한양대학교 HANYANG UNIVERSITY

Recognition

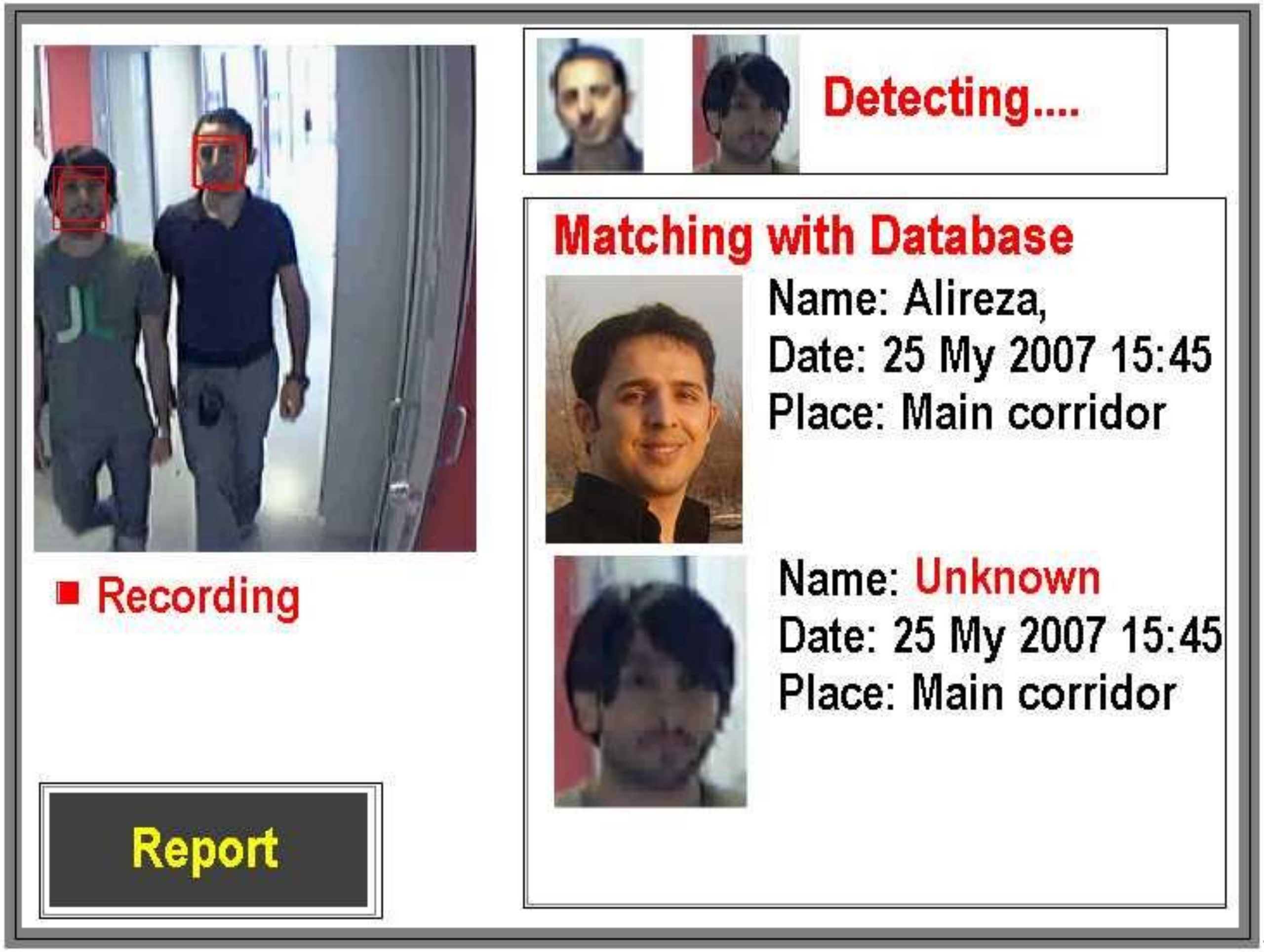
Tae Hyun Kim

This class

- Face recognition
  - Two traditional methods:
    - Eigenfaces
      - PCA
    - Fisherfaces
      - FLD
  - Recent method: DeepFace

