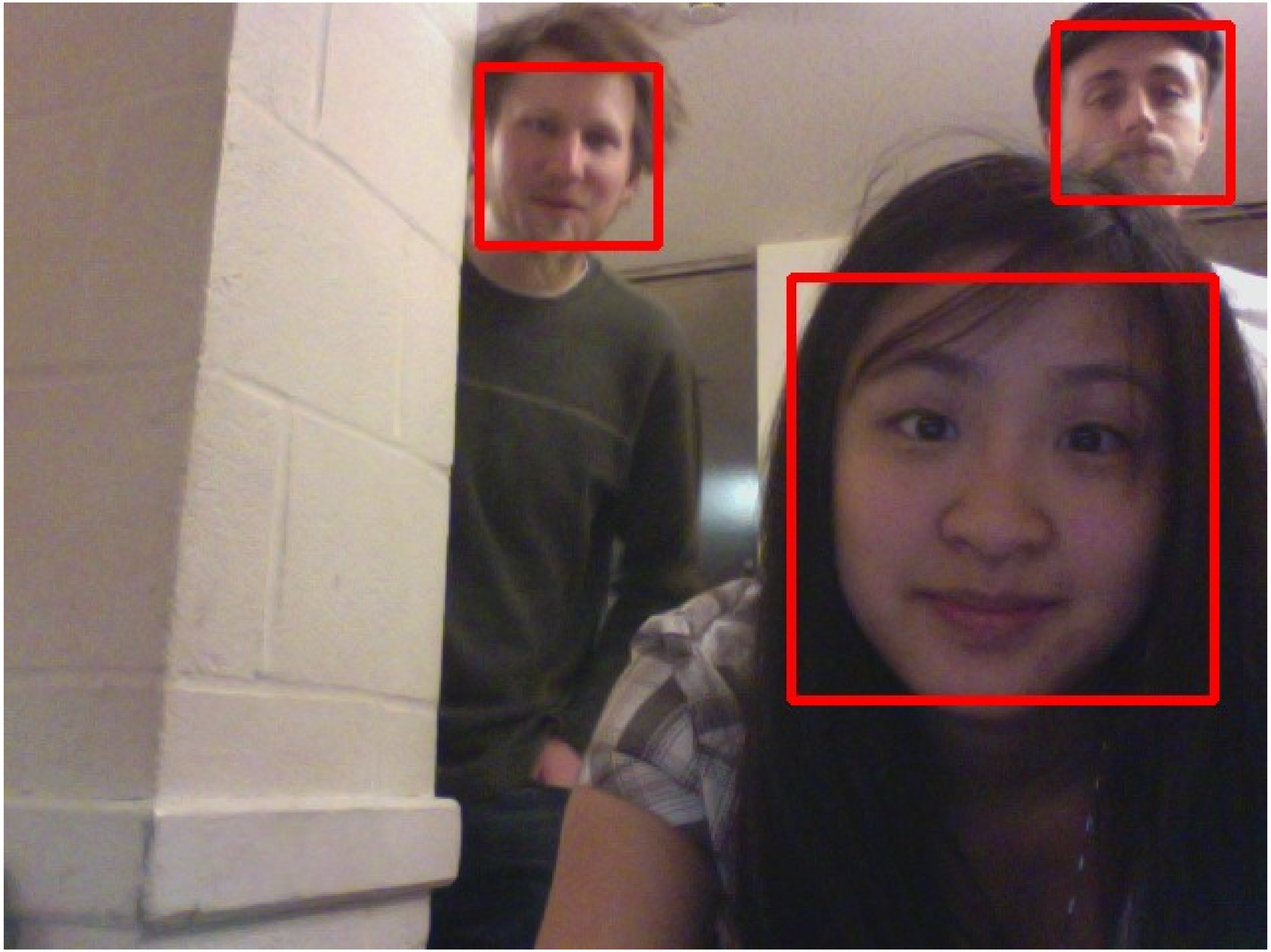
Facial Recognition on Mobile Platforms

Cooper Bills (csb88)
Jason Yosinski (jy495)

