Face Recognition

Shenghua Gao

ShanghaiTech University

Face recognition by humans

Face recognition by humans: 20 results (2005)

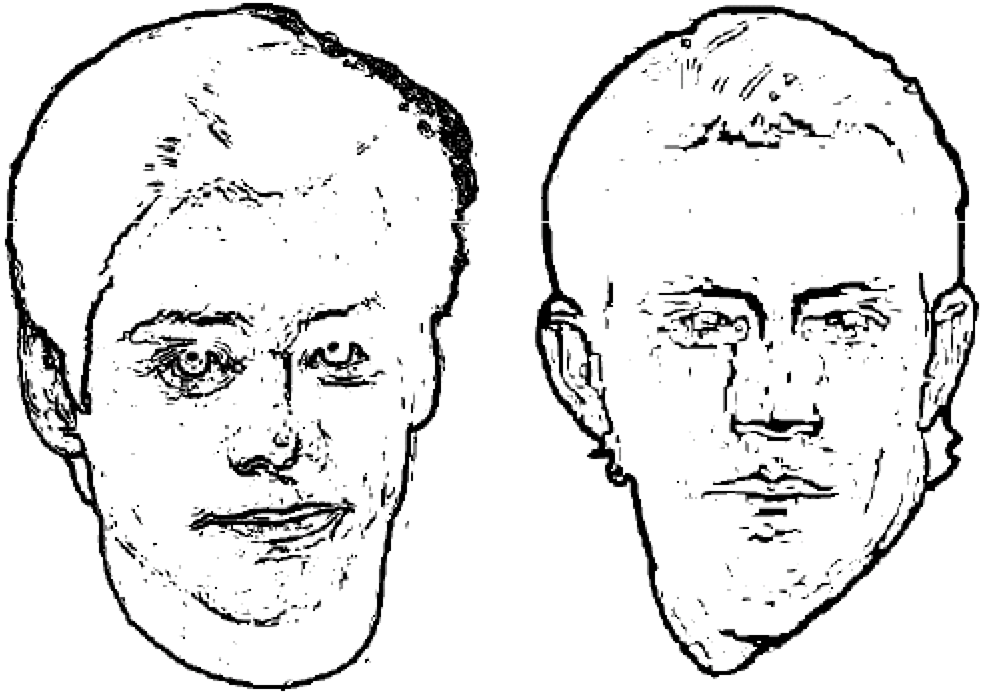
Slides by Jianchao Yang

Result 1

- Humans can recognize faces in extremely low resolution images.