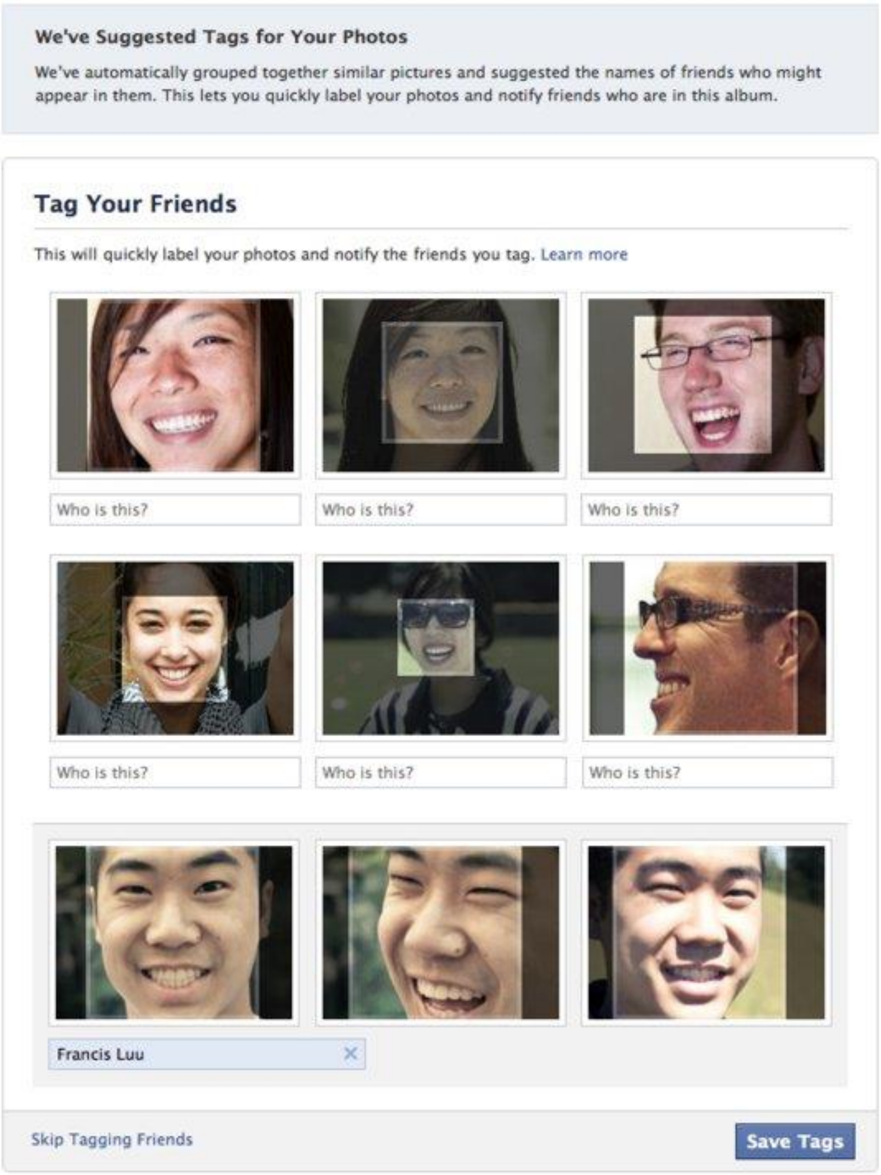
Applications of Face Recognition

- Surveillance
  감시



Applications of Face Recognition

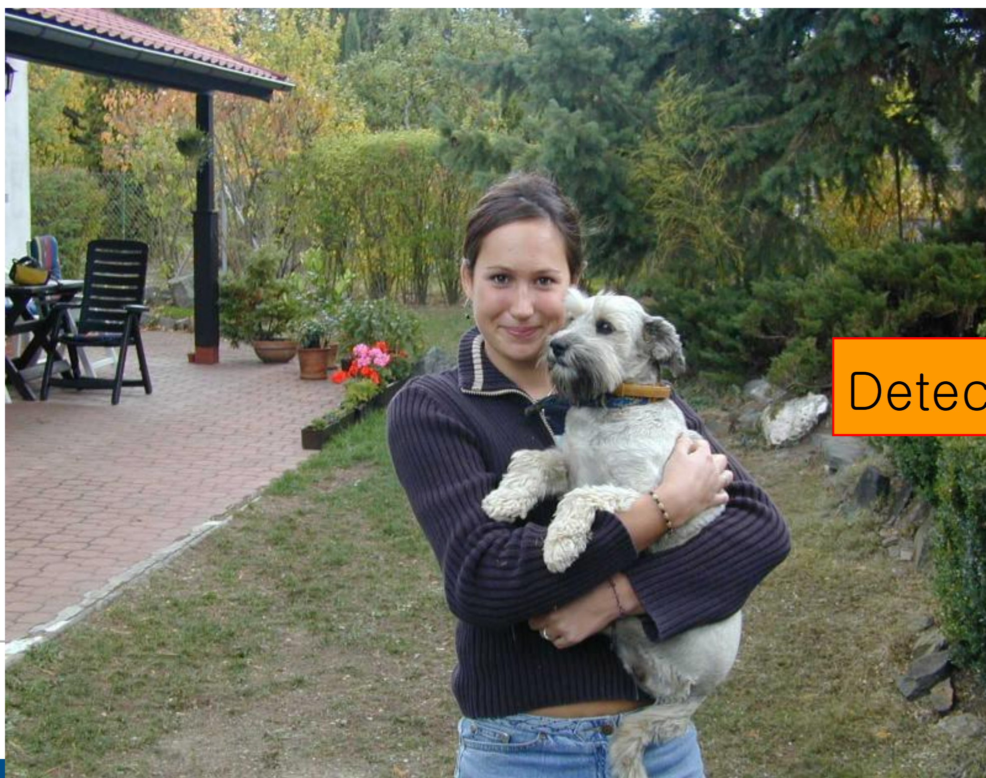
- Facebook friend-tagging with auto-suggest



Face Recognition

- Overall flow

Image / Video → [Face Detection → Recognition] → Identity



Detection →

Candidate

Recognition →

ID return

"Sally"

Typical face recognition scenarios

- Verification: a person is claiming a particular identity; verify whether that is true
  - E.g., security

- Closed-world identification: assign a face to one person from among a known set

- General identification: assign a face to a known person or to "unknown"

What makes face recognition hard?

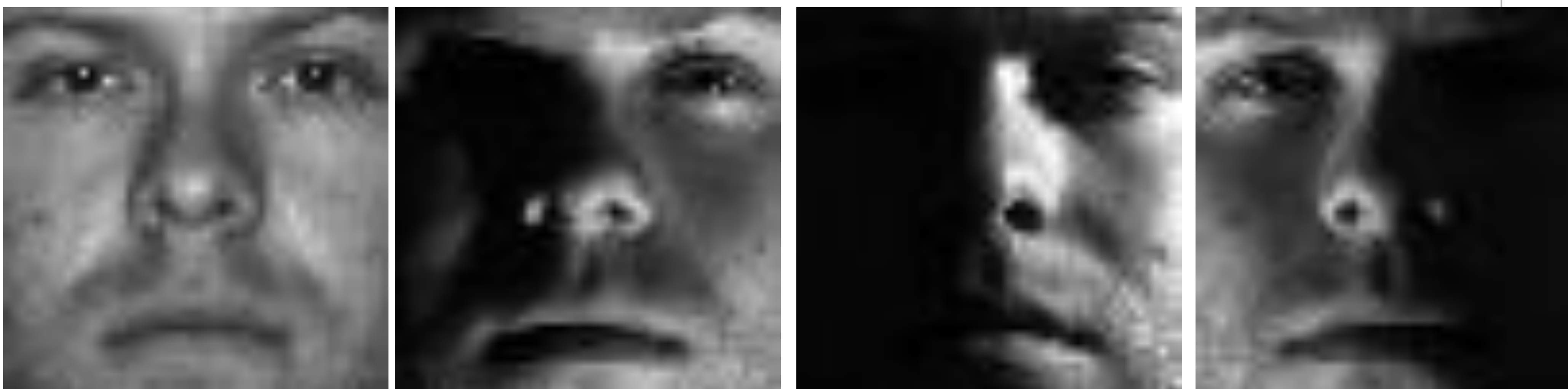
- Expression

표정, 감정에 따라 얼굴 외형이 다름

What makes face recognition hard?

- Lighting

Light source 위치에 따라
다른 느낌의 이미지 얻어짐



동일한 ID return 해야됨
recognition을 어렵게 함

What makes face recognition hard?

- Occlusion 가려진
  폐쇄

What makes face recognition hard?

- Viewpoint

Simple idea for face recognition

1. Treat face image as a vector of intensities

vector (밝기)

$$\mathbf{x}$$

query

1:1 비교 → similarity 大 (distance 小) 찾음

↓

그 face ID를 리턴

2. Recognize face by nearest neighbor in database

DB

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

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

Nearest neighbor classifier

- Label test sample with label of most similar training sample
- ***Good choice if you have few examples*** per class
- ***No training time***, once feature representation is determined

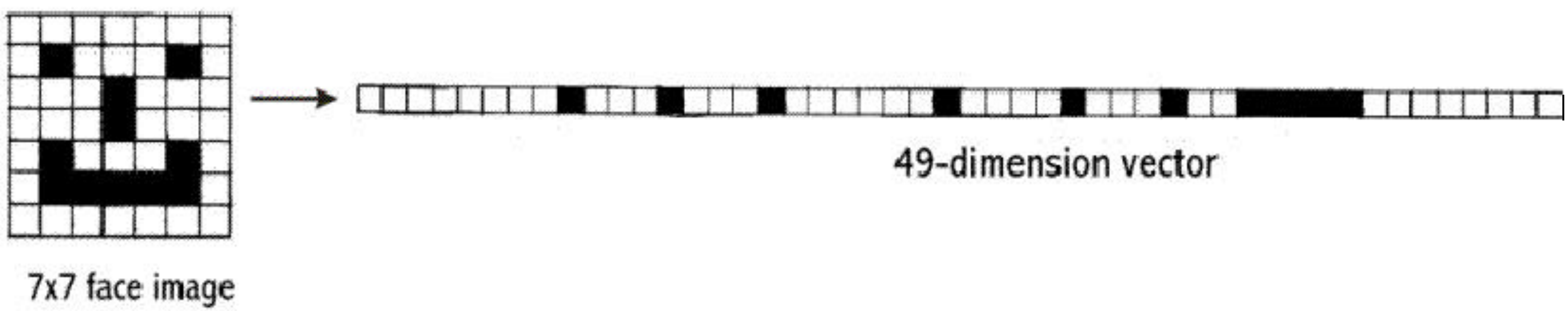
① 한 element당 한 이미지 밖에 없을때

② 백 그라운드에서 training 할 여건이 안될때

이건 제한된 상황에서 Nearest neighbor classifier를 쓸수있다

The space of all face images

- Images as high dimensional vector
  - 100x100 facial image = 10,000 dimensions (밝기값)
  - Slow and lots of storage



The space of all face images

- Very few 10,000-dimensional vectors are valid face images
  - Face images are highly correlated

Redundant information

중복되는게 많음

정보가 redundant.
거의 똑같음

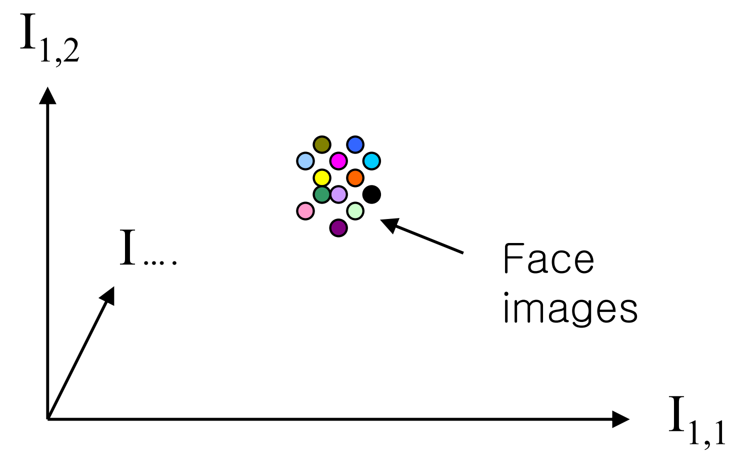


Image space (high dimensional space of all possible images)