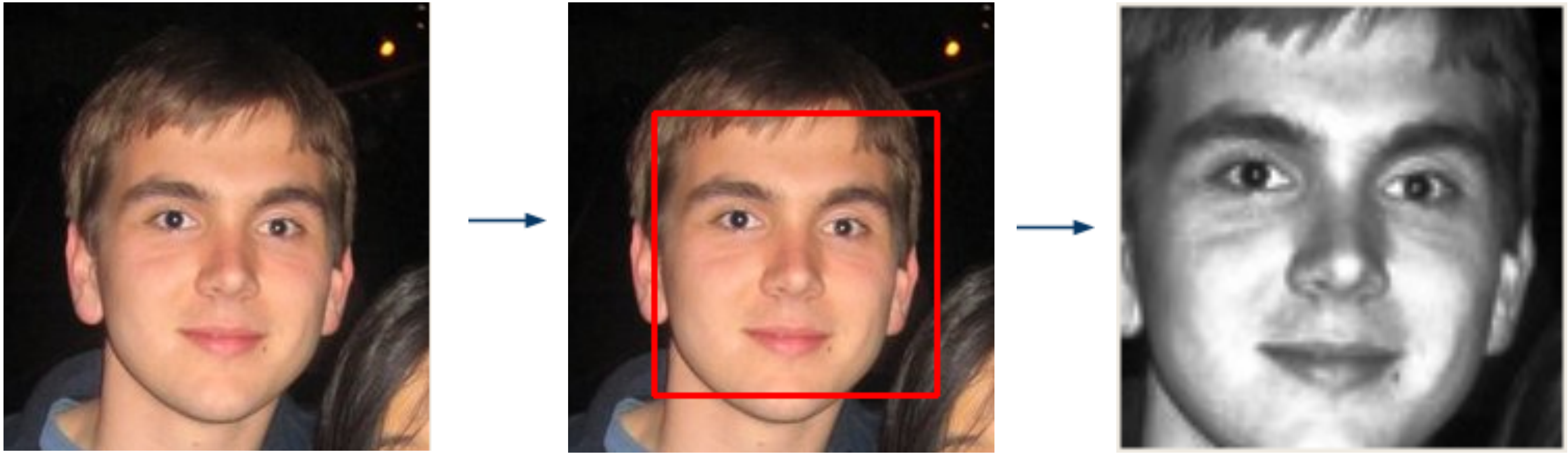
Approach - Face Detection

Harr cascades

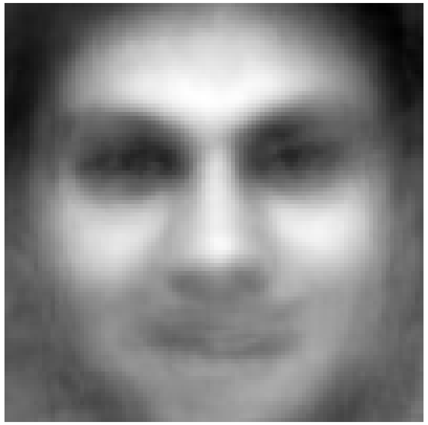


Approach - Database Generation

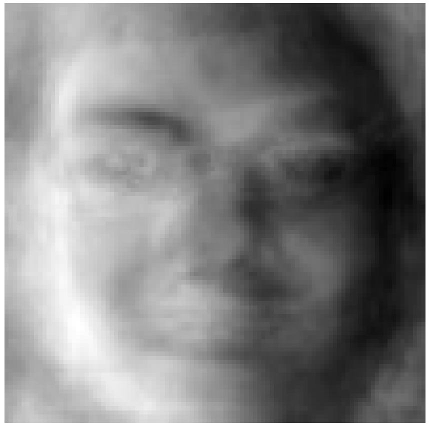
Universally crop and equalize images:



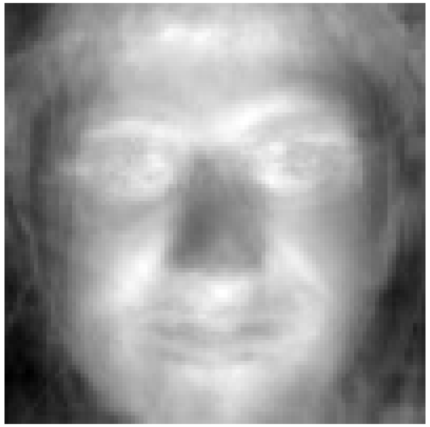
Approach - Eigenfaces



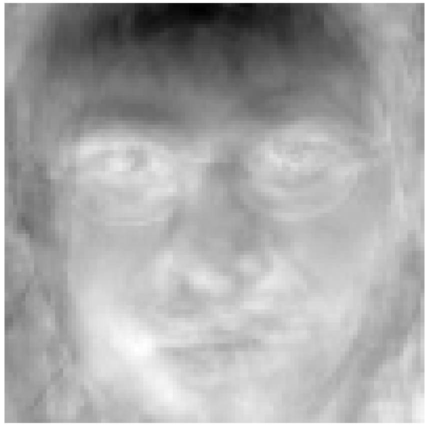
Vector 0 (avg)



Vector 1



Vector 2



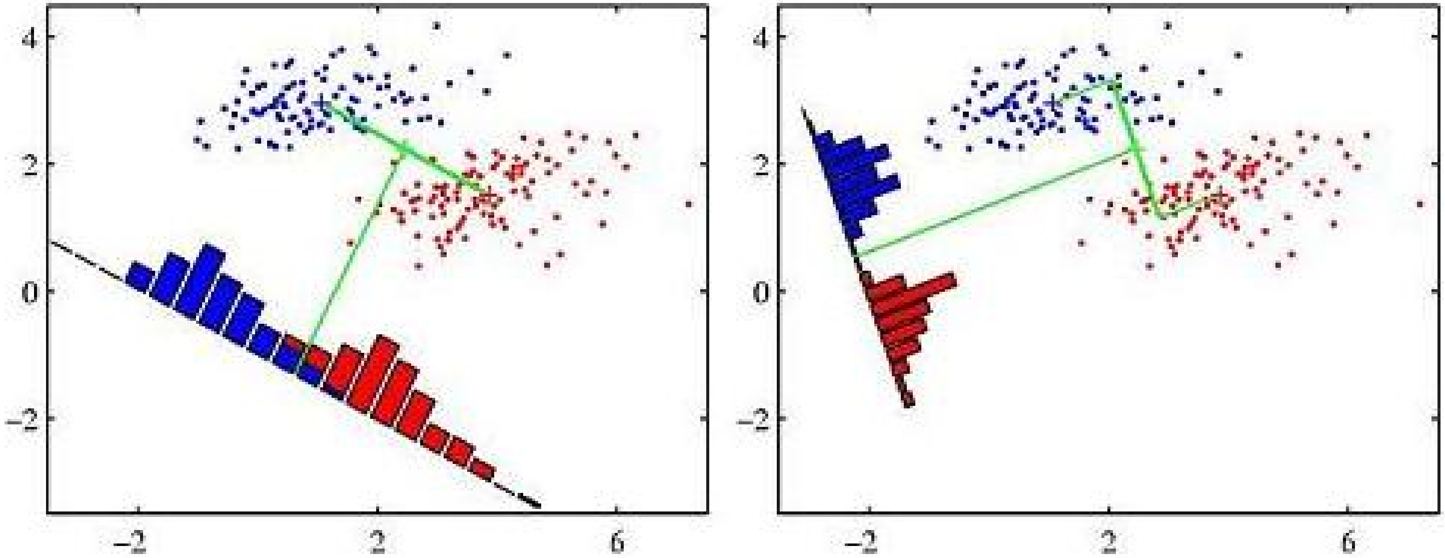
Vector 3

...

