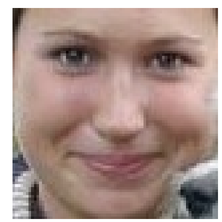
FACE ANALYSIS AND RECOGNITION

Class 9: Artificial Vision

The “Margaret Thatcher Illusion”, by Peter Thompson

Face detection and recognition

Detection

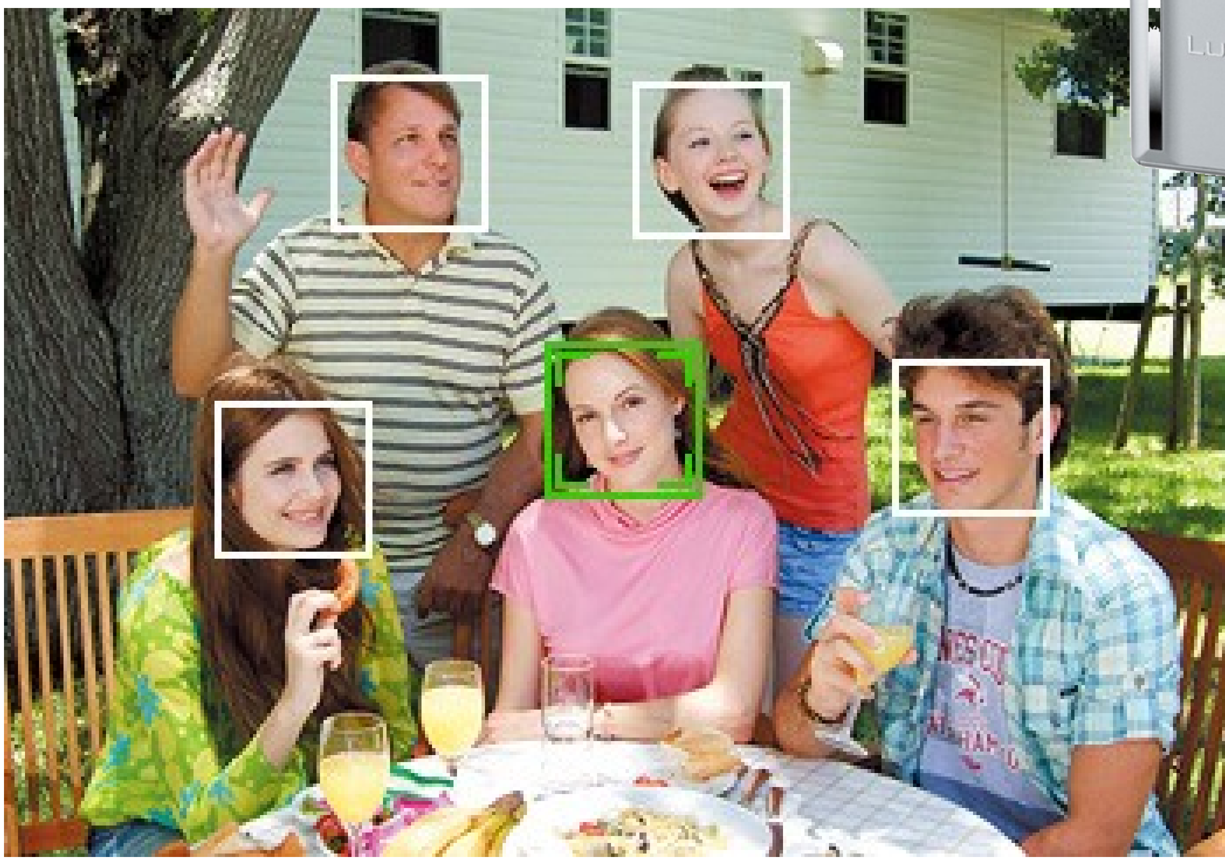


Recognition

“Sal
ly”

Applications of Face Recognition

- Digital photography

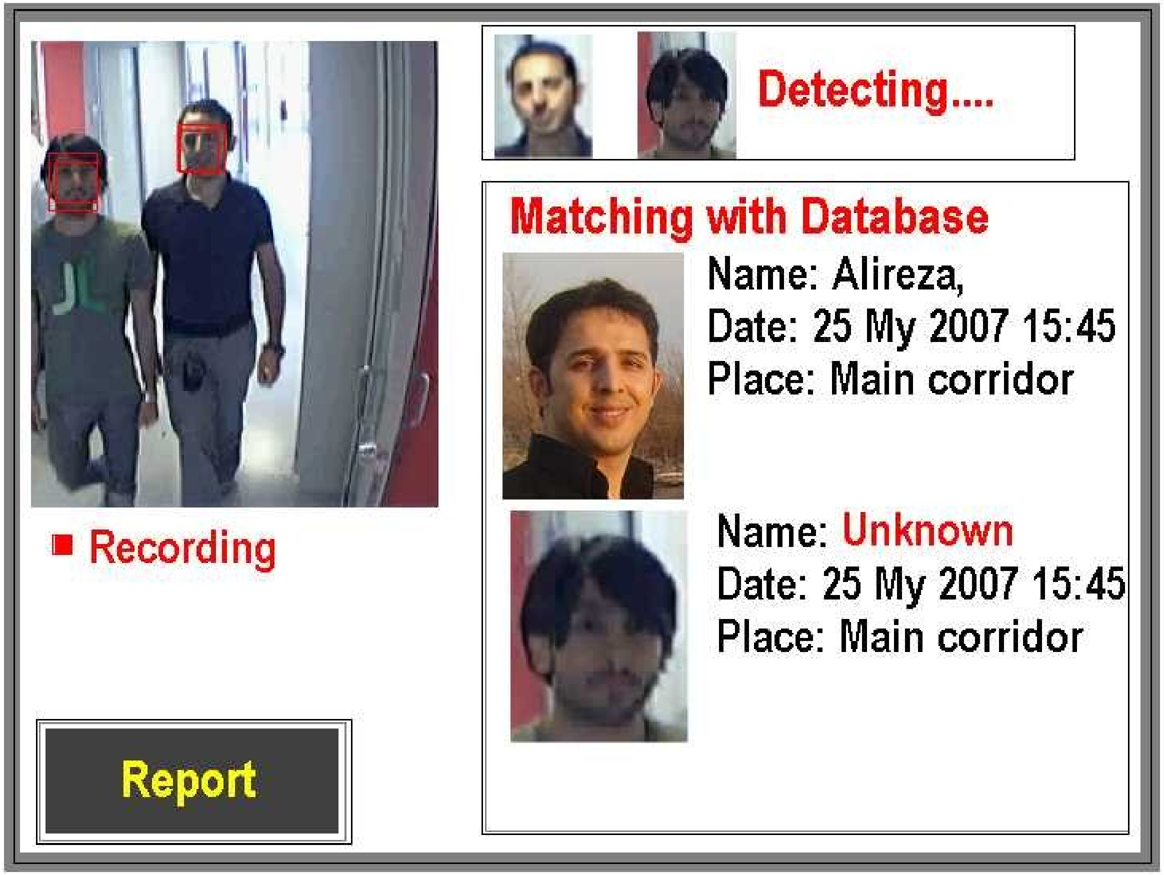


Applications of Face Recognition

- Album organization

Applications of Face Recognition

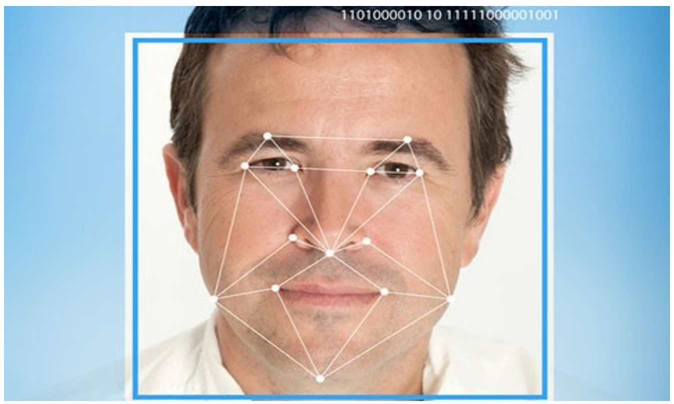
- Surveillance

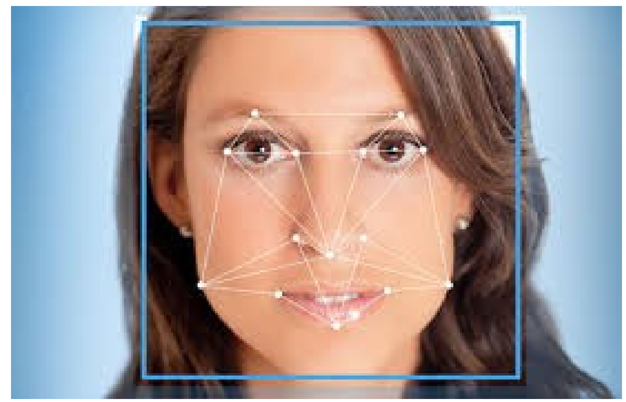


Applications of Face Recognition

Biometrics systems

Biometric (in greek bios= life and metron = measurement) is a science that, through automated methods, is able to verify or recognize the identity of a person by measuring and identifying personal features (morphological, physiological, behavioural)