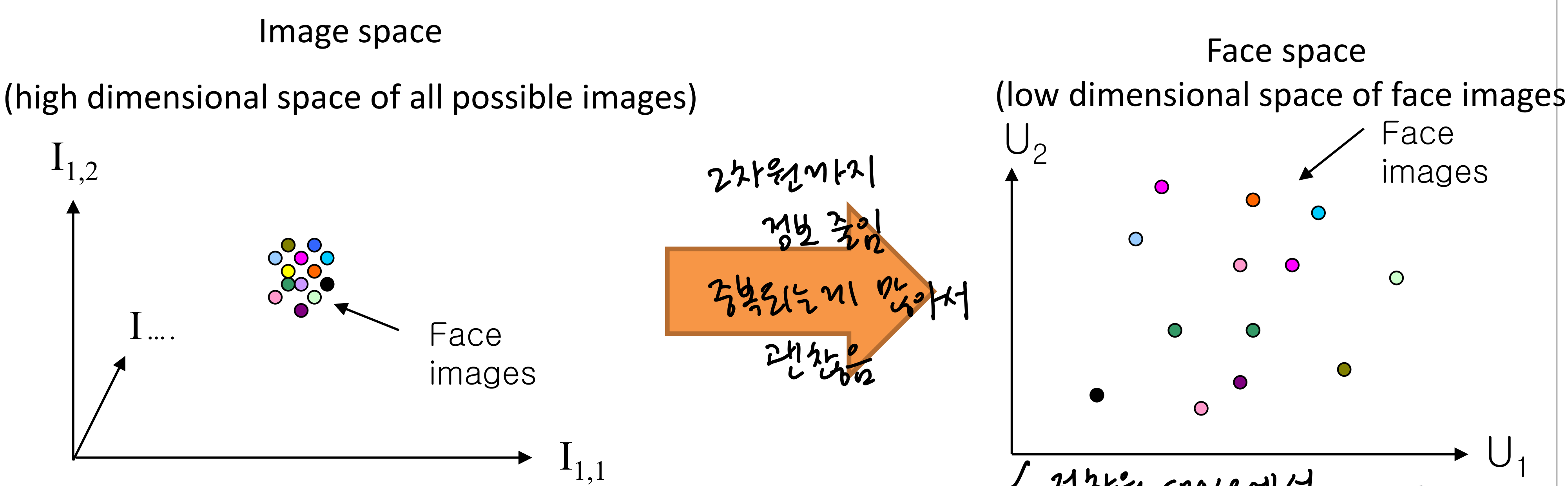
<redundancy 제거 해보면 어떨까?>

→ dimension reduction ex) PCA

- How to model the subspace of face images?

Transform Face images to a 'Face Space'

- Linear transformation that maps data from a high dimensional space to a lower dimensional sub-space.



<저차원 space에서 recognition 해보기>

Principal Component Analysis (PCA)

- Goal of PCA

  다시 X로 되돌아갔을 때 최대한 비슷해지는 y

  – Reduce the dimensionality of the data while retaining as much information as possible in the original dataset
  – Significant improvements can be achieved by mapping the data into a *lower-dimensional sub*-space.

$$\mathrm{x} = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_N \end{bmatrix} \xrightarrow[\text{transform}]{\text{Dimensionality reduction}} \mathrm{y} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_K \end{bmatrix}$$

$$K \ll N$$

Principal Component Analysis (PCA)

- How can we determine the best low dimensional subspace?

$$X = v_1 a_1 + v_2 a_2 + \cdots v_N a_N$$ weighted sum

$a_1, a_2, \cdots a_N$: X space의 basis

줄인 차원으로 reconstruction $$\hat{X} = u_1 b_1 + u_2 b_2 + \cdots u_K b_K$$

적은 basis로 만든 data, $\hat{X} \approx X$

u,b 찾는게 pca

where $K \ll N$

- Information loss
  - Dimensional reduction implies information loss!
  - PCA preserves as much information as possible by minimizing;
    - $\boxed{||X - \hat{X}||}$

Principal Component Analysis (PCA)

- Methodology
  - Given: M data points $\mathbf{x_1}, \dots, \mathbf{x_M}$ are $N \times 1$ vectors

Step 1: compute mean, $\mu = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{x}_i$

평균

Step 2: subtract the mean; $\boldsymbol{\Phi}_i = \boldsymbol{x}_i - \mu$

difference

Step 3: compute covariance, $C = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\Phi}_i \boldsymbol{\Phi}_i^T$

covariance matrix
↓
decomposition

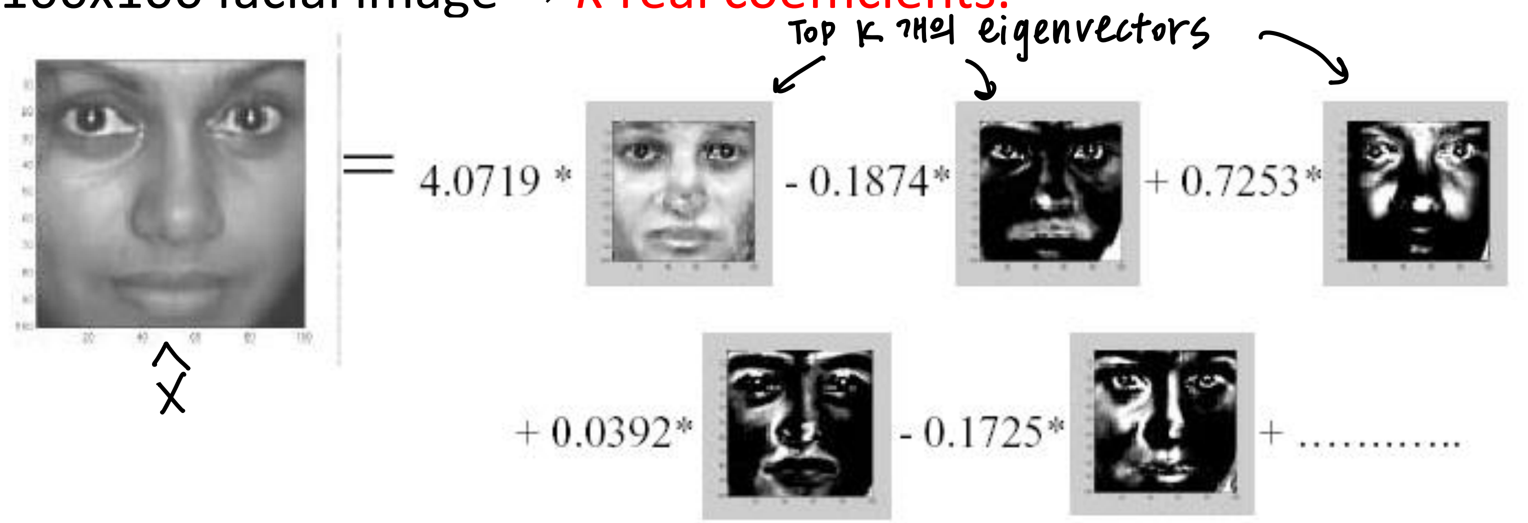
Step 4: compute eigenvectors of C

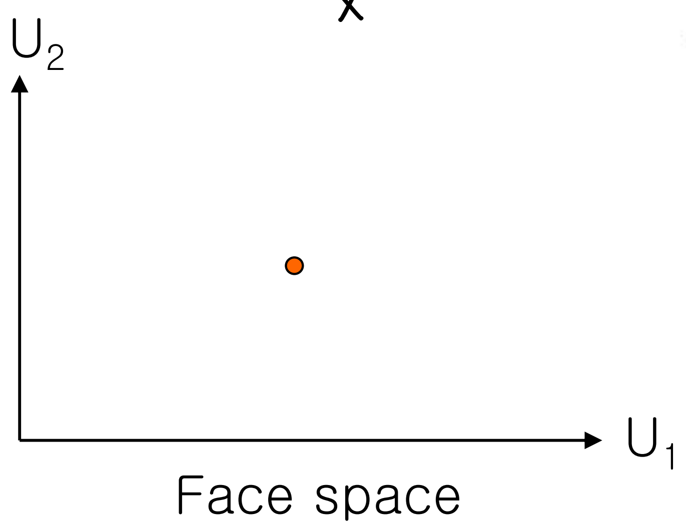
Step 5: keep K eigenvectors corresponding to K largest eigen values $\rightarrow b_1, b_2, \cdots, b_K$ Top k eigenvector만 저장

Now, we can approximate $\boldsymbol{x}_i$ with weighted sum of these K eigenvectors!

