Face Recognition Using PCA

Slides adapted from Robert Pless and Hongcheng Wang

Global Image Models

- Local models
  - Local image features (SIFT, MOPS)
- Global models
  - Captures overall appearance changes across image
- Today, we are going to focus on correlated changes that occur in the image
  - Using a tool called Principal Components Analysis (PCA)

PCA Idea

- Given: a collection of sample images, $\{I_1, \ldots I_n\}$

- Find: A mean image μ, and a collection of principle components (images) $\{B_1, B_2, \ldots B_k\}$, such that each sample image I_i can be approximated as:

$$I_i \approx \mu + c_1 B_1 + c_2 B_2 + \ldots + c_k B_k$$

- $c_1, c_2, \ldots c_k$ are *coefficients*.
- Each image has different coefficients

PCA

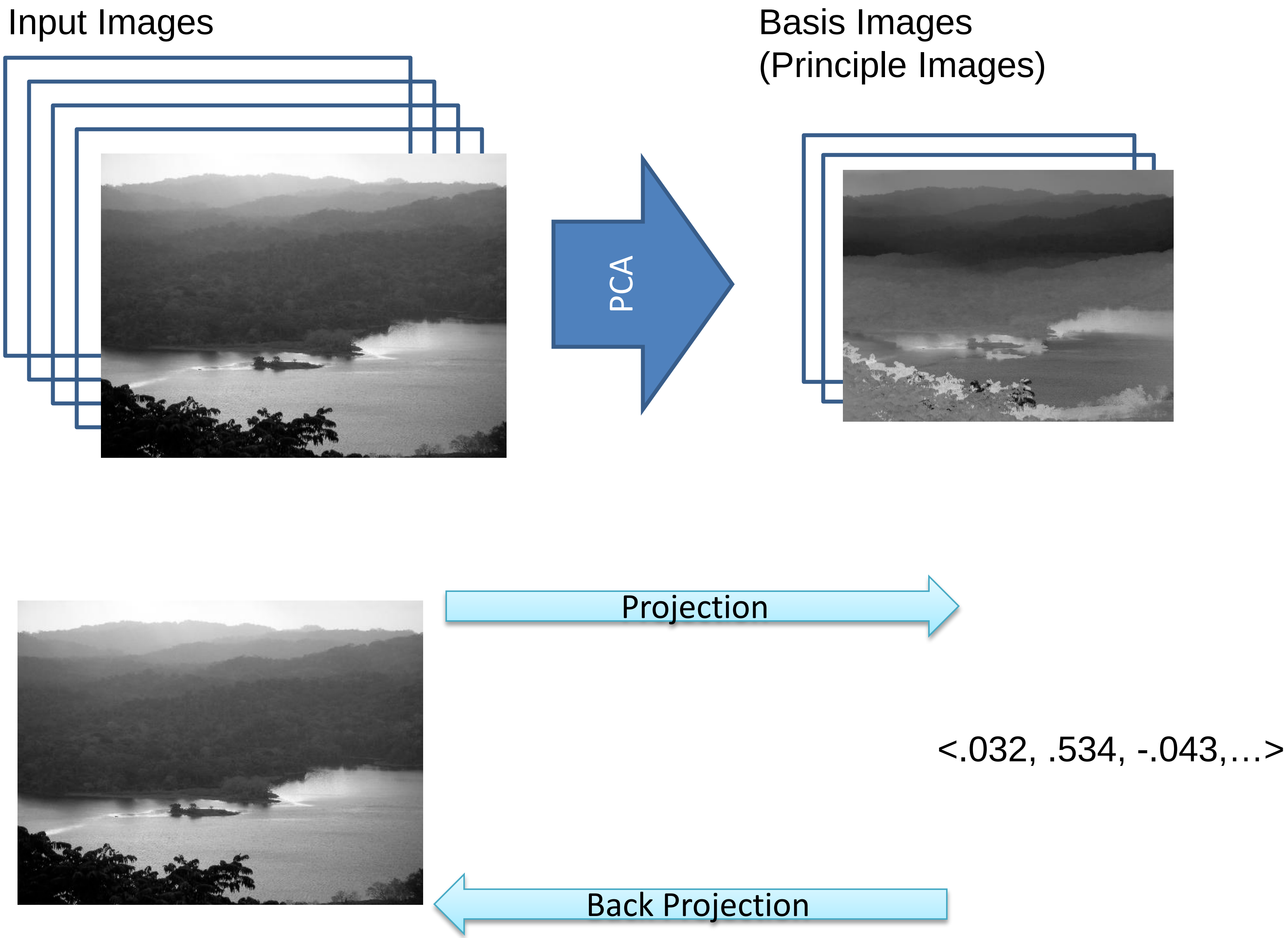
$$I_i \approx \mu + c_1 B_1 + c_2 B_2 + \ldots + c_k B_k$$