Why it works? Each person has morphological features that are unique and therefore, different from everyone else.
Biometric systems focus on something you are not something you have
(as a password)

Applications of Face Recognition

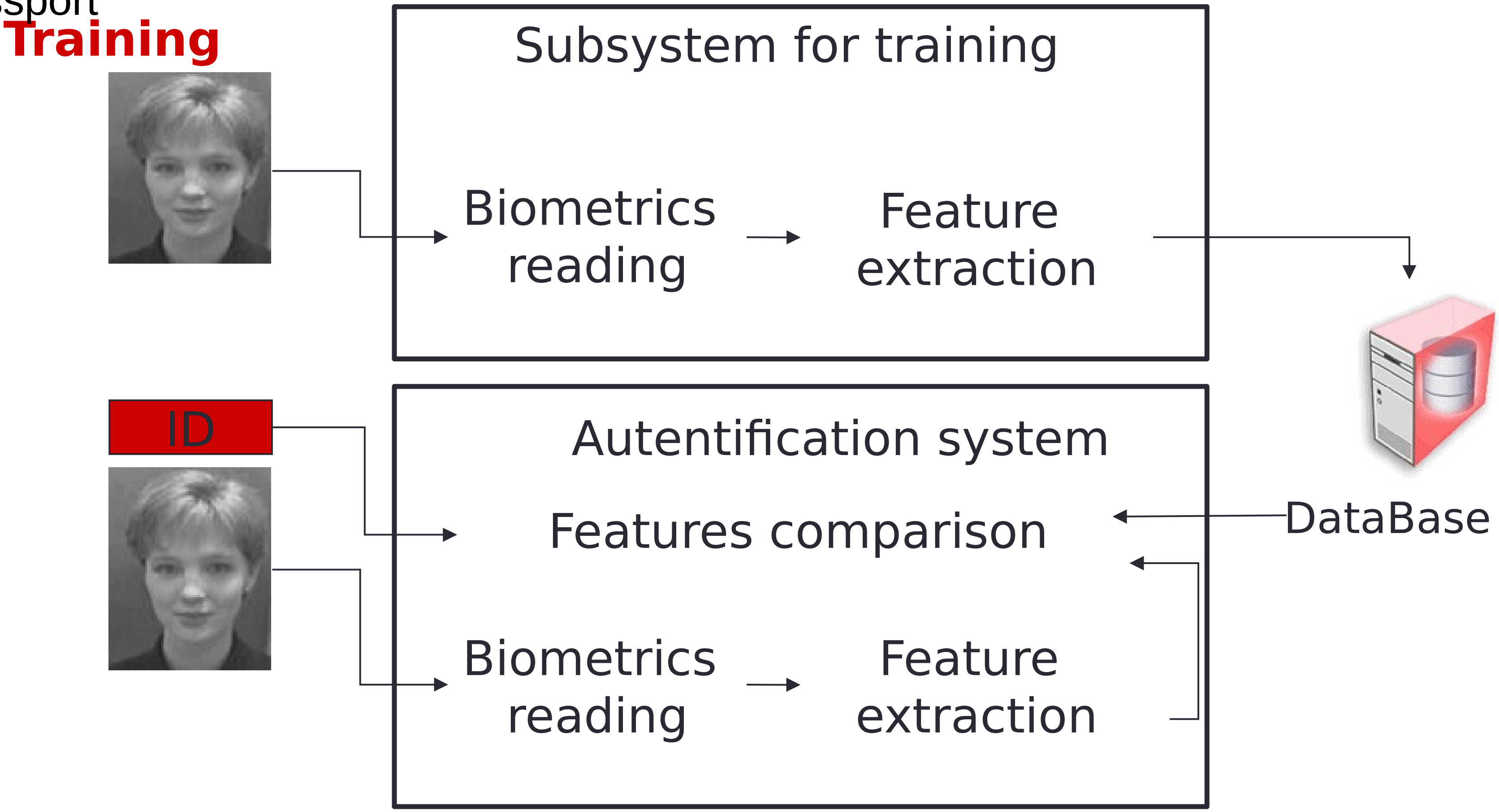
Biometrics systems

1. Authentification

Answer to the question: **Are you are the same person as before?**

Comparing your today's face with the picture on your passport

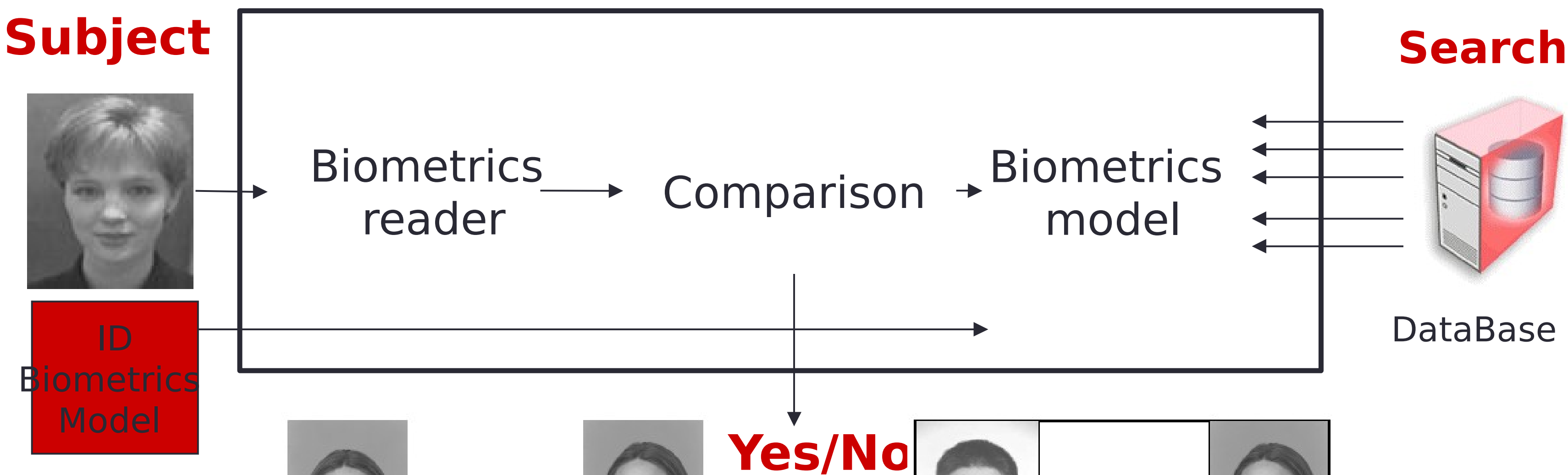
Extract biometric measures from your face and compare with the biometric in your passport



Biometrics systems

2. Verification

Answer to the question: **Is this person who they say they are?**
The person claim to be Sally, the system compare with
the biometric model of Sally in the database and says yes or not.
Compared with only one identity. 1-to-n matching system.