Transform into 'Face Space'

- Face representation with top *K* eigenvectors
  - 100x100 facial image → *K* real coefficients.





Transform known face to (low-dimensional) face space

Principal Component Analysis (PCA)

'물리적인 의미'

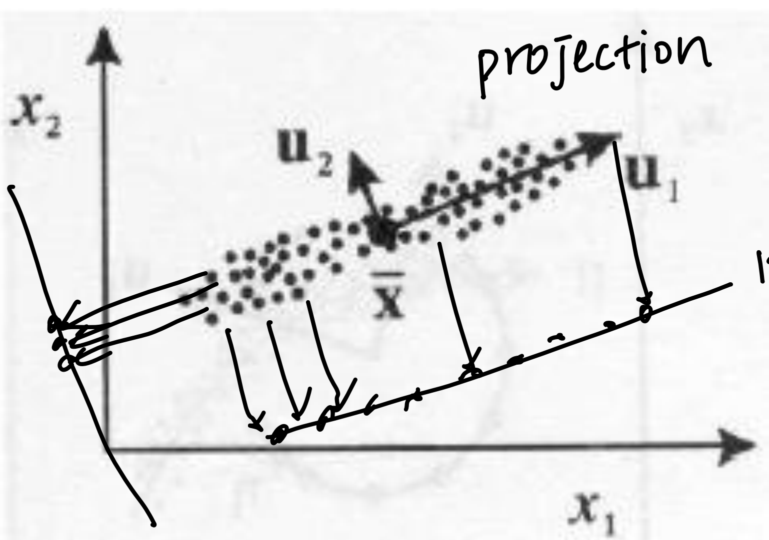
- Why eigen decomposition?
  - Geometric interpretation
    - Direction that ***maximizes the variance of the projected data***
    - Directions are determined by the eigenvectors of the covariance matrix corresponding to the largest eigenvalues
    - The magnitude of the eigenvalues corresponds to the variance of the data along the eigenvector directions

k개의 eigenvector

$$A x = \lambda \underline{x}$$

ev vec

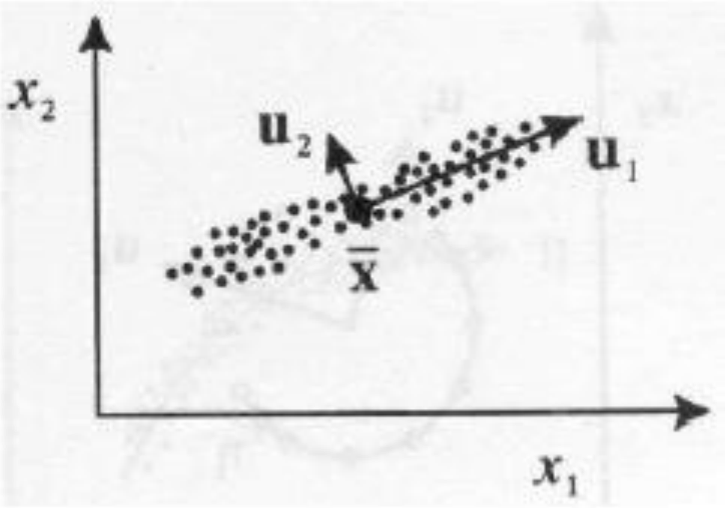
A를 선형변환



1차원 data의
variance가 가장 큰 방향으로
projection

Principal Component Analysis (PCA)

- Why eigen decomposition? (math)
  - Data projection
    - Projection vectors are principal components of face imgs
    - New set of features: $u(\mathbf{x}_i) = \mathbf{u}^T(\mathbf{x}_i - \boldsymbol{\mu})$ where $\mathbf{u}$ in R^N



- PCA goal
  - Choose ***unit*** vector **u** captures the most ***data variance***

Principal Component Analysis (PCA)

- Why eigen decomposition? (math)
  - Direction that maximizes the variance of the projected data

$X = x_1 - \mu$

$E(X) = 0$

$$maximize\ \frac{1}{M}\sum_{i=1}^{M}\Big(\underbrace{\boldsymbol{u}^T}_{(1\times 10{,}000)}\ \times\ (\overbrace{\underset{\text{original}}{\boldsymbol{x}_i} - \underset{\text{mean}}{\mu}}^{(10{,}000\times 1)})\Big)\Big(\boldsymbol{u}^T(\boldsymbol{x}_i-\mu)\Big)^T$$

$= 1\times 1$

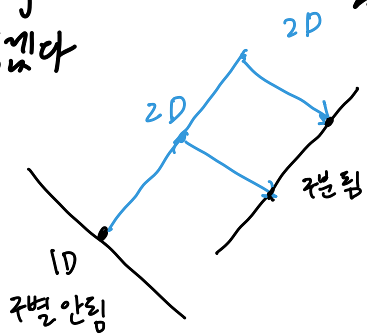
1차원

projected data

이 eigenvector를 이용해서 proj 하면 data의 variance가 굉장히 넓게 유지된다 그 측면에서 optimal한 dimension reduction 방법이다

projection된 point의 variance를 maximize

차원 줄인 후 original 정보를 많이 가졌음 좋겠다



정보 많이 유지하는 방향으로 projection

$$= \boldsymbol{u}^T\left[\frac{1}{M}\sum_{i}^{j}(\boldsymbol{x}_i-\mu)(\boldsymbol{x}_i-\mu)^T\right]\boldsymbol{u}$$

Covariance matrix of data

$$= \boldsymbol{u}^T\Sigma\boldsymbol{u}$$

Practical Issue #1

- Covariance matrix is huge (e.g., 100x100 facial image)

  $C = \boldsymbol{AA}^T$ is very ***huge***, where $\boldsymbol{A} = [(x_1-\mu) \quad \cdots \quad (x_m-\mu)]$

  Calculate eig($\mathbf{AA}^\mathrm{T}$) is not practical

  → eigenvector를 구하기엔 너무 큼 → $C^T (= A^TA)$의 eigenvector 구함

- But, typically # face images M << N, thus
  - Find eigenvectors of $\mathbf{A}^\mathrm{T}\mathbf{A}$ $\overset{500 \times 500}{(M \times M)}$ instead of $\mathbf{AA}^\mathrm{T}$ $\overset{10,000 \times 10,000}{(N \times N)}$
  - Relationship b.t.w. eigenvectors of $\mathbf{A}^\mathrm{T}\mathbf{A}$ & $\mathbf{AA}^\mathrm{T}$

eigenvector u 찾기

$$\boldsymbol{A}^T\boldsymbol{A}v_i = \lambda_i v_i \rightarrow \boldsymbol{AA}^T\boldsymbol{A}v_i = \lambda_i \boldsymbol{A}v_i \rightarrow \boldsymbol{AA}^T \underbrace{u_i}_{\text{C의 eigenvector 중 1}} = \lambda_i u_i$$

v_i 찾고 A 곱해주면 u_i가 됨

- Same eigenvalues, transformed eigenvectors (v_i-> $\boldsymbol{A}v_i$)
- Keep only *K* out of *M* eigenvectors corresponding to *K* largest eigenvalues of $\mathbf{A}^\mathrm{T}\mathbf{A}$
- **Normalize** *K* eigenvectors $\boldsymbol{A}v_i$ to unit length
  - $\|\boldsymbol{A}v_i\| = \|u_i\| = 1$