- This literally means that I want to recreate I_i by adding other images together.
- There is a "space of images" spanned by a mean image and set of principal components.
- That space of images is formed by choosing all possible combinations of the coefficients c_1, c_2, … c_k

Given a set of images, PCA finds all of the B's and c's

$I_i = \mu + 1\ B_1$

$I_2 = \mu + -1\ B_1$


Input Images
Basis Images
(Principle Images)
PCA
Projection
<.032, .534, -.043,…>
Back Projection

PCA and Faces

- Now, we will see how PCA can be used for face recognition
- Falls under the umbrella of *biometrics*
  - Uniquely recognizing humans based on physical traits
  - Used for security and/or access control

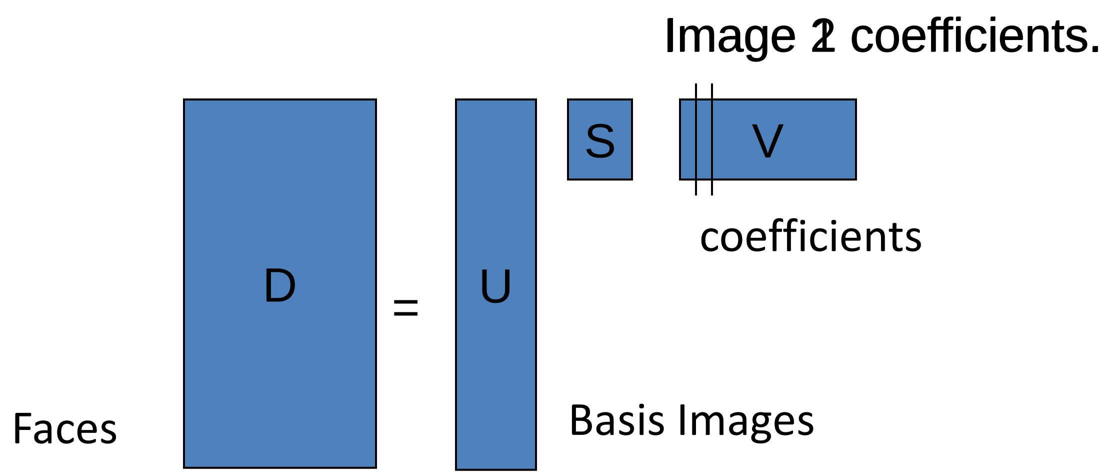
PCA for Faces

- Property of a set of images
  - Average image
  - The difference of each image from the average image
- Models the ways in which the images vary from each other

PCA Input = Original - Average

PCA Math

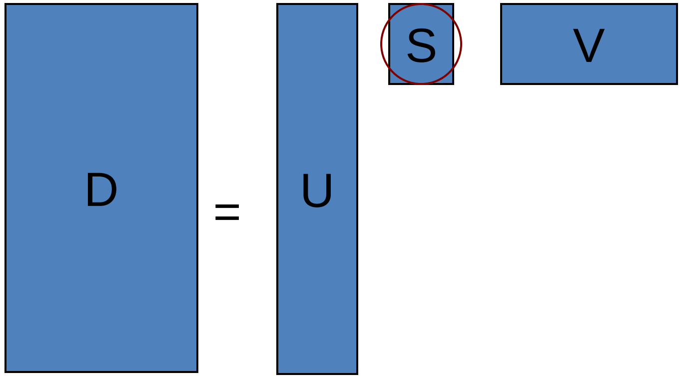
- Trying to represent each image as the average image and the weighted sum
- Our set of images is a 3D matrix I(x,y,n)
  - n is the number of people in the data set
  - Vectorize the matrix into a data matrix D(p,n)
  - Perform "Singular Value Decomposition" (SVD)
  - D = U S V