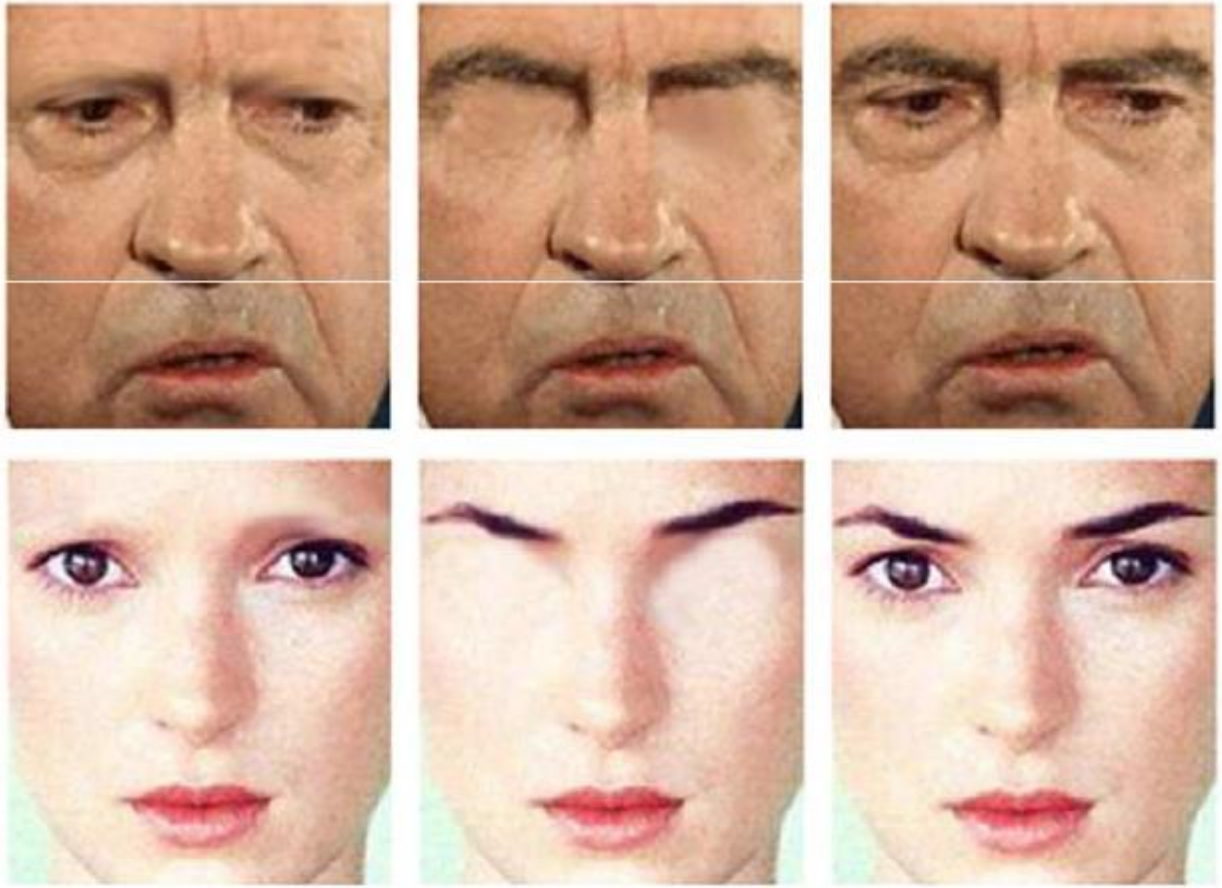
Result 3

- High-frequency information by itself does not lead to good face recognition performance



Result 5

- Eyebrows are among the most important for recognition



Result 8

- Vertical inversion dramatically reduces recognition performance

Result 17: Vision progresses from piecemeal to holistic