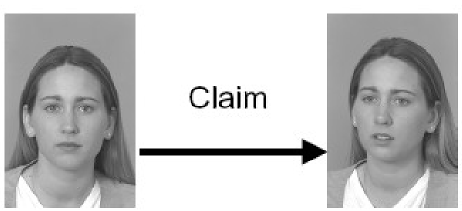


Figure 3[6] – Correct Verification Claim

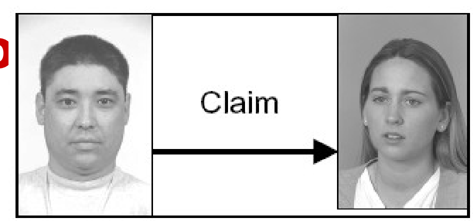


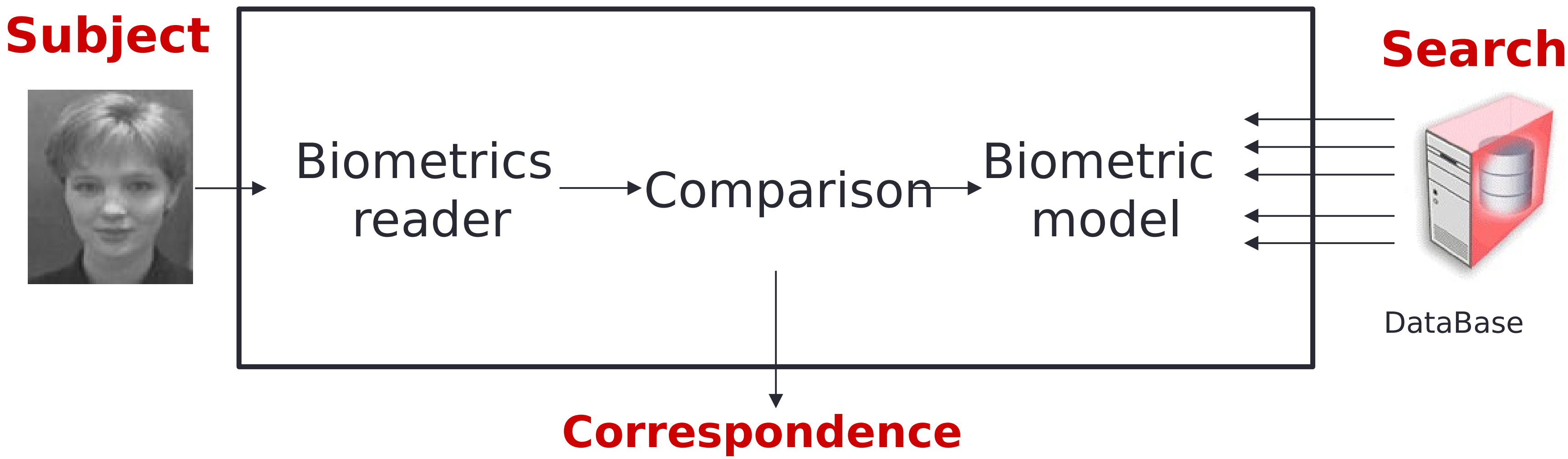
Figure 4 - False Verification Claim

Biometrics system

3. Identification (recognition)

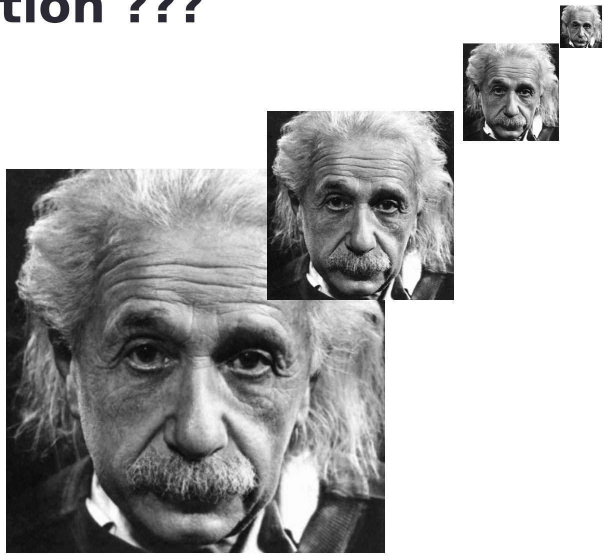
Answer to the question: **Who generated this biometric?**
The biometric of the subject is extracted and compared with all the biometrics in the database.

1-to-n matching system, where n is the total number of biometrics in the database



Problems and difficulties of facial analysis

How complex is the task of automatic recognition ???



40 years ago , researchers started to ask:

- How much the recognition degrades as a function of the **spatial resolution** (number of pixels)?!
- What kind of **degradation of the image** affect the capacity to perceive images of faces?!
- How **facial expressions, pose, illumination** affect facial recognition?!
- What is the minimum size to detect a face in an image?!

Asperger

Problems and difficulties of facial analysis

Blurred images

36x36 24x24 16x16

Low spatial resolution

Goal of the automatic analysis

The problem of face recognition: recognize the identity of a person (from a given set of face images).

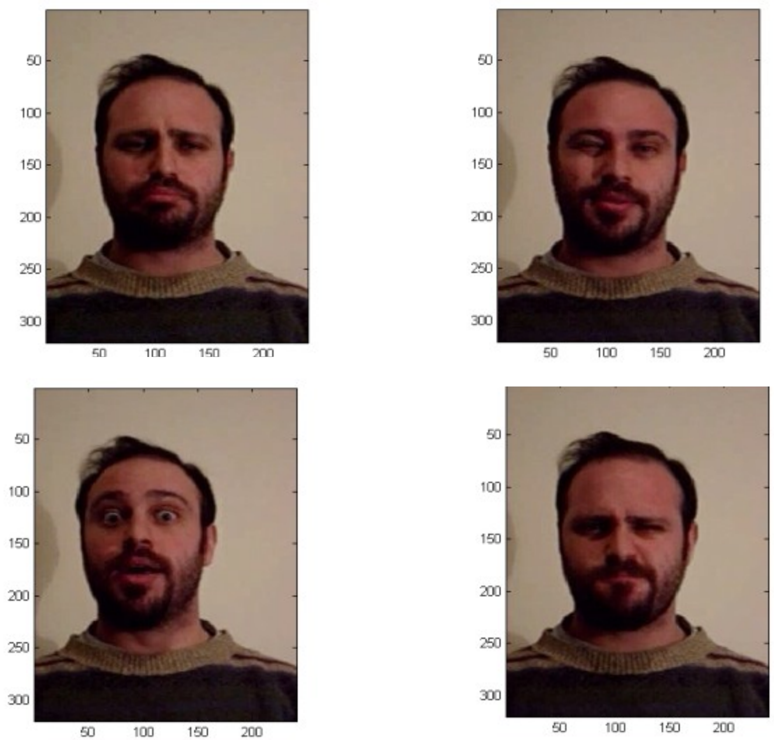
Goal: *To define a space of image features that allow to represent faces based on their appearance (or a set of local features) in the image.*

There are several parts to consider:

- **Define** an appropriate **representation** (descriptors of faces)
  - Normally, **reduce** the size of the data preserving the invariance and removing redundant dimensions.
- **Train a classifier** from a set of examples with their descriptors.
- **Recognize** a new face example using the learned model.

Problems and difficulties of facial analysis

Problem: Equivalence between stimuli
Same person, but very different expression: how to capture the discriminative features of his face?



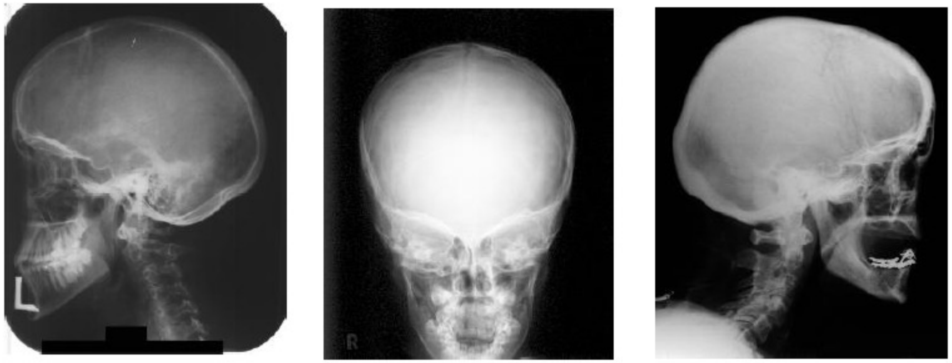
Dimensionality and redundancy

The appearance ...

Can we directly store a collection of many views of the object?

How can we store this information in a compact and efficient way?

How do we encode this information as interesting visual representation?



Starting idea of "eigenfaces"

1. Treat pixels as a vector

$\mathbf{x}$

Consider as feature vector of a image face, the same image reshaped to a vector

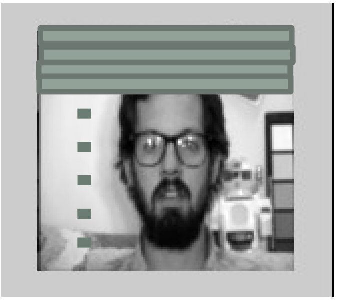
2. Recognize face by nearest neighbor

$\mathbf{y}_1 \ldots \mathbf{y}_n$

$$k = \underset{k}{\operatorname{argmin}} \left\| \mathbf{y}_k{}^T - \mathbf{x} \right\|$$

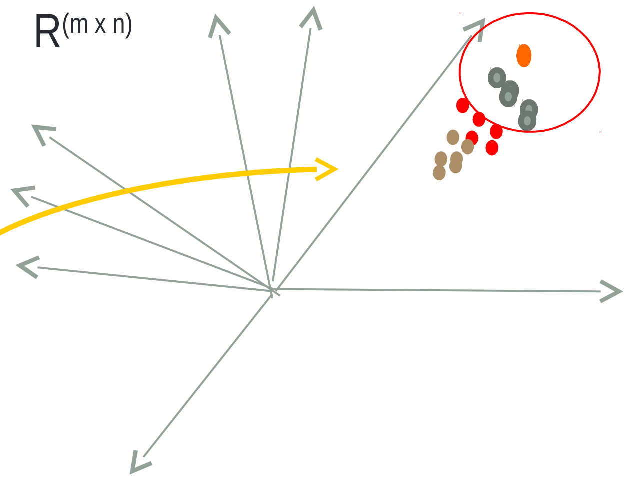
The feature space of faces and eigenfaces recognition