Practical Issue #2

- preprocessing 필요

(밝은 → 어둡게 / 어둡게 → 밝게) 톤 맞춰줌 : Normalization

- Standardization
  - The principal components are dependent on the ***range*** of values
    - If some variables have a large variance and some small, PCA will be biased toward the large ones
    - Should ***normalize*** the data prior to using PCA
  - A common standardization
    - Transform all the data to have **zero mean** and **unit standard deviation**: $\frac{x_i - \mu}{\sigma}$

Practical Issue #3

- How to choose *K* (i.e., number of principal components)
  - To choose *K*, use the following criterion:
    - $\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} > Thr.$ (e.g., 0.9 or 0.95)

      N개 중 Top K eigenvector로 ...
  - In this case, we say that we "preserve" 90% or 95% of the information in our data
  - If *K*=*N*, then we "preserve" 100% of the information in our data

Eigenfaces example

- *M* Training images
  - $\mathbf{x}_1,\ldots,\mathbf{x}_M$

Eigenfaces example

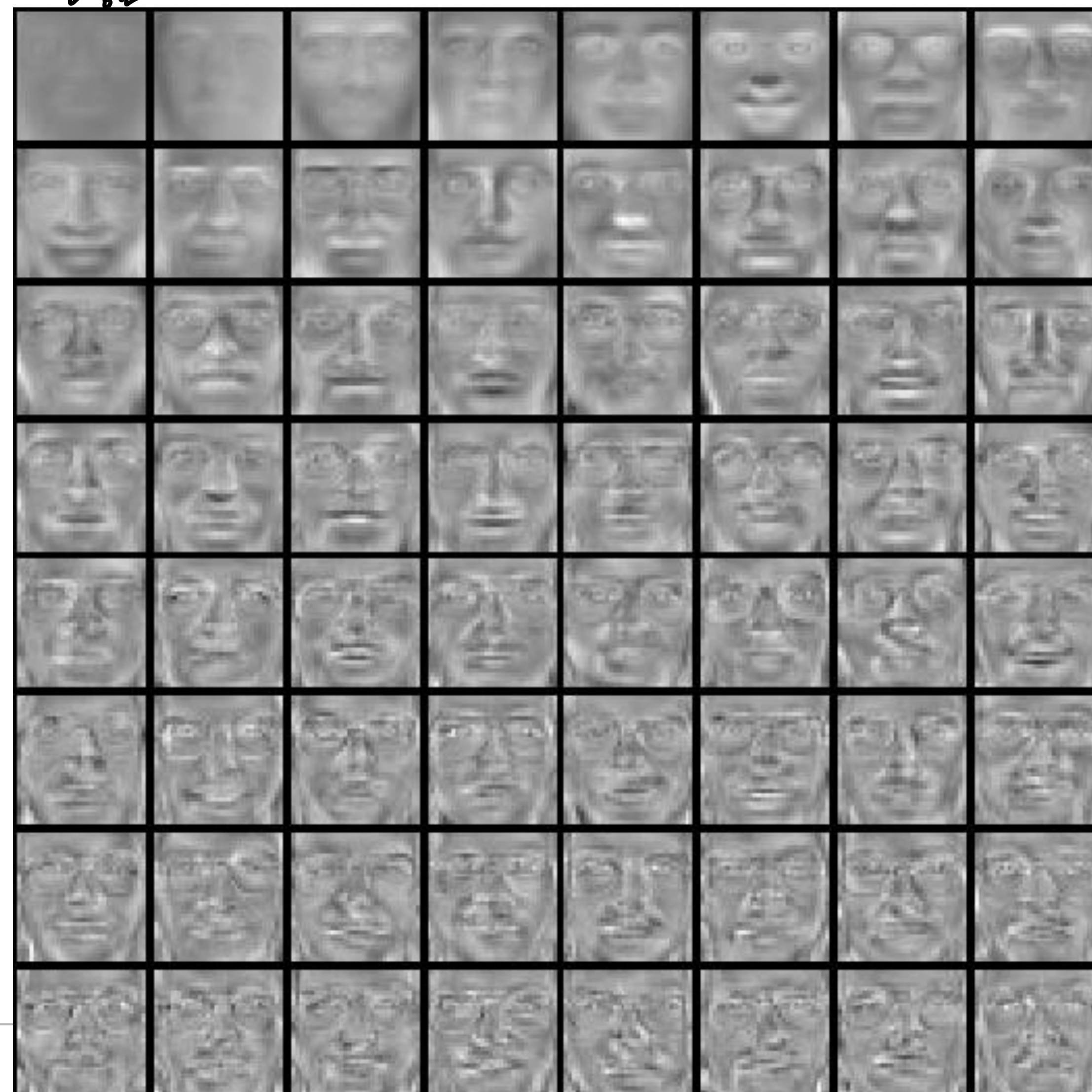
- Top *K* eigenvectors

$u_i = (\underbrace{0.1, 0.5, \cdots}_{10,000})$

Mean: μ

M개의 평균

제일 중요



Representation and reconstruction

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

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

$$= \quad w_1, \ldots, w_k$$ → 줄어든 정보

- Reconstruction:

k개 weighted sum

 =  + 

$$\hat{x} = \mu + w_1u_1 + w_2u_2 + w_3u_3 + w_4u_4 + \ldots$$

Reconstruction

- Reconstruction err.

$P = 4$

$P = 200$

$P = 400$

After computing eigenfaces using 400 face images from ORL face database

Recognition with eigenfaces

- Process labeled training images
  - Find mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
  - Find *K* principal components (eigenvectors of $\boldsymbol{\Sigma}$) $\mathbf{u}_1,\ldots,\mathbf{u}_k$
  - Project each training image $\mathbf{x}_i$ onto subspace spanned by principal components:
    - $(w_{i1},\ldots,w_{ik}) = (\mathbf{u}_1^T(\mathbf{x}_i - \boldsymbol{\mu}), \ldots, \mathbf{u}_k^T(\mathbf{x}_i - \boldsymbol{\mu}))$
- Given target image $\mathbf{x}$
  - Project onto subspace (coefficients):
    $(w_1,\ldots,w_k) = (\mathbf{u}_1^T(\mathbf{x} - \boldsymbol{\mu}), \ldots, \mathbf{u}_k^T(\mathbf{x} - \boldsymbol{\mu}))$
  - Classify as closest training face in *K*-dimensional subspace w.r.t.
    - $\sum_{j=1}^{K} \|w_{ij} - w_j\|$

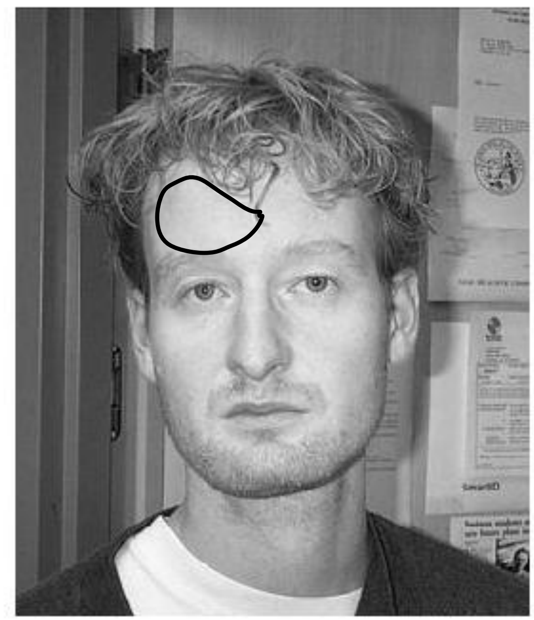
PCA summary

- General dimensionality reduction technique
- Preserves most of variance with a compact representation
  - Lower storage requirements (eigenvectors + a few numbers per face)
  - Faster matching
- What are the problems for face recognition?