Age	Correct responses (%)			
	Faces		Houses	
	Upright	Inverted	Upright	Inverted
6	69	64	71	58*†
8	81	67	74	64
10	89	68‡	73	77

Applications of Face Recognition

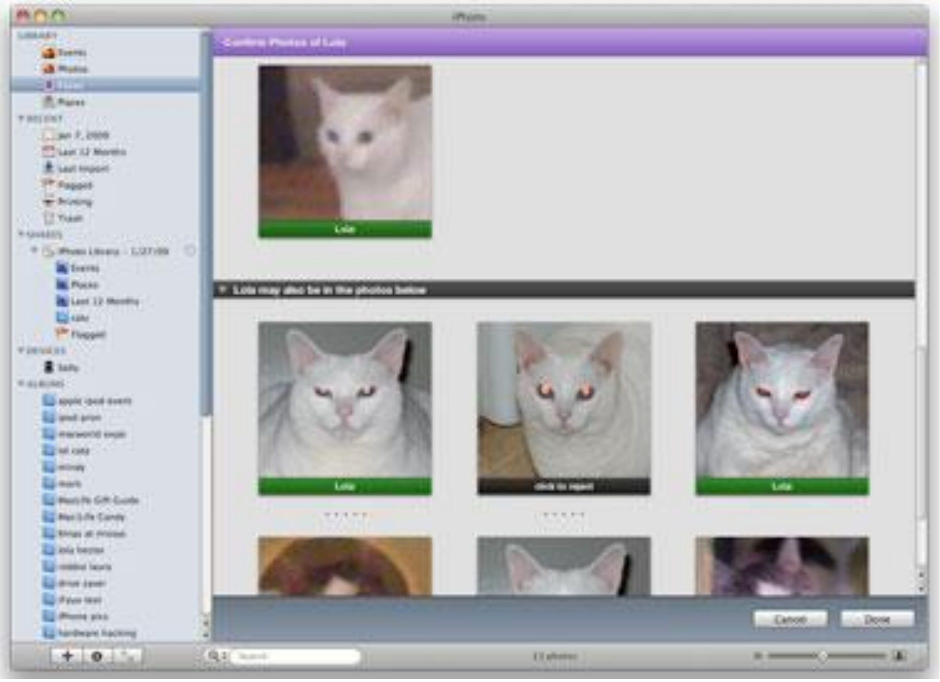
- Surveillance

Applications of Face Recognition

- Album organization: iPhoto 2009(Apple/Microsoft)