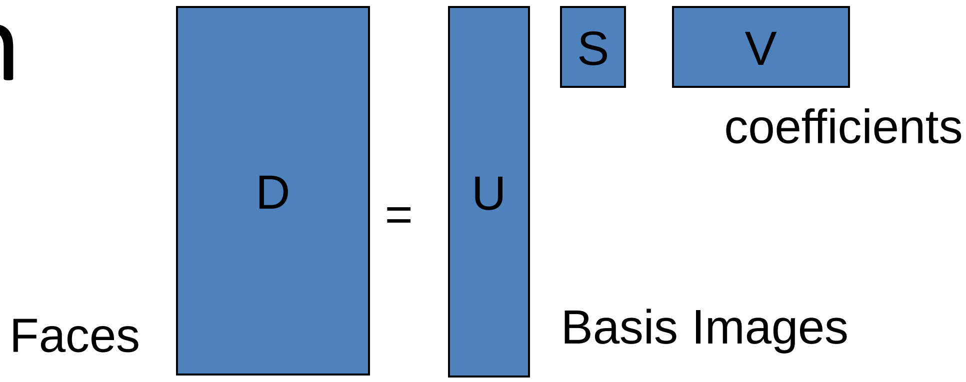
More PCA Math

- S is a diagonal matrix
  - Diagonal elements called singular values
  - These numbers are the **relative importance** of each of the principle components

- Two ways to look at components & coefficients
  - We can make the principal components be the columns of U * S, and have the columns of V be the coefficients.
  - Or, we can keep the columns of U, and make the coefficients be S * the columns of V. (more common)



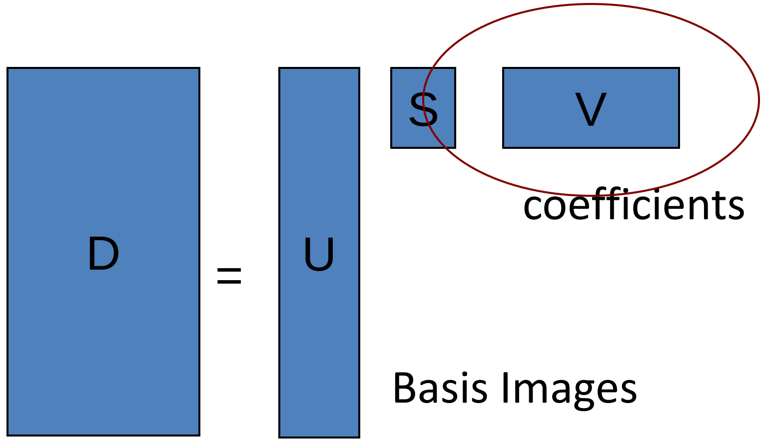
Projection



- You can *project* a new image W, to get its coefficients v_w simply as:
  $v_w = U^T W$

- Then U v_w approximately reconstructs W
  $U v_w = U (U^T W)$
  $= (U U^T) W$
  $= I W$
  $= W$

Backprojection

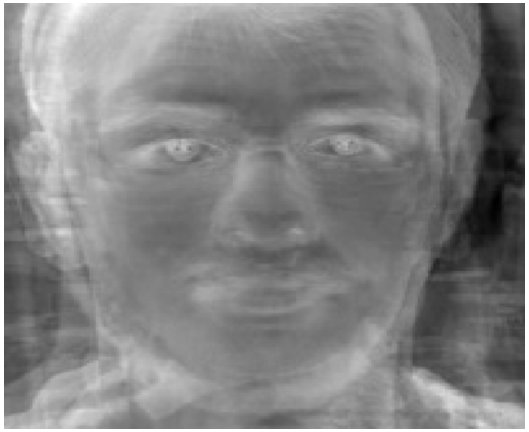


- Coefficients define the appearance of the image.
- The U matrix defines the space of possible images within this data set