Problem with Eigenfaces:

```
$ ../../eigenDecomp Result0.cooper01.jpg      (not in dataset)
Results:
Index Match0:  157, dist:  45990788.000, Label: Result1.subject05.rightlight.jpg
Index Match0:   78, dist:  49232532.000, Label: Result0.subject08.rightlight.jpg
Index Match0:  131, dist:  63325640.000, Label: Result0.subject13.rightlight.jpg
Index Match0:  142, dist:  64011600.000, Label: Result0.subject14.rightlight.jpg
Index Match0:  152, dist:  64910080.000, Label: Result0.subject15.rightlight.jpg
Index Match0:   57, dist:  65682916.000, Label: Result0.subject06.rightlight.jpg
Index Match0:   39, dist:  66772448.000, Label: Result0.subject04.rightlight.jpg
Index Match0:   17, dist:  68427568.000, Label: Result0.subject02.rightlight.jpg
Index Match0:   88, dist:  68480248.000, Label: Result0.subject09.rightlight.jpg
Index Match0:  120, dist:  70358344.000, Label: Result0.subject12.rightlight.jpg
Done.
```

Approach - Fisher Faces



- Find the single dimension that best separates two classes
- Separation defined as:

$$S = \frac{\sigma^2_{between}}{\sigma^2_{within}} = \frac{(\vec{w}\cdot\vec{\mu}_{y=1} - \vec{w}\cdot\vec{\mu}_{y=0})^2}{\vec{w}^T\Sigma_{y=1}\vec{w} + \vec{w}^T\Sigma_{y=0}\vec{w}} = \frac{(\vec{w}\cdot(\vec{\mu}_{y=1} - \vec{\mu}_{y=0}))^2}{\vec{w}^T(\Sigma_{y=0} + \Sigma_{y=1})\vec{w}}$$

- Make *k* separate one-vs-all classifiers

Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Wikipedia

Approach - Fisher Faces

- Tricky to calculate covariances

$$\vec{w} = (\Sigma_{y=0} + \Sigma_{y=1})^{-1}(\vec{\mu}_{y=1} - \vec{\mu}_{y=0})$$

  - Raw images: 10000 x 10000
  - As many dimensions as images: 165 x 165
  - First N eigenvectors: N x N