Limitations of Face Reconstruction with PCA

- Global appearance method
  - Not robust to misalignment, background variation

align이 안돼서 조금밖에 못날림



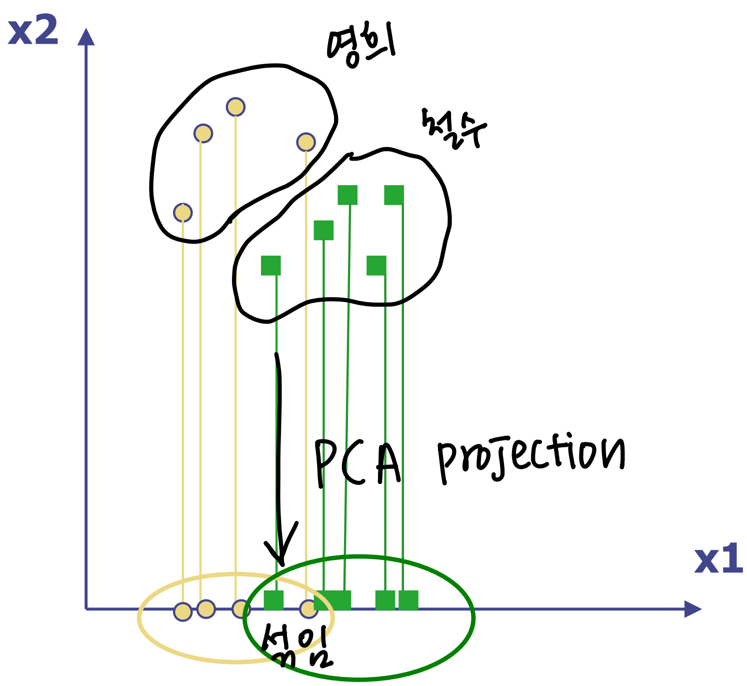
dataset 가공 중요한 정보만 남기기 → eigenvector 수 줄일수 있음

Limitations of Face Reconstruction with PCA

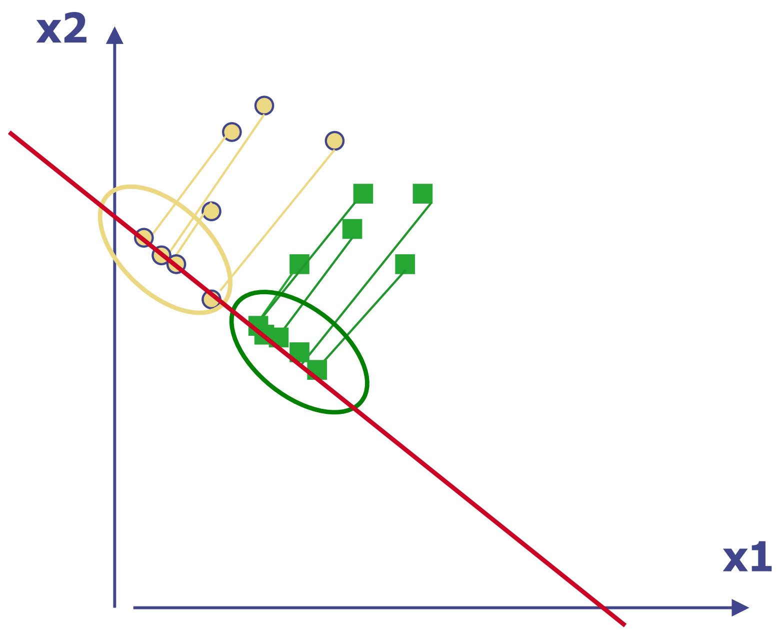
- ***Background*** changes cause problems
  - De-emphasize the outside of the face (e.g., by multiplying the input image
    덜강조하다
    by a 2D Gaussian window centered on the face).
- ***Light*** changes degrade performance
  - Light normalization helps
- Sensitive to changes of ***face size***
  - Multi-scale eigenspaces
  - Scale input image to multiple sizes
- Weak to ***face orientation***
  - Plane rotations are easier to handle
  - Out-of-plane rotations are more difficult to handle