- Given a new set of coefficients, we can make a new image
  - New image = U v
  - *This will give us a column vector of the pixel values, you have to rearrange it into the shape of the image*

Linear Combinations of Images

- For this data set, these are the (first three) principal vectors:



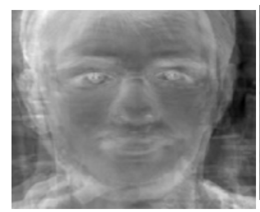
Linear Combinations of Images

- We can represent an image as the avg. image + “how much of each principal vector it uses”

=

-1.21 -2.88 .894

+



Avg Image

-.961 3.57 -3.60

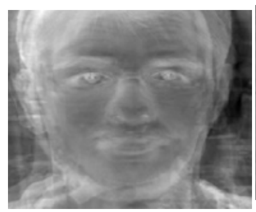
=

Linear Combinations of Images

What does this combination look like?

-1.21 -2.88 .894 =

+



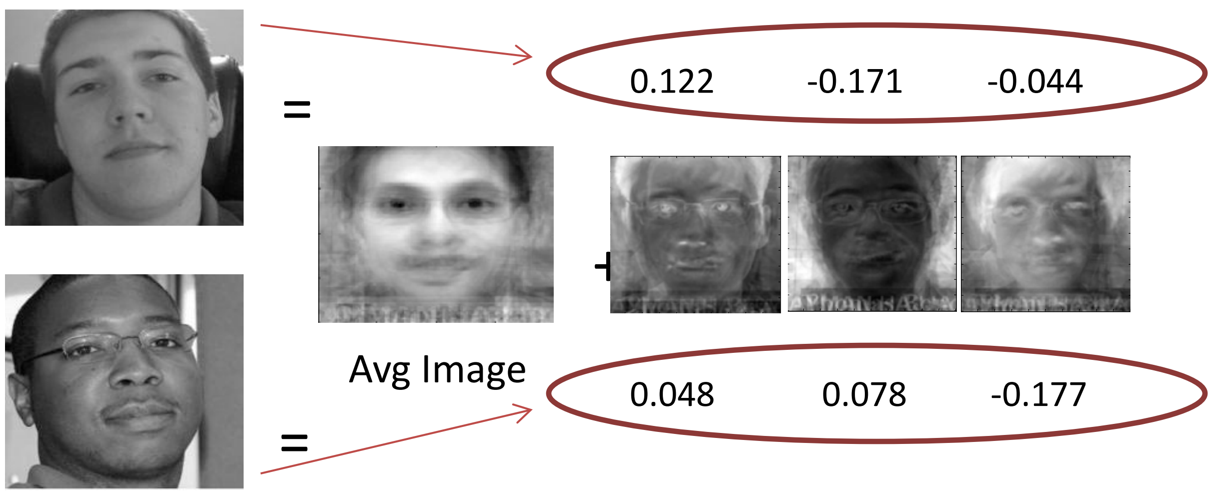
Avg Image

-.961 3.57 -3.60 =

Did it work?

PCA for Recognition

- PCA is a representational choice
  - Turns images into k-vectors
    - k is the number of basis images



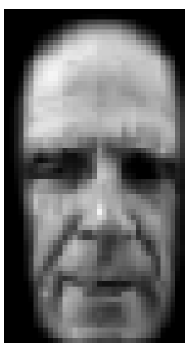
- How could we use this for recognition?

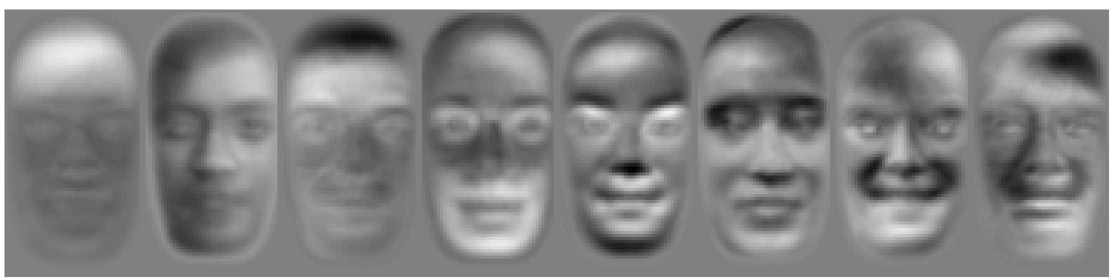
“Eigenfaces” Algorithm

“eigenfaces”