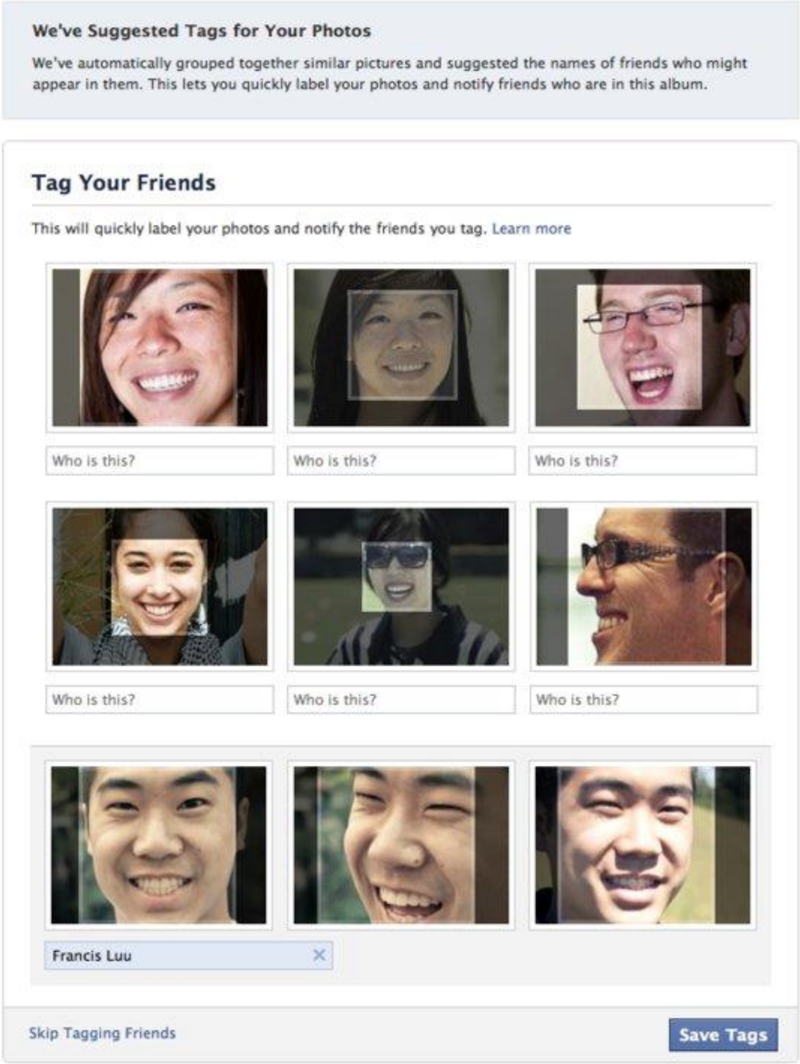
http://www.apple.com/ilife/iphoto/

- Can be trained to recognize pets!



http://www.maclife.com/article/news/iphotos_faces_recognizes_cats

Facebook friend-tagging with auto-suggest
校内网



Face recognition: once you've detected and cropped a face, try to recognize it

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

What makes face recognition hard?

Lighting

What makes face recognition hard?

Occlusion

What makes face recognition hard?