1. Features

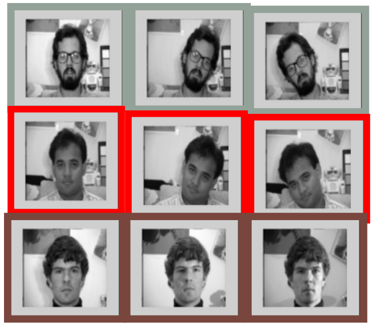


$(x_1, x_2, \ldots\ldots\ldots\ldots x_{(m \times n)})$

2. Classifier - knn

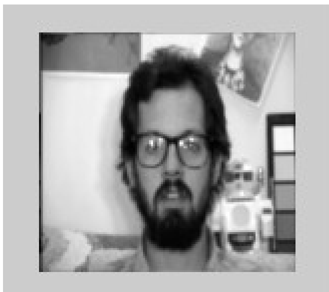


Training



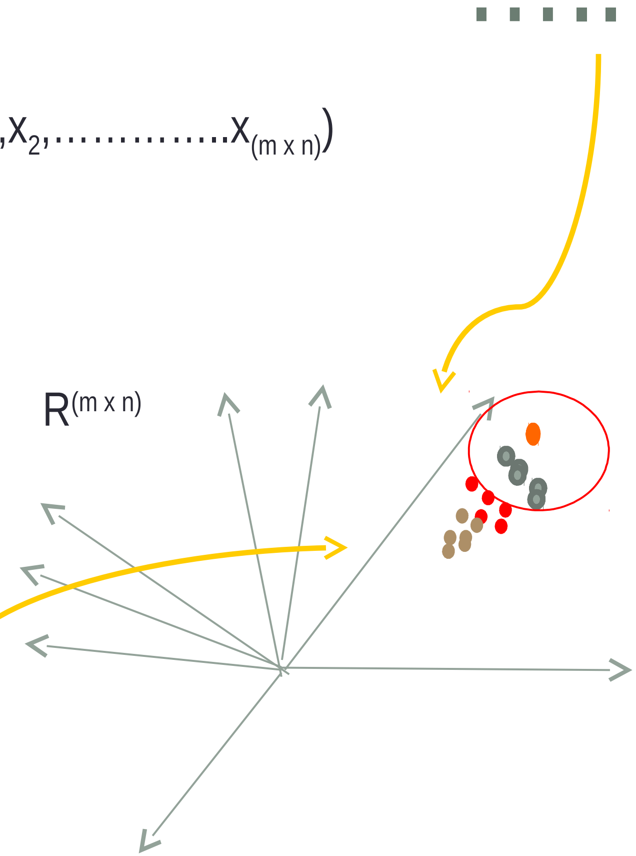
The feature space of faces and eigenfaces recognition

Representation of faces as points in high dimensional space



$(x_1, x_2, \ldots\ldots\ldots\ldots x_{(m \times n)})$

Each image has m rows and n columns -> defines a vector of (mxn) elements.



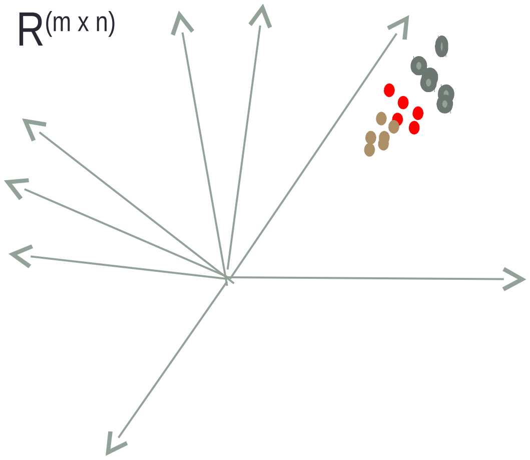
Training

Recognition:

Classifier KNN – assigns the majority label of the k closest neighbors of the training set.

The space of all face images

- **Problem:**
- When viewed as vectors of pixel values, face images are extremely high-dimensional
  - 100x100 image = 10,000 dimensions
- We use KNN in the R^{mxn} space -> costly and slow (m x n = 256 * 256 = 65536).
- However, relatively few 10,000-dimensional vectors correspond to valid face images
- We want to effectively model the subspace of face images

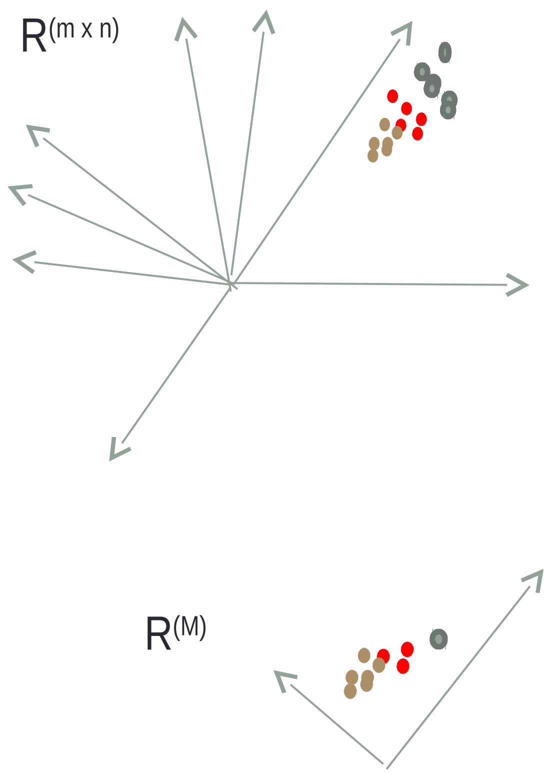


Eigenfaces

- **Can we find a more compact representation of images where each face is represented via a small set of parameters**?

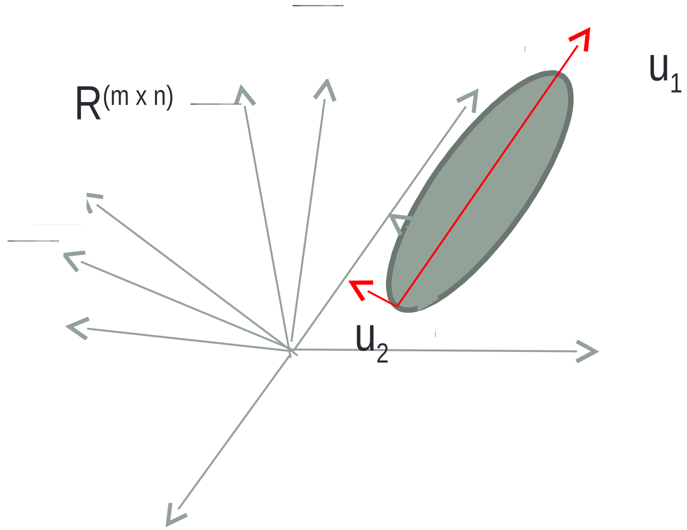
- We look for a transformation of the original space to a smaller space (M << (m x n)), where faces are represented with their coordinates in this new space R^M? M is the number of images !

- Retaining the information necessary to classify, recognize, etc!
- And removing - or minimizing - information that is not relevant (lighting, small variations ..).



Eigenfaces

How to classify the faces?



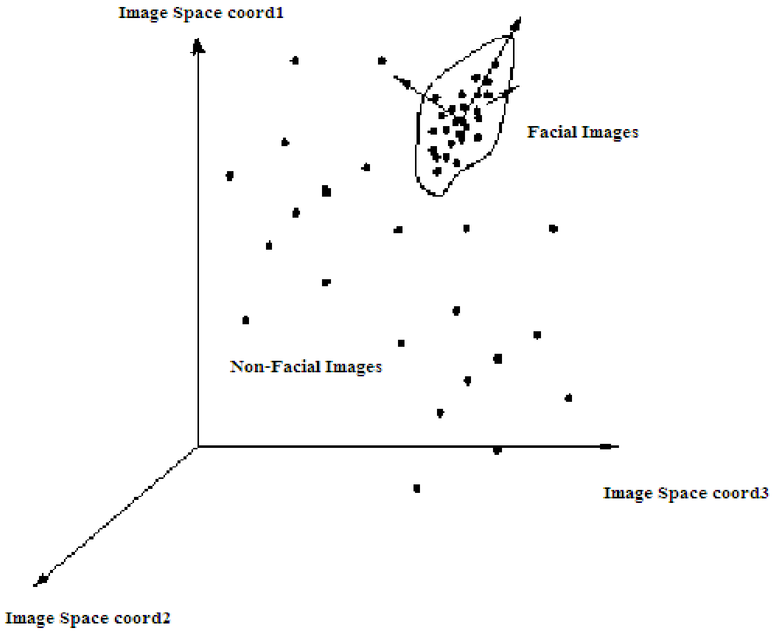
If you have the reduced space and want to classify a face X_i:

1. Project it into the new (reduced) space:
2. apply the KNN (considering the k projected neighbor faces of the training set).

How to build a reduced space?

Principal Component Analysis (PCA)

- A m *x n* pixel image of a face, represented as a vector occupies a single point in (mxn)-dimensional image space.
- Images of faces being similar in overall configuration, will not be randomly distributed in this huge image space.
- Therefore, they can be described by a low dimensional subspace.
- Main idea of PCA for faces:
  - To find vectors that best account for variation of face images in entire image space.
  - These vectors are called eigen vectors.
  - Construct a face space and project the images into this face space (eigenfaces).



Eigenfaces

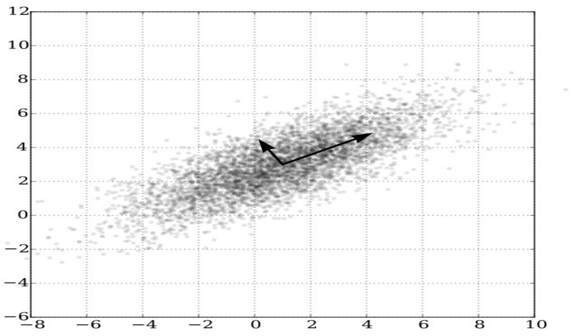
- **Principal Component Analysis (PCA)**

- It is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible) and the all others are orthogonal each other and with decreasing variance.