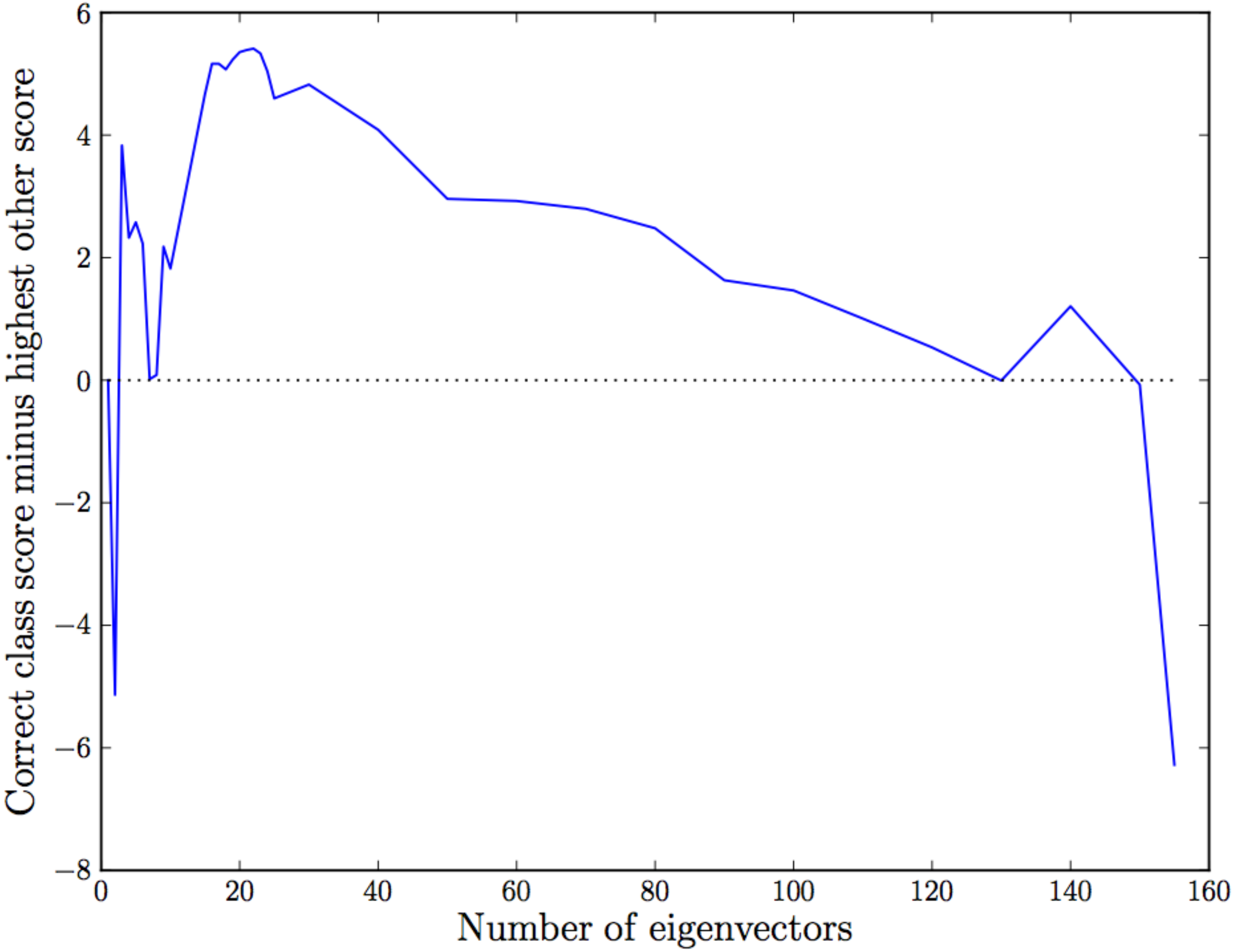
- Regularization

$$\vec{w} = (\Sigma_{y=0} + \Sigma_{y=1} + \epsilon I)^{-1}(\vec{\mu}_{y=1} - \vec{\mu}_{y=0})$$

- Score for each class:

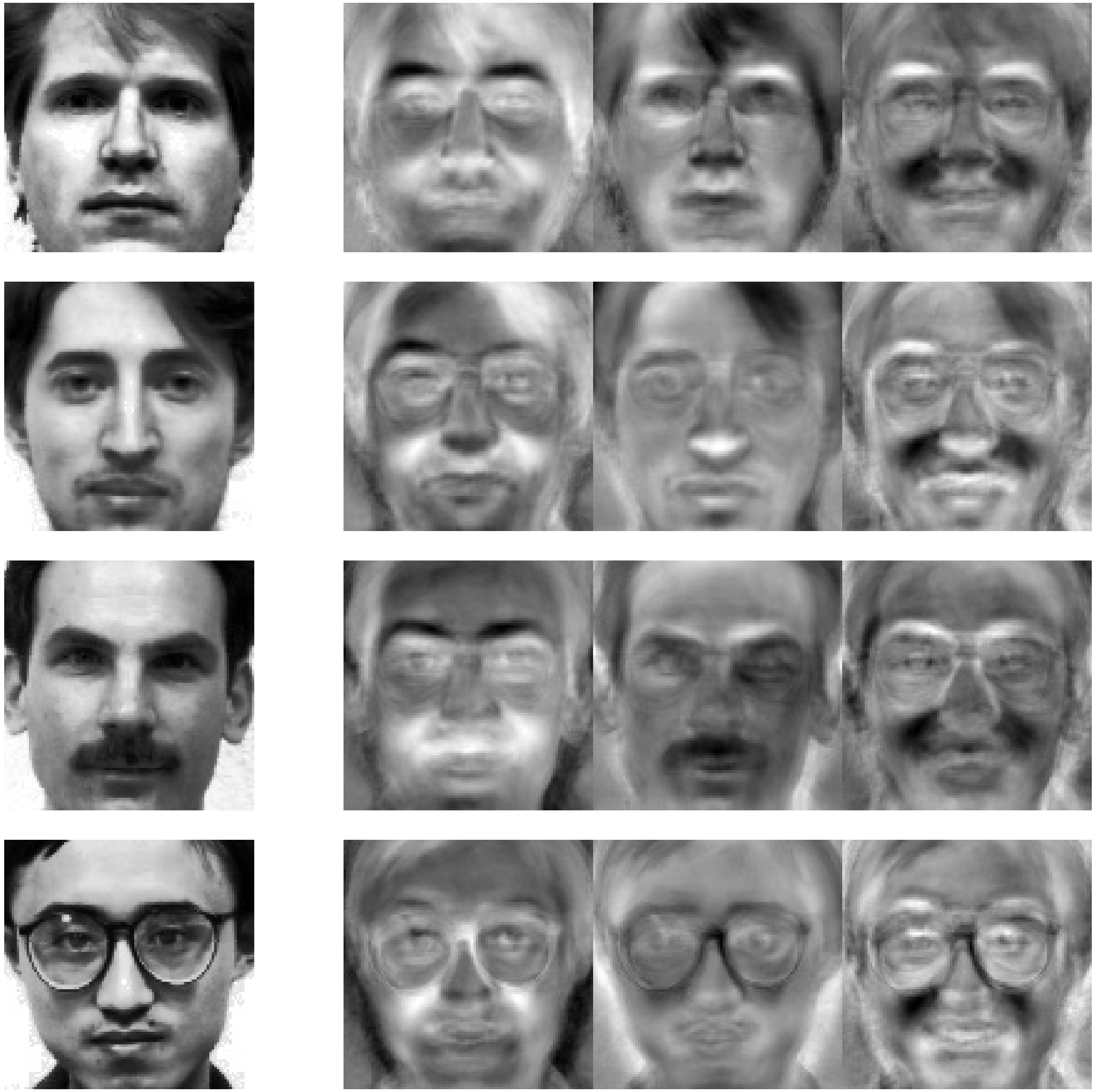
$$s_{ratio} = \frac{s_{test} - s_{\mu_0}}{s_{\mu_1} - s_{\mu_0}}$$

Results - Fisher Faces



Settled on 24 dimensions for rest of work

Results - Fisher Faces



Results - Fisher Faces

Results on Yale Database:

- Perfect recognition of "normal" faces (easier).

- Perfect recognition of "happy" faces (harder).

Demo

Live Demo! (Hopefully)

Questions?

References:

Yale Face Database - Retrieved from: http://cvc.yale.edu/projects/yalefaces/yalefaces.html

Bradski, G. (2000) The OpenCV Library - Dr. Dobb's Journal of Software Tools

Shervin Emami (2010) Introduction to Face Detection and Face Recognition. Retrieved from: http://www.shervinemami.co.cc/faceRecognition.html

M. Turk, A. Pentland, Eigenfaces for Recognition, Journal of Cognitive Neurosicence, Vol. 3, No. 1, 1991, pp. 71-86

Christopher M. Bishop, *Pattern Recognition and Machine Learning*