Viewpoint

Simple idea for face recognition

1. Treat face image as a vector of intensities

$$\mathbf{x}$$

2. Recognize face by nearest neighbor in database

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

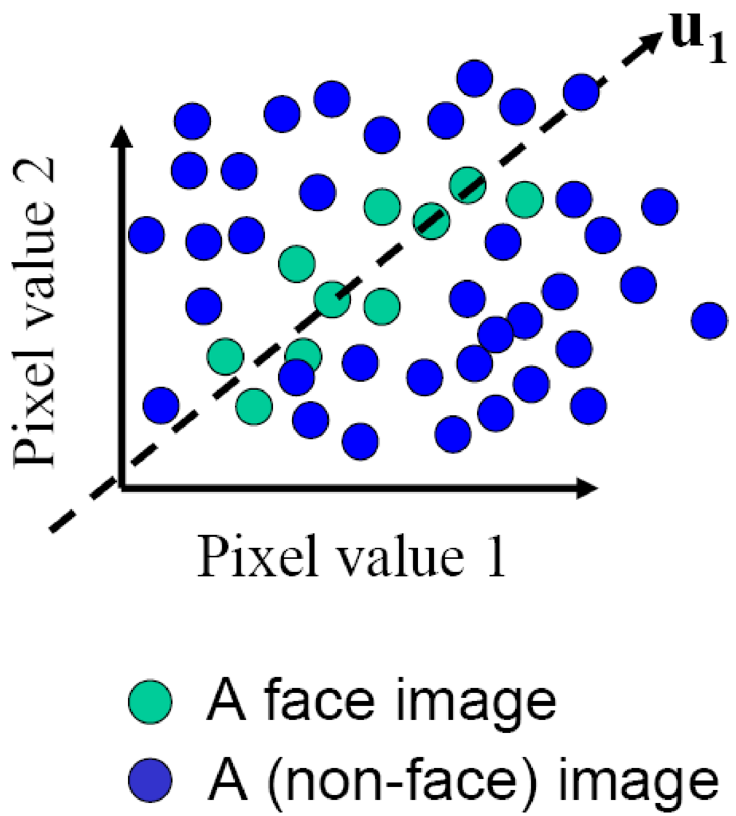
$$k = \underset{k}{\operatorname{argmin}} \|\mathbf{y}_k - \mathbf{x}\|$$

The space of all face images

- When viewed as vectors of pixel values, face images are extremely high-dimensional
  - 100x100 image = 10,000 dimensions
  - Slow and lots of storage
- But very few 10,000-dimensional vectors are valid face images
- We want to effectively model the subspace of face images

The space of all face images

- Eigenface idea: construct a low-dimensional linear subspace that best explains the variation in the set of face images



Principal Component Analysis (PCA)

- Given: N data points $\mathbf{x_1, \ldots ,x_N}$ in R^d
- We want to find a new set of features that are linear combinations of original ones:

$$u(\mathbf{x}_i) = \mathbf{u}^T(\mathbf{x}_i - \boldsymbol{\mu})$$

  ($\boldsymbol{\mu}$: mean of data points)

- Choose unit vector **u** in R^d that captures the most data variance

Forsyth & Ponce, Sec. 22.3.1, 22.3.2

Principal Component Analysis

- Direction that maximizes the variance of the projected data:

$$\text{Maximize} \quad \frac{1}{N}\sum_{i=1}^{N} \underbrace{\mathbf{u}^{\mathrm{T}}(\mathbf{x}_i-\mu)}_{\text{Projection of data point}}(\mathbf{u}^{\mathrm{T}}(\mathbf{x}_i-\mu))^{\mathrm{T}} \quad \text{subject to } ||\mathbf{u}||=1$$

$$= \quad \mathbf{u}^{\mathrm{T}}\underbrace{\left[1/N\sum_{i=1}^{N}(\mathbf{x}_i-\mu)(\mathbf{x}_i-\mu)^{\mathrm{T}}\right]}_{\text{Covariance matrix of data}}\mathbf{u}$$

$$= \quad \mathbf{u}^{\mathrm{T}}\Sigma\mathbf{u}$$

The direction that maximizes the variance is the eigenvector associated with the largest eigenvalue of Σ (can be derived using Raleigh's quotient or Lagrange multiplier)

Implementation issue

- Covariance matrix is huge (M^2 for M pixels)
- typically # examples << M
- Simple trick
  - $\mathbf{X}$ is MxN matrix of normalized training data
  - Solve for eigenvectors $\mathbf{u}$ of $\mathbf{X}^T\mathbf{X}$ instead of $\mathbf{X}\mathbf{X}^T$
  - Then $\mathbf{Xu}$ is eigenvector of covariance $\mathbf{X}\mathbf{X}^T$
  - Need to normalize each vector of $\mathbf{Xu}$ into unit length

Eigenfaces (PCA on face images)