- Set up the image database as K-vectors of coefficients
- When a new image arrives…

1. subtract the median image (to center at the origin)
2. project onto the K eigenvectors to get the K most important coefficients





3. Use a classifier (neural net, nearest neighbors, SVM) to determine which face this K-vector is

match

Problems w/ Eigenfaces?

- Not capable of discovering nonlinear DoF
  - Real images tend not to be linear combinations of other images
- Registration and scaling issues
- Sensitive to changes in lighting conditions
- PCA projection is **optimal for reconstruction** from a low dimensional basis, but **may not be optimal for discrimination…**

Extensions to Eigenfaces

- Use the same basic idea
  - Change the low-dimensional image representation

Dimensionality Reduction Technique	Face Recognition Algorithm
Linear Discriminant Analysis (LDA) or Fisher's Linear Discriminant (FLD)	Fisherfaces
Independent Component Analysis (ICA)	ICA Faces
Tensor decomposition	Tensorfaces

How to test these algorithms?

Sample images in the training set.

(neutral expression, anger, and right-light-on from first session; smile and left-light-on from second session)

Sample images in the test set.

(smile and left-light-on from first session; neutral expression, anger, and right-light-on from second session)

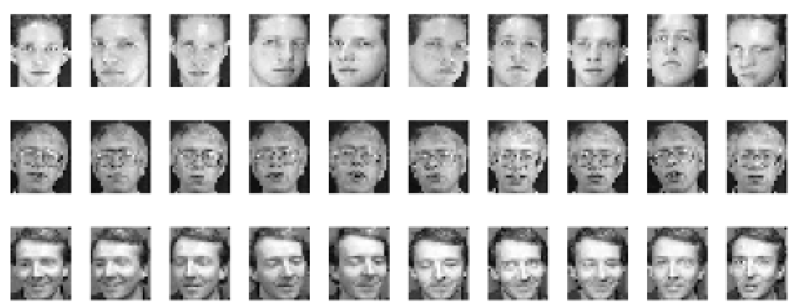
AT&T Database

- Pose variation
- 40 classes, 10 images/class, 28 by 23

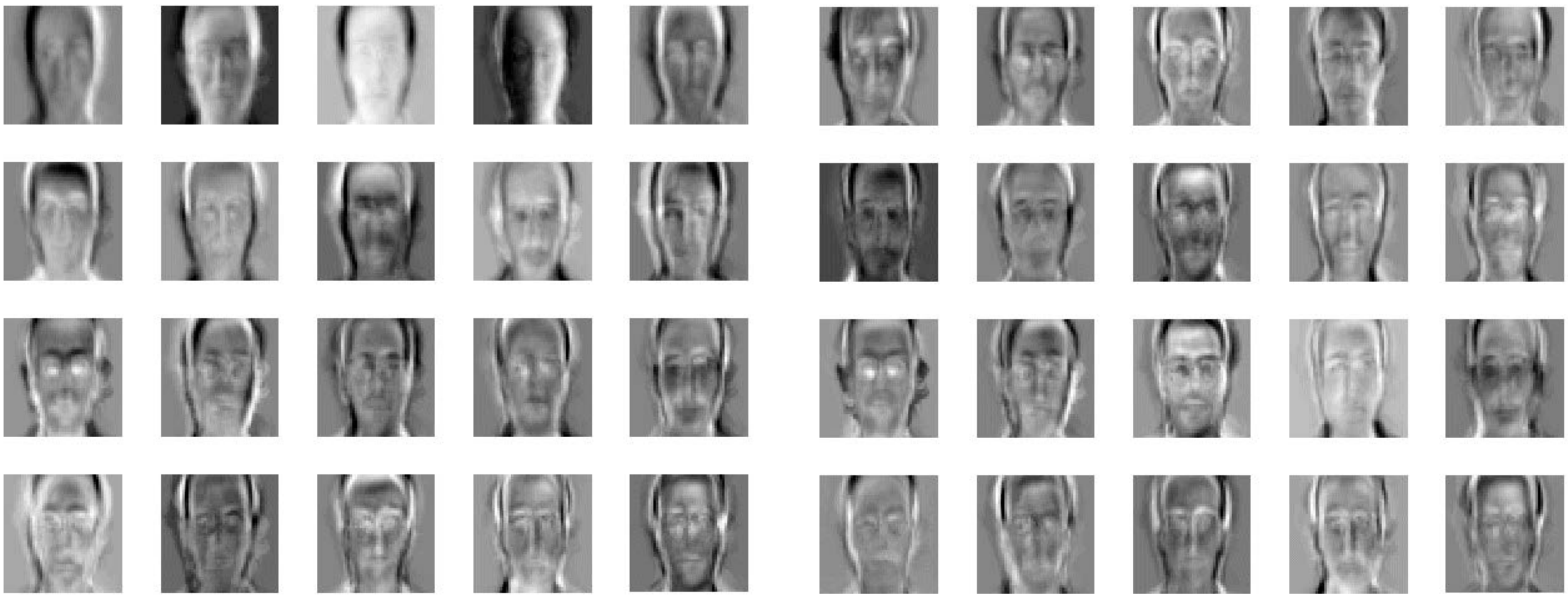
Set1

Set2

(Mirror of Set1)



Eigenfaces vs ICA Faces



(a) (b)

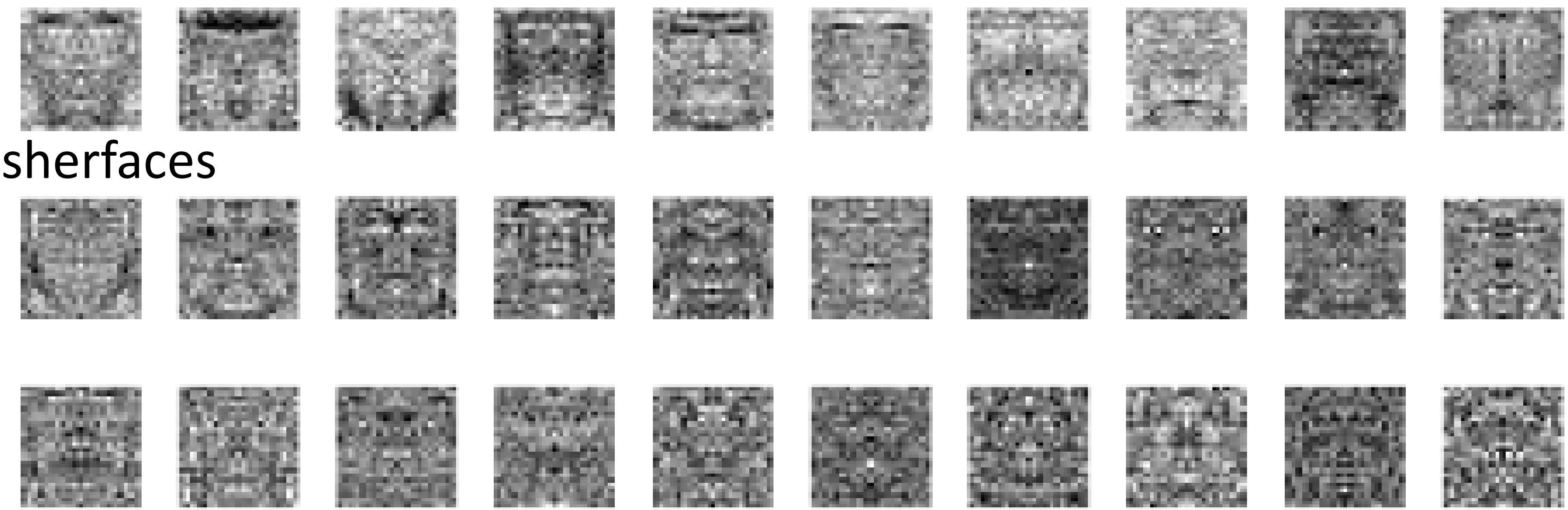
First 20 basis images: (a) in eigenface method; (b) factorial code. They are ordered by column, then, by row.