- Given: M data points $\mathbf{x_1, \ldots, x_M}$ in $R^{(m \times n)}$

If $\mathbf{u_k}$ in $R^{(m \times n)}$ are the principal components that best captures the most data variance, we can express each image as linear combinations of these vectors

-



Eigenvalues and Eigenvectors - Definition

- If v is a nonzero vector and λ is a number such that

  Av = λv, then

  v is said to be an *eigenvector* of A with *eigenvalue* λ.

<u>Example</u>

λ (eigenvalues)

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 3 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

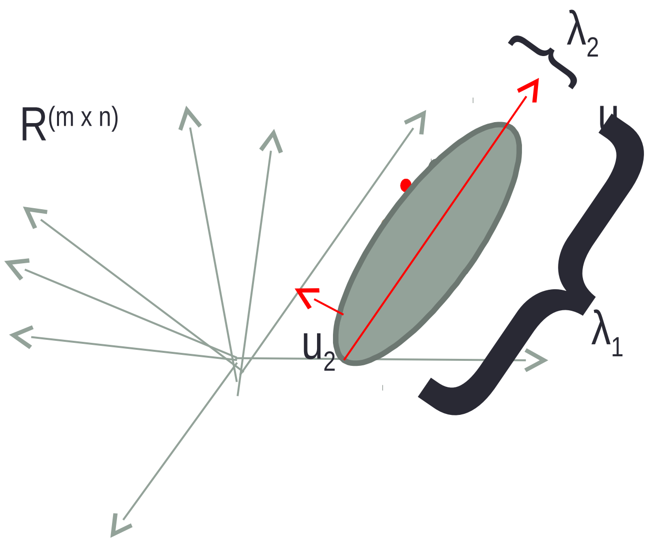
A

V (eigenvectors)

Principal Component Analysis: the algorithm

Remember:

- The covariance between two variable is a measure of how much the two variables change together
- The covariance matrix of the data generalizes the notion of covariance to multiple dimensions



According to the PCA, if we compute:

- The covariance matrix of the data
- the eigenvectors (e1, e2, ...) of the covariance matrix define the axis of maximum variance, they are orthogonal and form a basis.
- and the eigenvalues give a measure of the variance of the data in the direction of the corresponding eigenvalues

Eigenfaces

How to find the best sub-space to represent the face family?

- Given M images of size (mxn) -> let's construct vector X_i, i = 1 ... M in the space $R^{m \times n}$.
- Given the images of training: X_i, i = 1 ... M, let's compute the mean image: $\overline{X} = \frac{1}{M}\sum_{i=1}^{M} X_i$
- Construct the covariance matrix Σ:

$$\Sigma = \frac{1}{M}\sum_{i=1}^{M}(X_i - \overline{X})(X_i - \overline{X})^T = AA^T; \qquad A = [X_1 - \overline{X}, X_2 - \overline{X}, \ldots X_M - \overline{X}]$$

- The matrix columns are called eigenfaces.
- => A is of size (mxn) x M, $A*A^T$ is of size (mxn) x (mxn)!

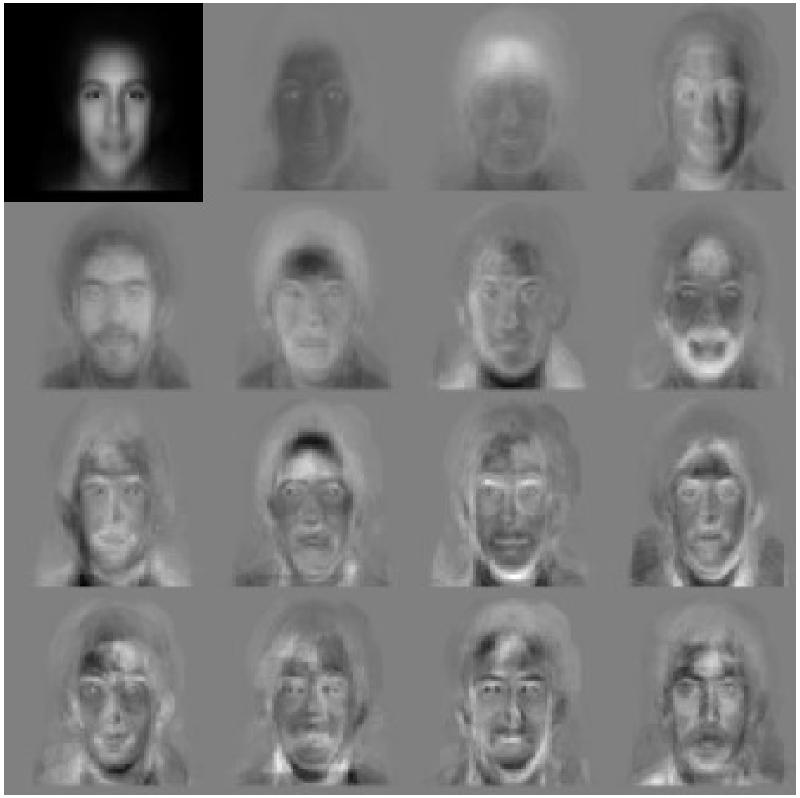
M. Turk and A. Pentland, Face Recognition using Eigenfaces, CVPR 1991

Eigenfaces illustration

Example of eigenvectors faces.

The mean face and the eigenfaces of the covariance matrix of the faces (shown as pictures):

Mean face $\overline{X}$

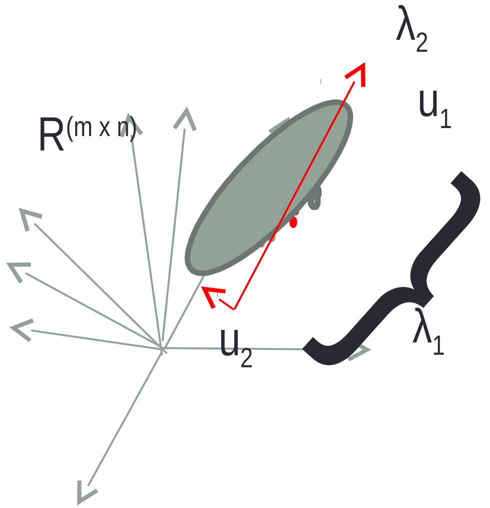


Eigenfaces (U)

Eigenfaces

How to find the best sub-space to represent the face family?

- Eigenvectors (u_1, u_2, ...) of the covariance matrix Σ define the subspace that represents the data (faces) distribution.



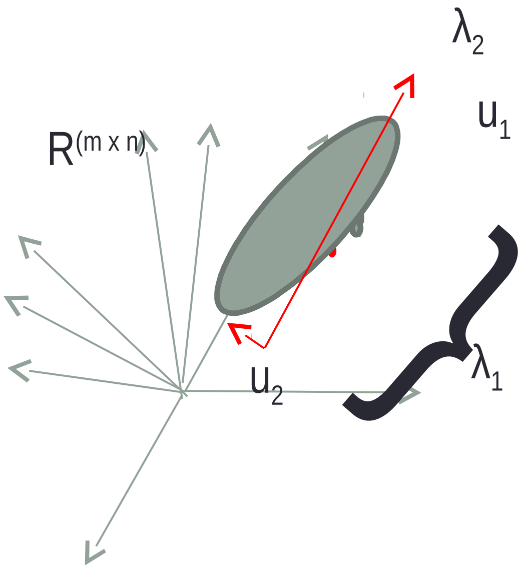
- The original image once centered is projected into the eigenspace: X-> Y:

$$Y_i = (X_i - \bar{X})^T * (u_1, u_2, \ldots u_M)$$

Eigenfaces

What is the eigenvalue representing?

- The eigenvalues λ (eigenvalues) measure the variance of the data in the direction of the eigenvector =>



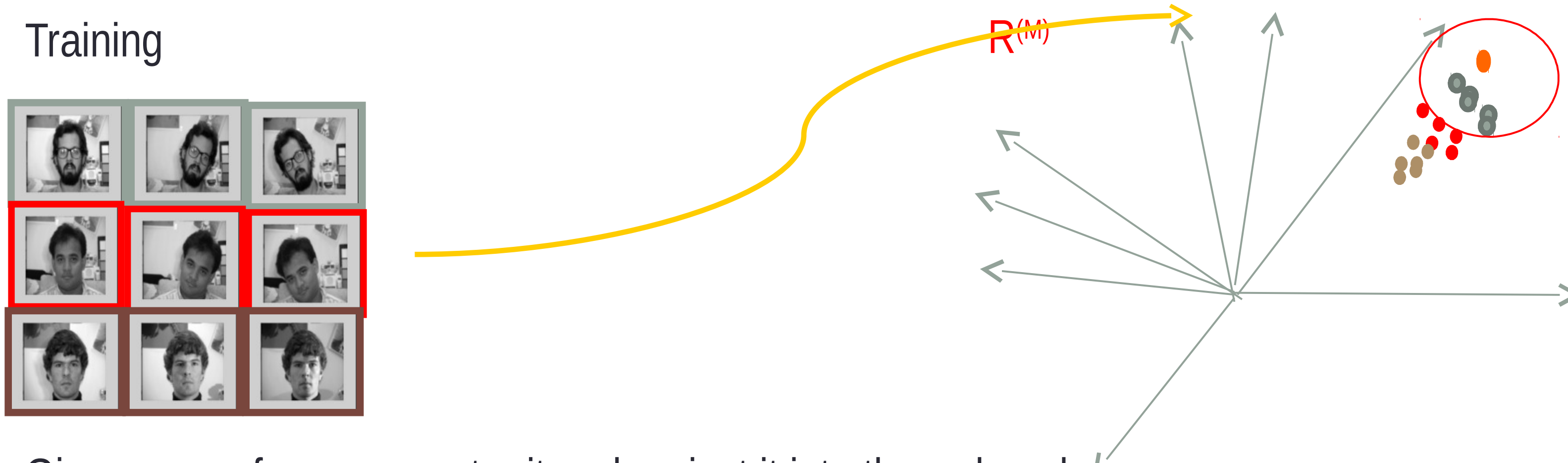
- The larger the λ_i, there is more variance in the data vector in the direction of the eigenvector u_i.