Limitations of Classification with PCA

- The direction of maximum variance is ***not good for classification***

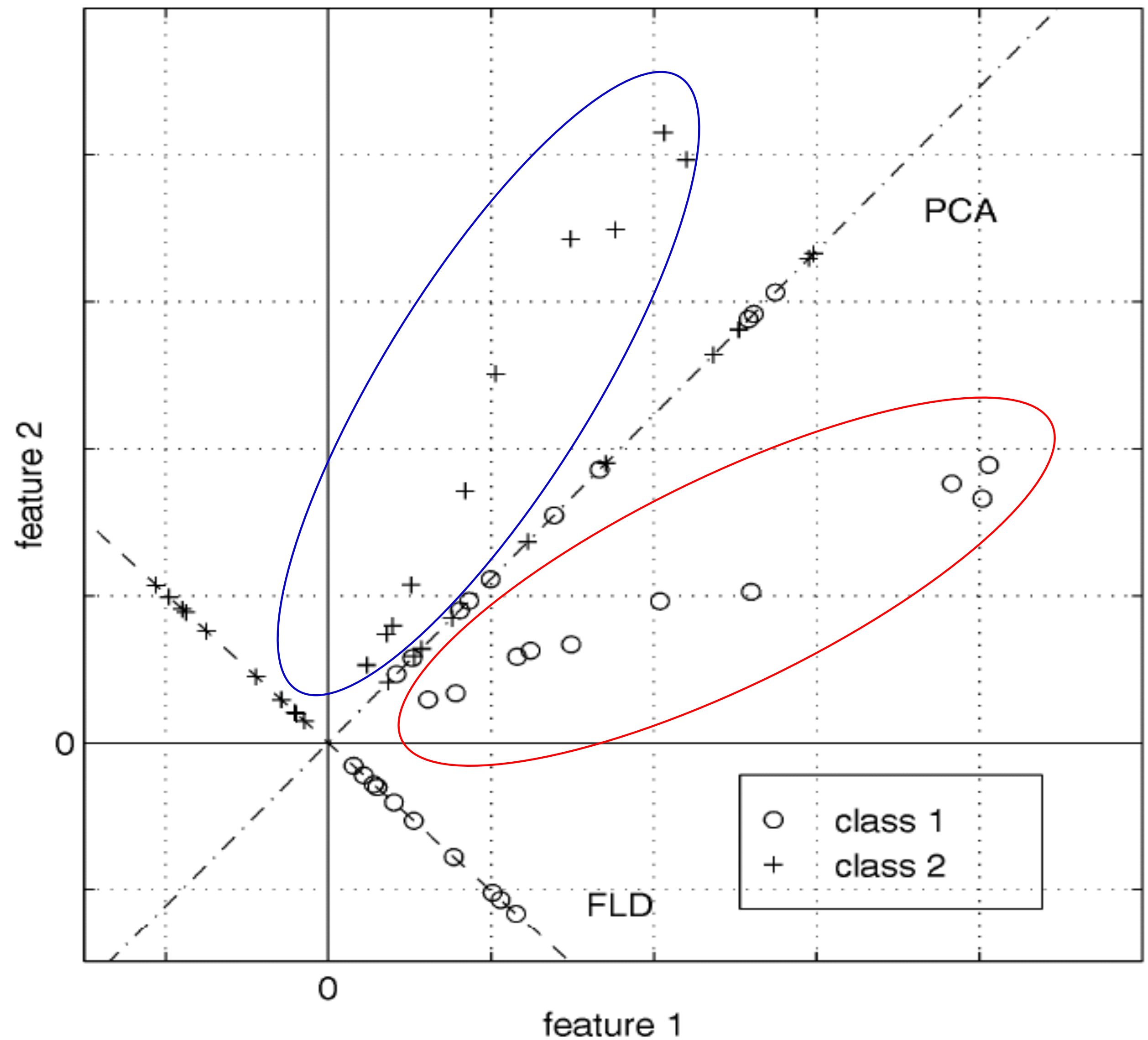


Poor Projection



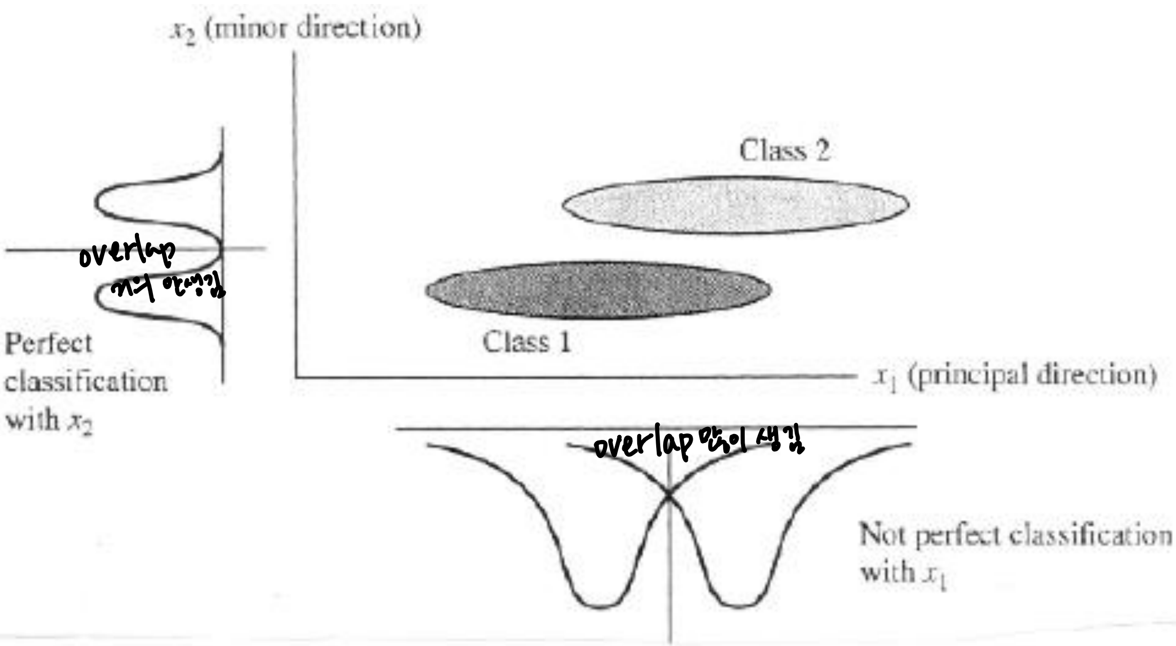
Good

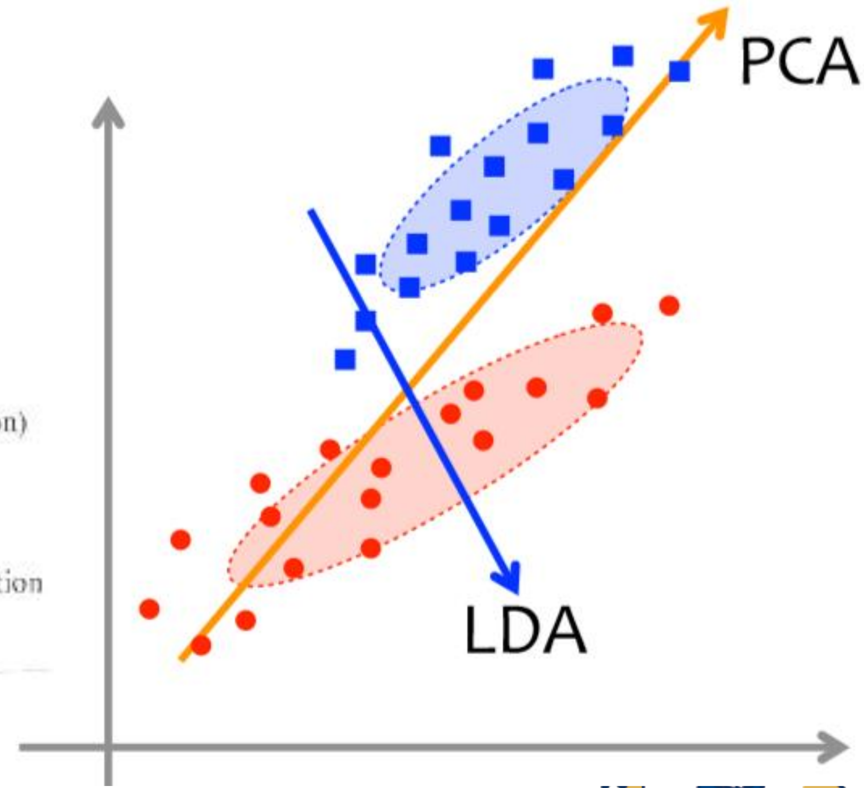
Limitations of Classification with PCA



Fisher Linear Discriminant (FLD/LDA)

- Linear discriminant analysis (LDA)
  - Maximizes the ***distances between classes*** of data
  - Approximation (PCA, max. variance) vs. classification (FLD/LDA).





Fisher Linear Discriminant (FLD/LDA)

- Linear discriminant analysis
  - Within class scatter and between class scatter
- Scatter matrices
  - Sample mean: μ
  - Sample mean within a class: μ_i
  - Scatter of class i
    - $S_i = \sum_{x_k \in \chi_i} (x_k - \mu_i)\,(x_k - \mu_i)^T$
  - Within class scatter
    - $S_W = \sum_{i=1}^{c} S_i$
  - Between class scatter 클래스 간 거리
    - $S_B = \sum_{i=1}^{c} \underline{N_i} (\mu_i - \mu)(\mu_i - \mu)^T$

      ↳ i 클래스에서 데이터 개수

Fisher Linear Discriminant (FLD/LDA)

- Linear discriminant analysis
  - Within class scatter and between class scatter
- After projection
  - Projected data
    - $y_k = W^T x_k$ original data
  - Between class scatter
    - $\tilde{S}_B = W^T S_B W$
  - Within class scatter
    - $\tilde{S}_W = W^T S_W W$
- LDA goal

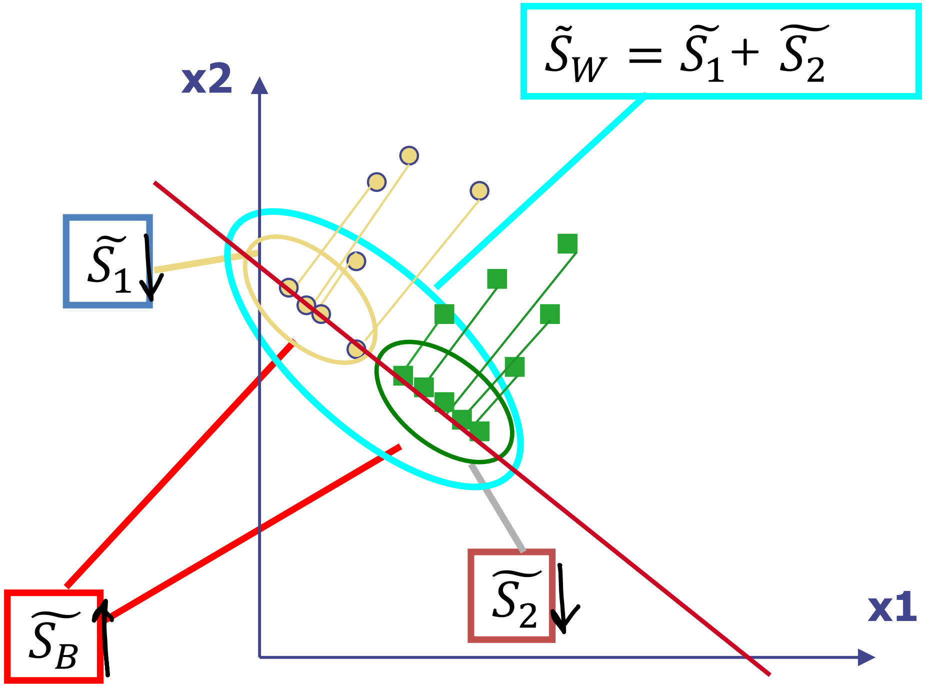
목표: W 찾기

가장 키우는 W가 LDA 관점에서 optimal 한 value 다

  - $W_{opt} = \arg\max_W \frac{|\tilde{S}_B|}{|\tilde{S}_W|} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|}$
  - Solution: Generalized Eigenvectors
    - $S_W^{-1} S_B w_i = \lambda_i w_i \qquad i = 1, \ldots, m$