FERET Database

- Facial expression and illumination variation
- 200 classes, 3 images/class, 24 by 21

Eigenfaces

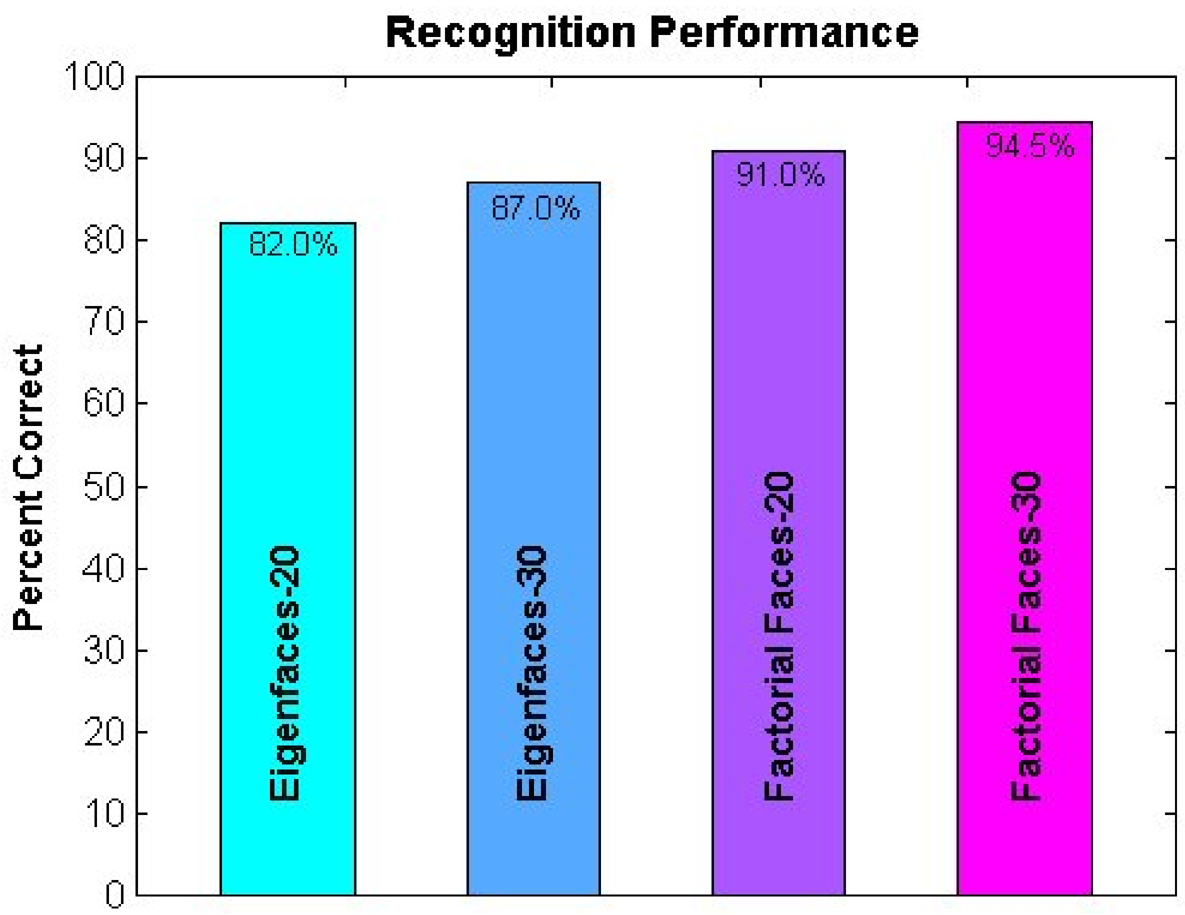
Fisherfaces



Eigenfaces

ICA faces

Experimental Results: NN



Summary

- Basic idea for image-based biometrics
  - Derive features from images
  - Use machine learning techniques in feature space for identification
- Eigenfaces uses PCA coefficients of images as representation
- Variants of Eigenfaces are used for face recognition