- if $\lambda_i = 0$ => we can avoid eigenvector u_i! Why?

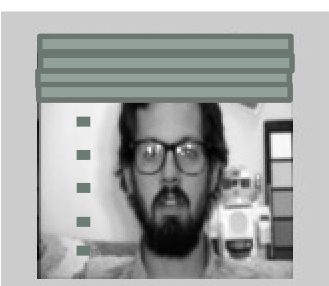
- => We only are interested in the first k eigenvectors of the largest eigenvalues (k = 1,2, ... M-1).

The procedure for recognition with Eigenfaces

1. Training: Given the training set, we compute the eigenfaces $U = (u_1, u_2, \ldots u_M)$, U - eigenvectors of $A*A^T$



2. Recognition: Given a new face, we center it and project it into the reduced space:



$(x_1, x_2, \ldots\ldots\ldots\ldots x_{(m \times n)}) \rightarrow Y = (\omega_1, \omega_2, \omega_3 \ldots \omega_M)$

Project X:

$$Y = (X - \overline{X})^T * U$$

3. Classify – using knn in R^M

Eigenfaces

How to obtain the eigenface sub-space?

- Objective: To find the eigenvectors and eigenvalues of $A*A^T$.

- M is the number of images in the dataset, (mxn) the number of pixels of each image

- **Problem**: The matrix A has size (mxn)x M and the covariance matrix $A*A^T$ has size (mxn) x (mxn) -> find the eigenvalues is untreatable!

Tip: Instead of $A*A^T V= \Lambda V$ consider $(A^T*A)V= \Lambda V$

- For the definition of eigenvalue and eigenvectors:
  - V is a matrix with columns - the eigenvectors of A^T*A,
  - Λ is a diagonal matrix with elements - the eigenvalues,
  - A^T*A is of size MxM (M << MXN)!

- But how to get the eigenvalues and eigenvectors of $A*A^T$?

A^T*A and $A*A^T$ have the same eigenvalues !

- **Trick**: Multiplying both sides with A:
  - $A*A^T AV=A\Lambda V \Rightarrow A*A^T \ AV= \Lambda \ AV \Rightarrow A*A^T U= \Lambda U$ where U=AV

- => U is the matrix with columns the eigenvectors of the matrix $A*A^T$, and U = AV.

Eigenfaces

How to obtain the eigenface sub-space?

- We can compute easily the eigenvector V of A^T*A and from them the eigenvectors U of $A*A^T$ as U = AV, which correspond to the eigenfaces !

The matrix A^T*A is of size M x M, where M is the number of images for training!

Note: If M << (m x n), it can be shown that there will be only M-1 values different from 0!

If the images are of size 256 x 256 = 65536 -> A: 65536xM, $A*A^T$: 65536x65536, A^T*A is of MxM.

!

Eigenfaces

Algorithm to obtain the eigenfaces

1. Construct the matrix A^T*A of size M x M where the columns are centered faces:

$$A = [X_1 - \bar{X}, X_2 - \bar{X}, \ldots X_M - \bar{X}]$$

2. Compute the eigenvectors $V = [v_1, v_2, \ldots v_m]$ matrix of A^T*A.

3. Sort by magnitude of their corresponding eigenvalues and keep the most important eigenvectors (with higher eigenvalues).

4. The M eigenfaces are obtained by multiplying the matrix A with v_i:

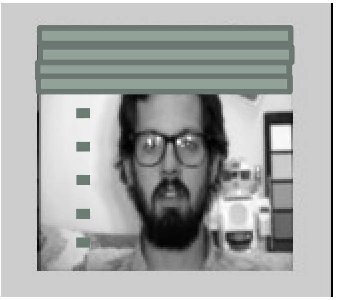
$$u_l = \sum_{k=1}^{M} v_{lk} A_k, \; l = 1, \ldots M$$

where A_k are the columns of the matrix A.

Eigenfaces: the Algorithm for Face Recognition

1. Given the training set, we compute the eigenfaces $U = (u_1, u_2, \dots u_m)$, $U = AV$, V - eigenvectors of $A^T * A$

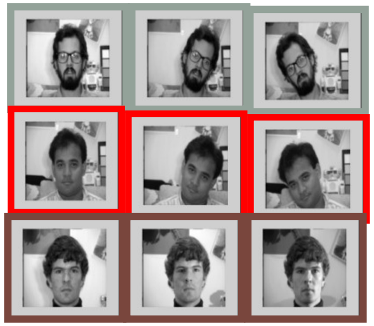
2. Center and project the new face to recognize it:



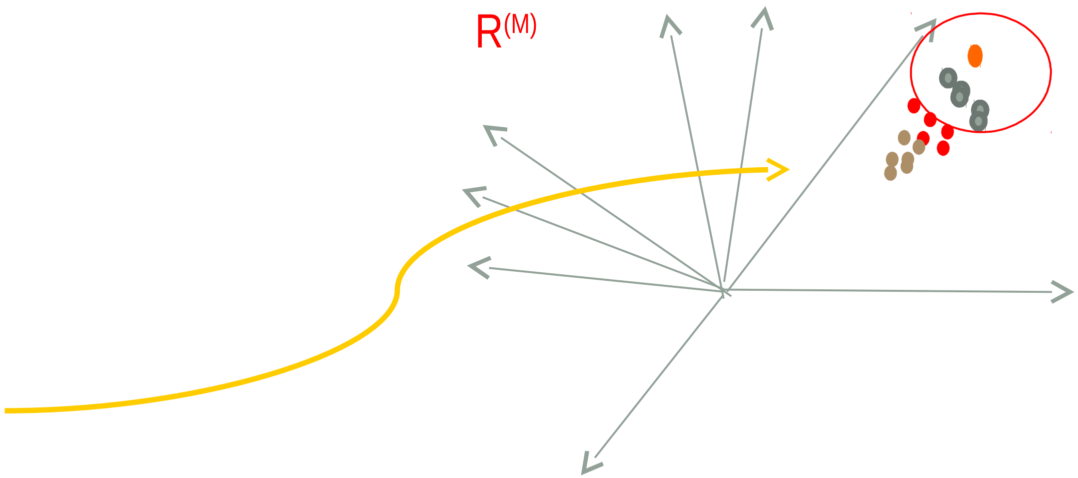
$(x_1, x_2, \dots\dots\dots\dots x_{(m \times n)}) \to Y = (\omega_1, \omega_2, \omega_3 \dots \omega_M)$

Project X: $$Y = (X - \bar{X})^T * U$$

3. Classify– by knn in R^M

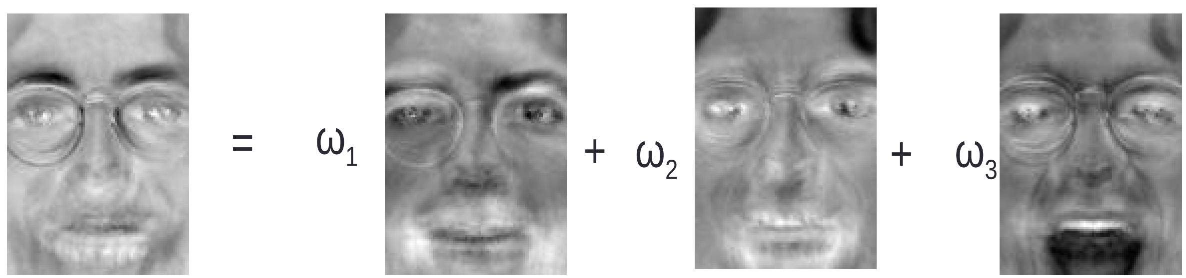


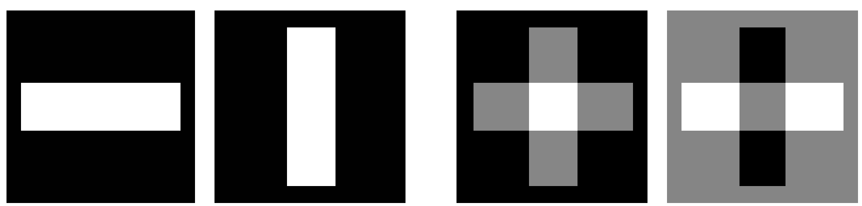
Training set



Eigenfaces

Interpretation: Since eigenfaces are forming a base, we can express a face as a linear combination of eigenfaces.