Fisher Linear Discriminant (FLD/LDA)

- Linear discriminant analysis
  - Illustration after projection



State-of-the-art Face Recognizers

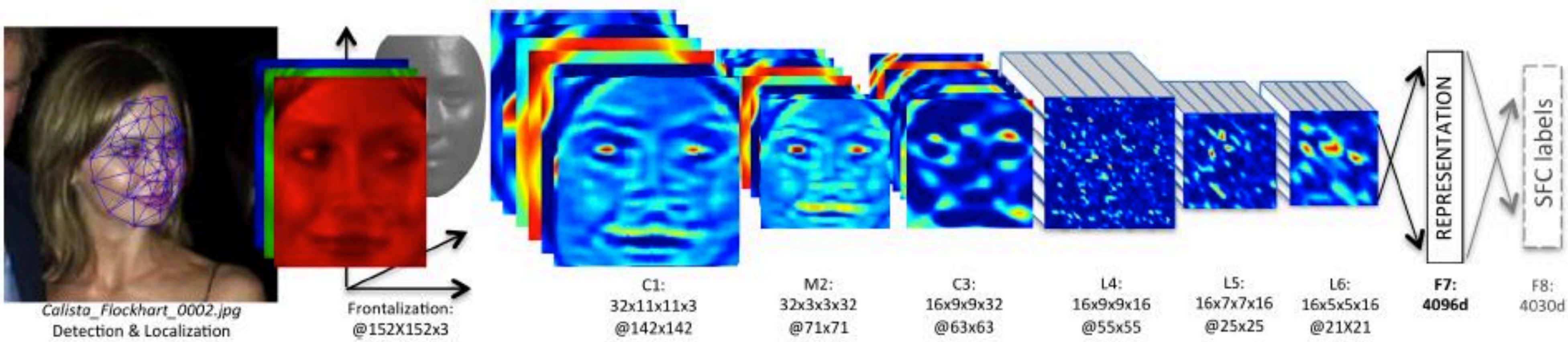
- Most recent research focuses on “faces in the wild”
  - Recognizing faces in normal (usual) photos
  - Classification: assign identity to face
  - Verification: say whether two people are the same
- Important steps
  1. Detect
  2. Align
  3. Represent
  4. Classify

DeepFace: Closing the Gap to Human-Level Performance in Face Verification

Yaniv Taigman   Ming Yang   Marc'Aurelio Ranzato   Lior Wolf

Facebook AI Research
Menlo Park, CA, USA
{yaniv, mingyang, ranzato}@fb.com

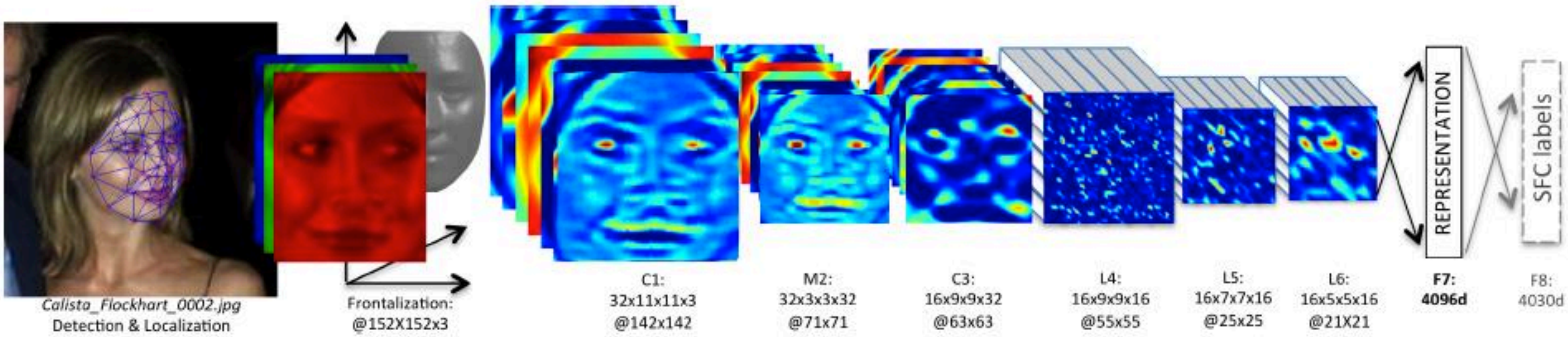
Tel Aviv University
Tel Aviv, Israel
wolf@cs.tau.ac.il



[DeepFace: Closing the Gap to Human-Level Performance in Face Verification]

Taigman, Yang, Ranzato, & Wolf (Facebook, Tel Aviv), CVPR 2014

Train DNN classifier on aligned faces



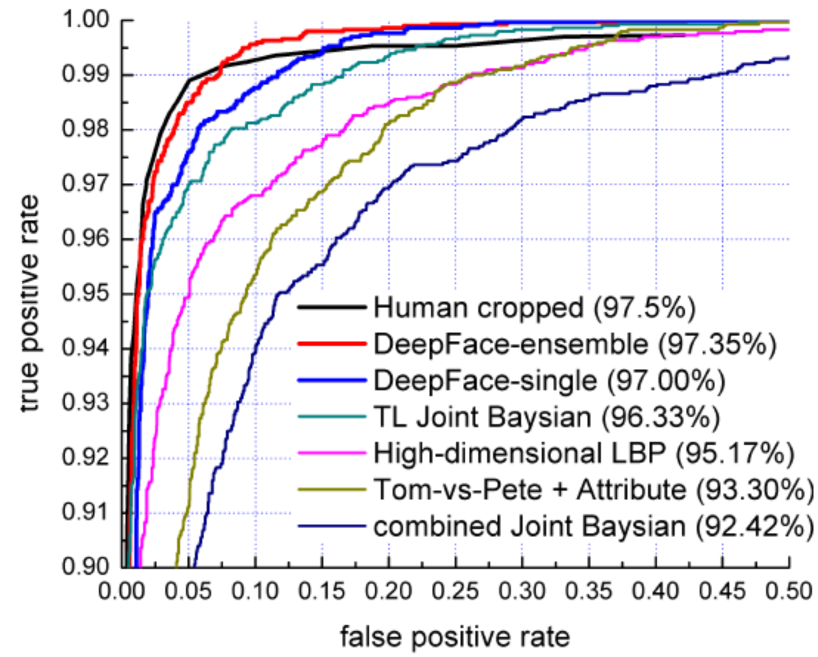
Architecture (deep neural network classifier)

- Two convolutional layers (with one pooling layer)
- 3 locally connected and 2 fully connected layers
- > 120 million parameters

Train on dataset with 4400 individuals, ~1000 images each

- Large dataset
- Train to identify face among set of possible people

Results: Labeled Faces in the Wild Dataset



Method	Accuracy ± SE	Protocol
Joint Bayesian [6]	0.9242 ±0.0108	restricted
Tom-vs-Pete [4]	0.9330 ±0.0128	restricted
High-dim LBP [7]	0.9517 ±0.0113	restricted
TL Joint Bayesian [5]	0.9633 ±0.0108	restricted
DeepFace-single	**0.9592** ±0.0029	unsupervised
DeepFace-single	**0.9700** ±0.0028	restricted
DeepFace-ensemble	**0.9715** ±0.0027	restricted
DeepFace-ensemble	**0.9735** ±0.0025	unrestricted
Human, cropped	0.9753	

- Performs similarly to humans!

(note: humans would do better with uncropped faces)

Experiments show that alignment is crucial (0.97 vs 0.88) and that deep features help (0.97 vs. 0.91)

Summary

- PCA is a dimensionality reduction technique
  - But not ideal for discrimination (classification)
- Simple nearest neighbor classifiers can be effective, but features matter
  - FLD, Deep networks
- Face recognition works very well under controlled environments
  - since 2006
- Perform at human level in uncontrolled settings
  - With recent progress: better alignment, features, more data

Thank you!