1. Compute the principal components ("eigenfaces") of the covariance matrix

$$\boldsymbol{X} = [(\boldsymbol{x_1} - \boldsymbol{\mu}) \;\; (\boldsymbol{x_2} - \boldsymbol{\mu}) \;\; \ldots \;\; (\boldsymbol{x_n} - \boldsymbol{\mu})]$$
$$[\boldsymbol{U}, \boldsymbol{\lambda}] = \mathrm{eig}(\boldsymbol{X^T X})$$
$$\boldsymbol{V} = \boldsymbol{XU}$$

2. Keep K eigenvectors with largest eigenvalues

$$\boldsymbol{V} = \boldsymbol{V}(:, \mathrm{largest\_eig})$$

3. Represent all face images in the dataset as linear combinations of eigenfaces
   – Perform nearest neighbor on these coefficients

$$\boldsymbol{X_{pca}} = \boldsymbol{V}\left(:, \mathrm{largest}_{\mathrm{eig}}\right)^{\boldsymbol{T}} \boldsymbol{X}$$

M. Turk and A. Pentland, [Face Recognition using Eigenfaces](), CVPR 1991

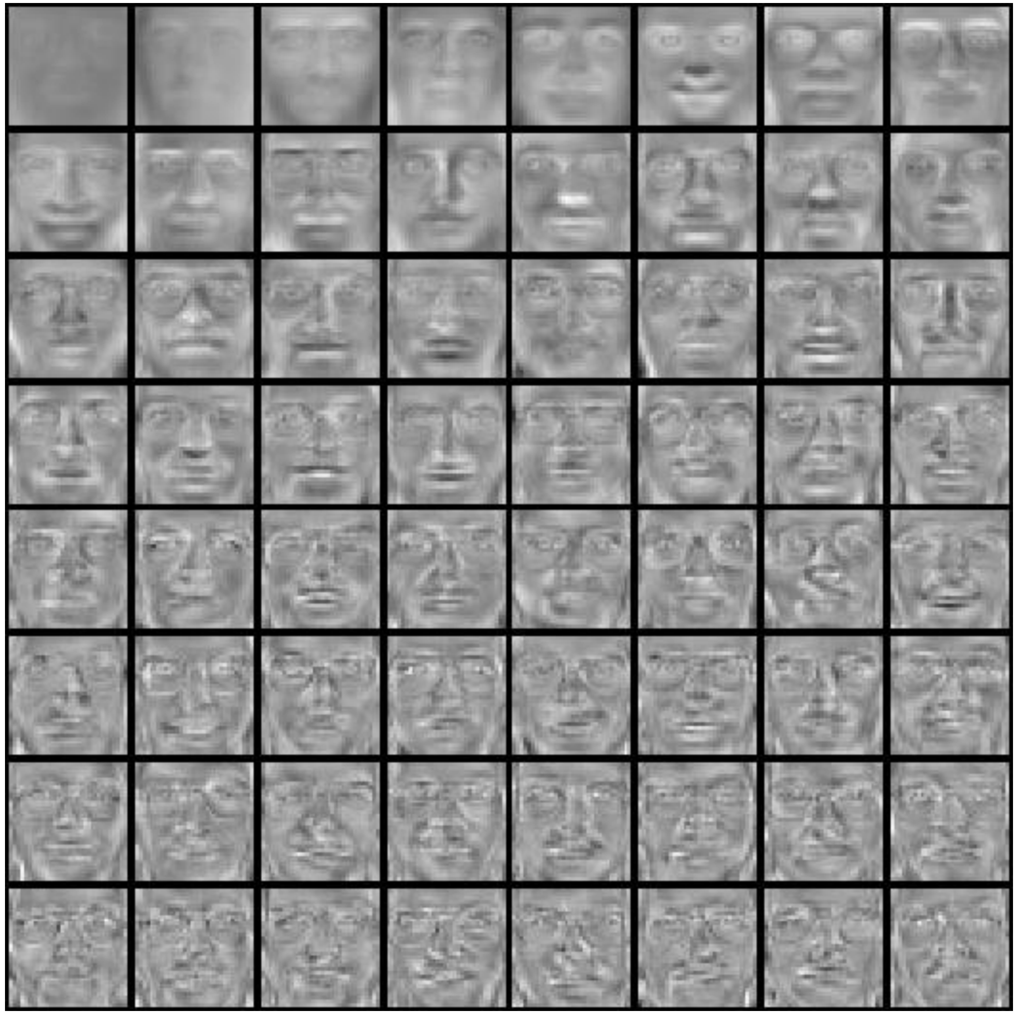
Eigenfaces example

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

Eigenfaces example

Mean: μ

Top eigenvectors: $u_1,\ldots u_k$



Representation and reconstruction

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

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x}-\mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x}-\mu)]$$
$$= \quad w_1, \ldots, w_k$$