Exercise: Given 2 images and 2 eigenfaces, express how each of the first two images is represented by the eigenfaces as base (what would be the weights)?

Note: The method recommends that the faces are aligned (usually, eyes-centered)!

Eigenfaces

What would be the representation of the original faces in the reduced space?

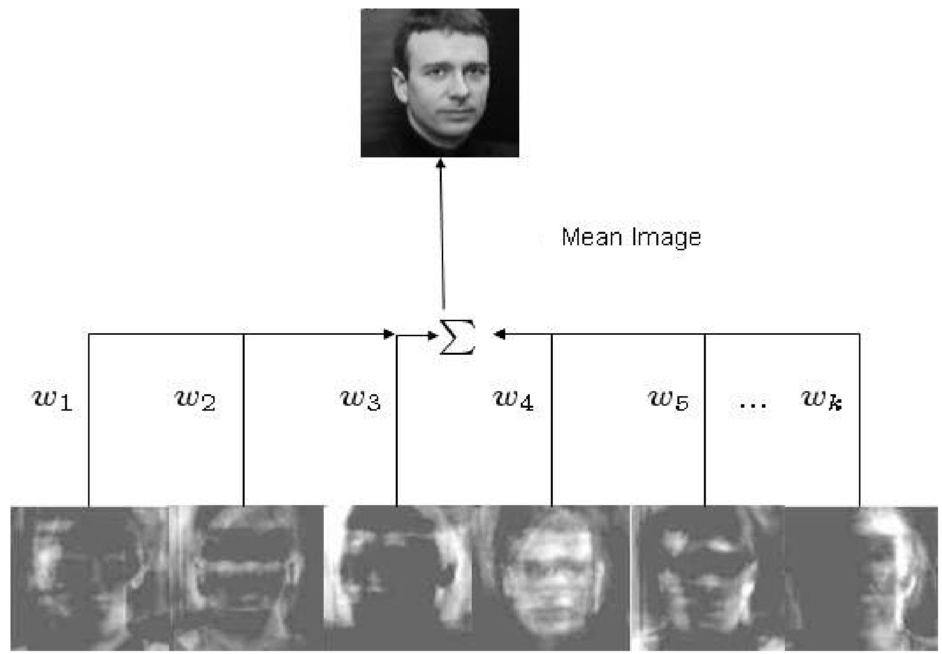
- The original image X is projected in the reduced space : X->Y:

$$Y_i = (X_i - \overline{X})^T * (u_1, u_2, \ldots u_k)$$

where $Y_i = (\omega_1, \omega_2, \ldots \omega_k)$ is the vector representation in the reduced space (k = 1,2, ... M).

Hence:

$$X_i = \overline{X} + \sum_{i=1}^{k} \omega_i u_i$$



u_1 u_2 u_3 u_4 u_5 u_6

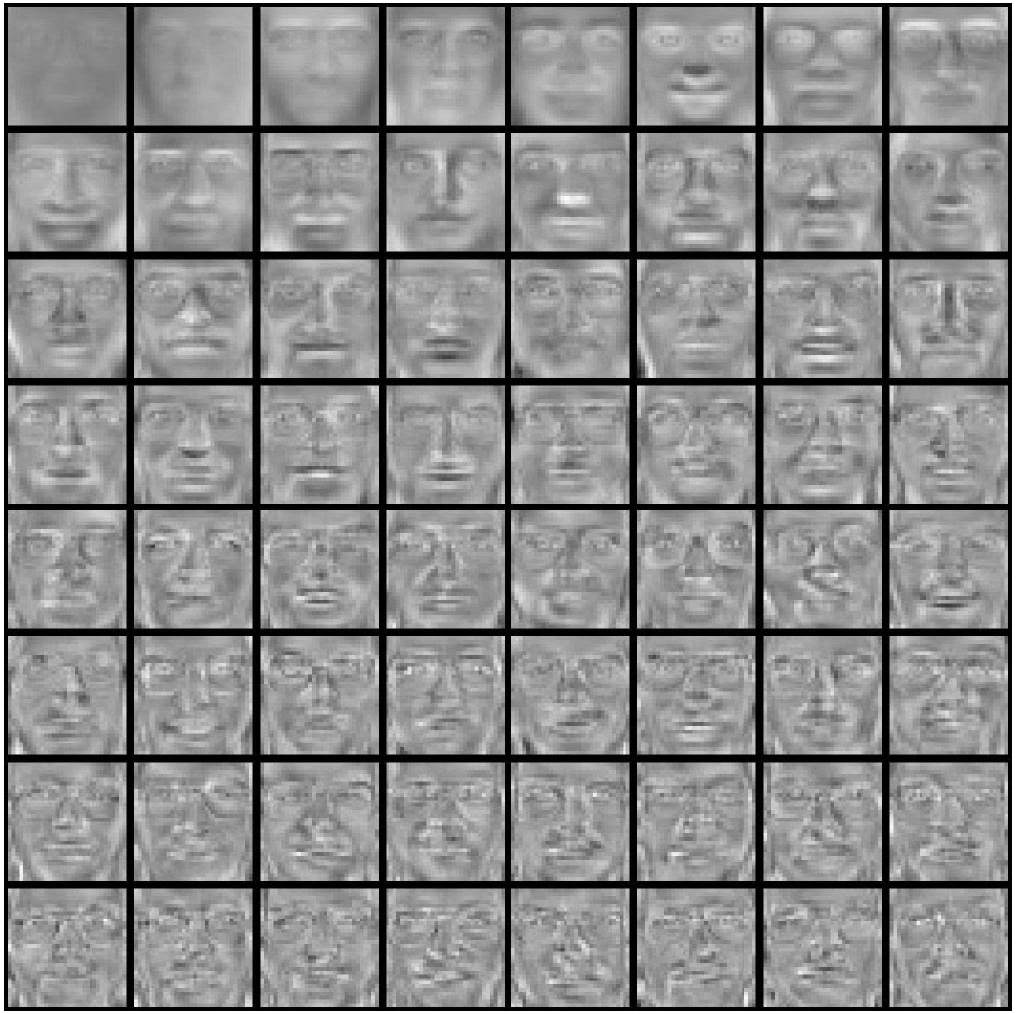
Eigenfaces example

- Training images
  - $\mathbf{x}_1,\ldots,\mathbf{x}_N$

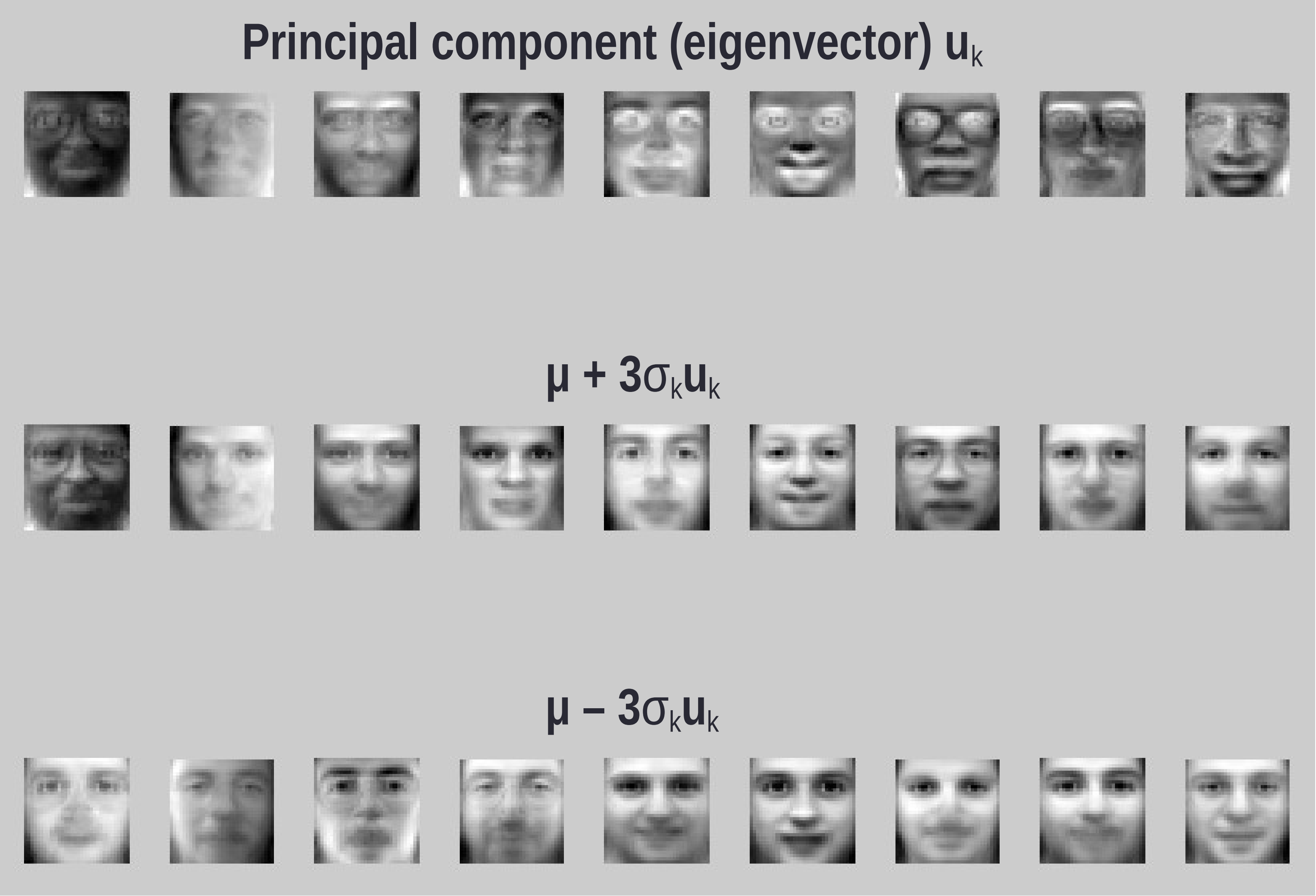
Eigenfaces example

Mean: $\boldsymbol{\mu}$

Top eigenvectors: $\mathbf{u}_1,\ldots\mathbf{u}_k$



Eigenfaces example



Eigenfaces example

Face **x** in “face space” coordinates:

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x} - \mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x} - \mu)]$$

$$= \quad w_1, \ldots, w_k$$

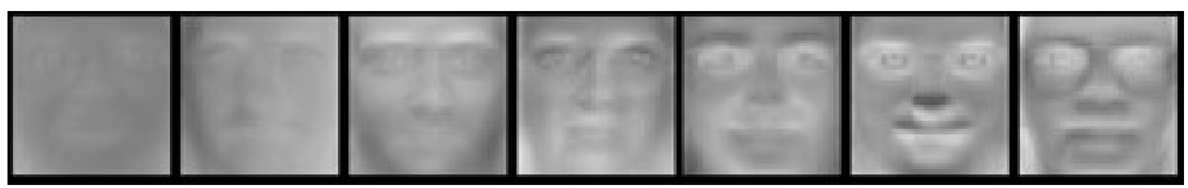
Eigenfaces example

Face **x** in "face space" coordinates:

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x} - \mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x} - \mu)]$$

$$= \quad w_1, \ldots, w_k$$

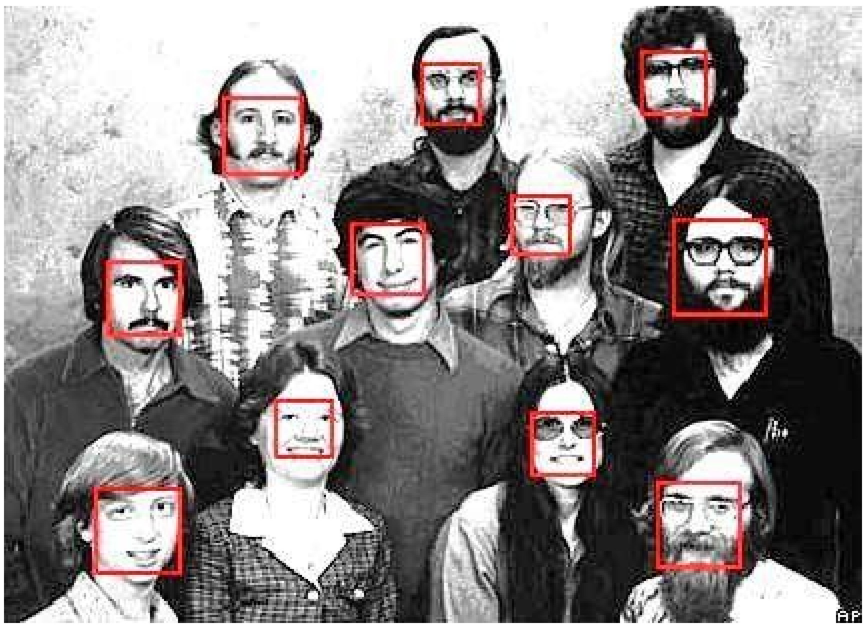
- Reconstruction:

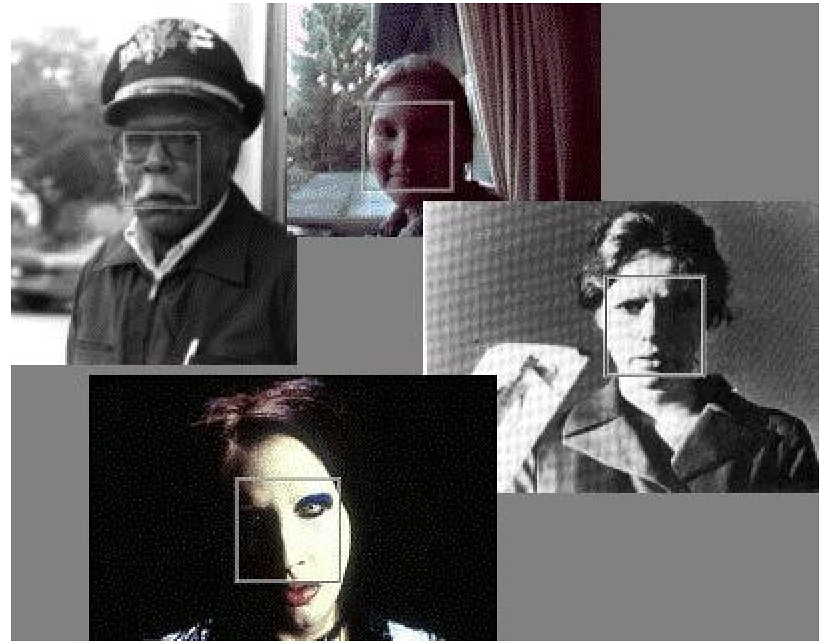
 =  + 

$$\hat{\mathbf{x}} = \boldsymbol{\mu} + w_1\mathbf{u}_1 + w_2\mathbf{u}_2 + w_3\mathbf{u}_3 + w_4\mathbf{u}_4 + \ldots$$

Eigenfaces: Example

Face detection and recognition





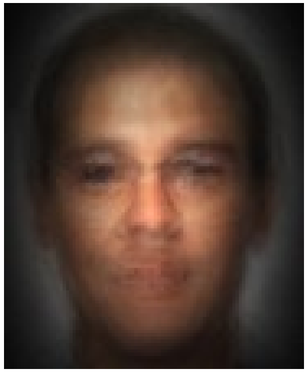
Eigenfaces

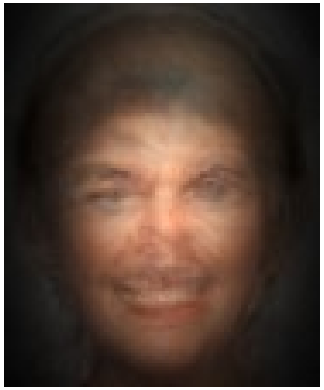
Exemple: Eigenfaces for face detection.

The space defined by the eigenfaces can be viewed as a subspace of the space of images.

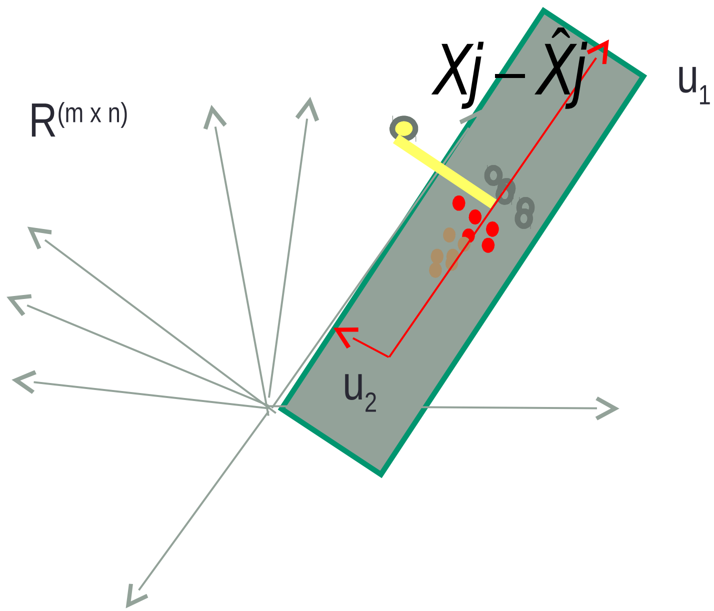
We can always calculate the distance between any image and the face subspace!

And use it as a criterion for deciding whether an image is a face or not.





Eigenfaces How do we detect a face?



If you want to know if an image is a face or not :

a) Project it in the eigenspace defined by the eigenfaces

b) Compute the distance between the position in the original space and the projection into the subspace

c) A face is assumed to be close to the subspace defined by the eigenfaces

Conclusions

- Eigenfaces is a technique to represent the appearance of faces in a compact way.
- **Main idea:** to compute the eigenvalues and eigenvectors of the covariance matrix of the training data.
- **Advantage:** it fast and robust.
- **Limitation:** it works only when the faces are aligned and have the same scale, and pose angle.
- **Applications:** face recognition, face detection, gender recognition, facial expressions recognition, digital photography, monitoring..
- http://www.youtube.com/watch?v=QZmPEl-YU8g