Representation and reconstruction

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

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x} - \mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x} - \mu)]$$
$$= \quad w_1, \ldots, w_k$$

- Reconstruction:

 =  + 

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

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 novel image $\mathbf{x}$

- Project onto subspace:
  $(w_1, \ldots, w_k) = (\mathbf{u}_1^T(\mathbf{x} - \boldsymbol{\mu}), \ldots, \mathbf{u}_k^T(\mathbf{x} - \boldsymbol{\mu}))$
- Optional: check reconstruction error $\mathbf{x} - \hat{\mathbf{x}}$ to determine whether image is really a face
- Classify as closest training face in k-dimensional subspace

M. Turk and A. Pentland, [Face Recognition using Eigenfaces](), CVPR 1991

PCA

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

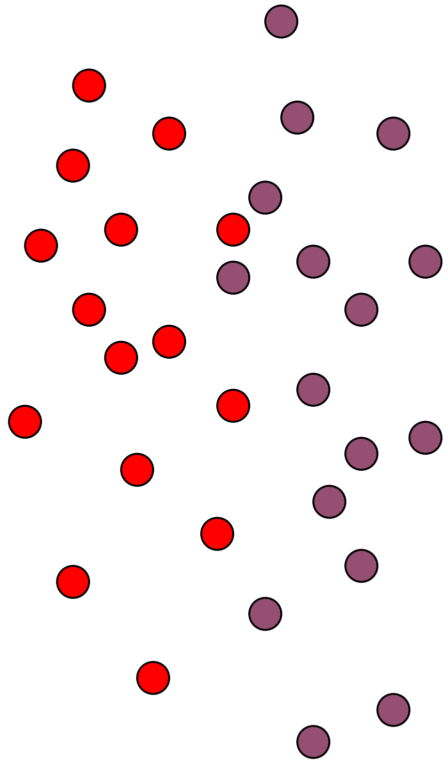
Limitations

Global appearance method: not robust to misalignment, background variation

Limitations

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

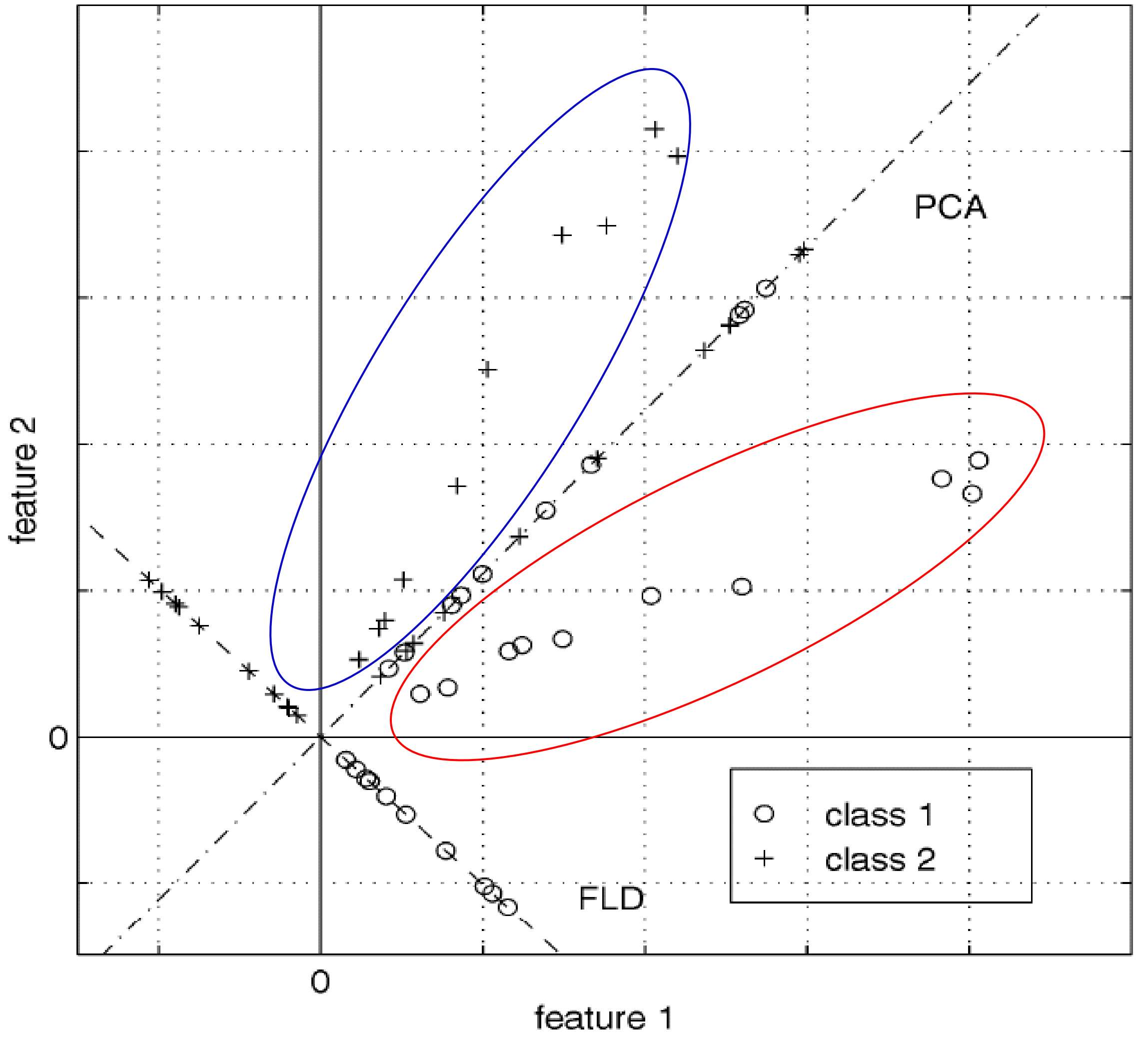


A more discriminative subspace: **Fisher's linear discriminant** (FLD)

- Fisher Linear Discriminant→ “Fisher Faces (Fisherfaces)”
- Fisher Linear Discriminant Analysis: F-LDA, LDA, FLD
- PCA preserves maximum variance
- FLD preserves discrimination
  - Find projection that maximizes scatter between classes and minimizes scatter within classes

Reference: Eigenfaces vs. Fisherfaces, Belheumer et al., PAMI 1997

Comparing with PCA



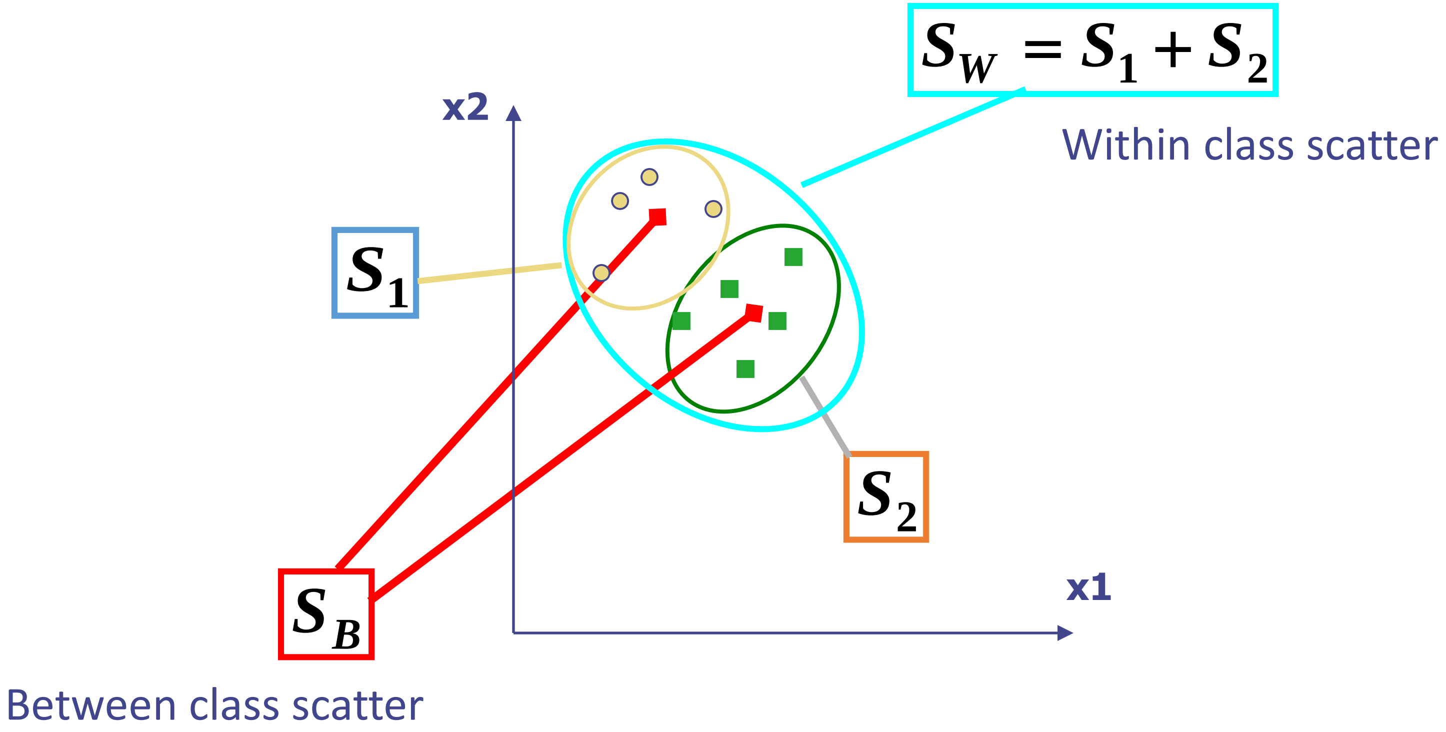
Variables

- N Sample images: $\{x_1, \cdots, x_N\}$
- c classes: $\{\chi_1, \cdots, \chi_c\}$
- Average of each class: $\mu_i = \frac{1}{N_i} \sum_{x_k \in \chi_i} x_k$
- Average of all data: $\mu = \frac{1}{N} \sum_{k=1}^{N} x_k$

Scatter Matrices

- 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: $$S_B = \sum_{i=1}^{c} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

Illustration



Mathematical Formulation

- After projection $\quad y_k = W^T x_k$
  - Between class scatter $\quad \tilde{S}_B = W^T S_B W$
  - Within class scatter $\quad \tilde{S}_W = W^T S_W W$
- Objective:

$$W_{opt} = \arg\max_{W} \frac{|\tilde{S}_B|}{|\tilde{S}_W|} = \arg\max_{W} \frac{|W^T S_B W|}{|W^T S_W W|}$$

- Solution: Generalized Eigenvectors

$$S_B w_i = \lambda_i S_W w_i \qquad i = 1, \ldots, m$$

- Rank of $W_{opt} = S_W^{-1} S_B$ is limited
  - Rank(S_B) <= |C|-1
  - Rank(S_W) <= N-C

Recognition with FLD

- Use PCA to reduce dimensions to N-C

$$W_{pca} = \mathrm{pca}(X)$$

- Compute within-class and between-class scatter matrices for PCA coefficients

$$S_i = \sum_{x_k \in \chi_i} (x_k - \mu_i)(x_k - \mu_i)^T \qquad S_W = \sum_{i=1}^{c} S_i \qquad S_B = \sum_{i=1}^{c} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

- Solve generalized eigenvector problem

$$W_{fld} = \arg\max_{\mathrm{W}} \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} \qquad S_B w_i = \lambda_i S_W w_i \qquad i = 1, \dots, m$$

- Project to FLD subspace (c-1 dimensions)

$$W^T{}_{opt} = W^T{}_{fld} W^T{}_{pca} \qquad \hat{x} = W_{opt}{}^T x$$

- Classify by nearest neighbor

Note: x in step 2 refers to PCA coef; x in step 4 refers to original data

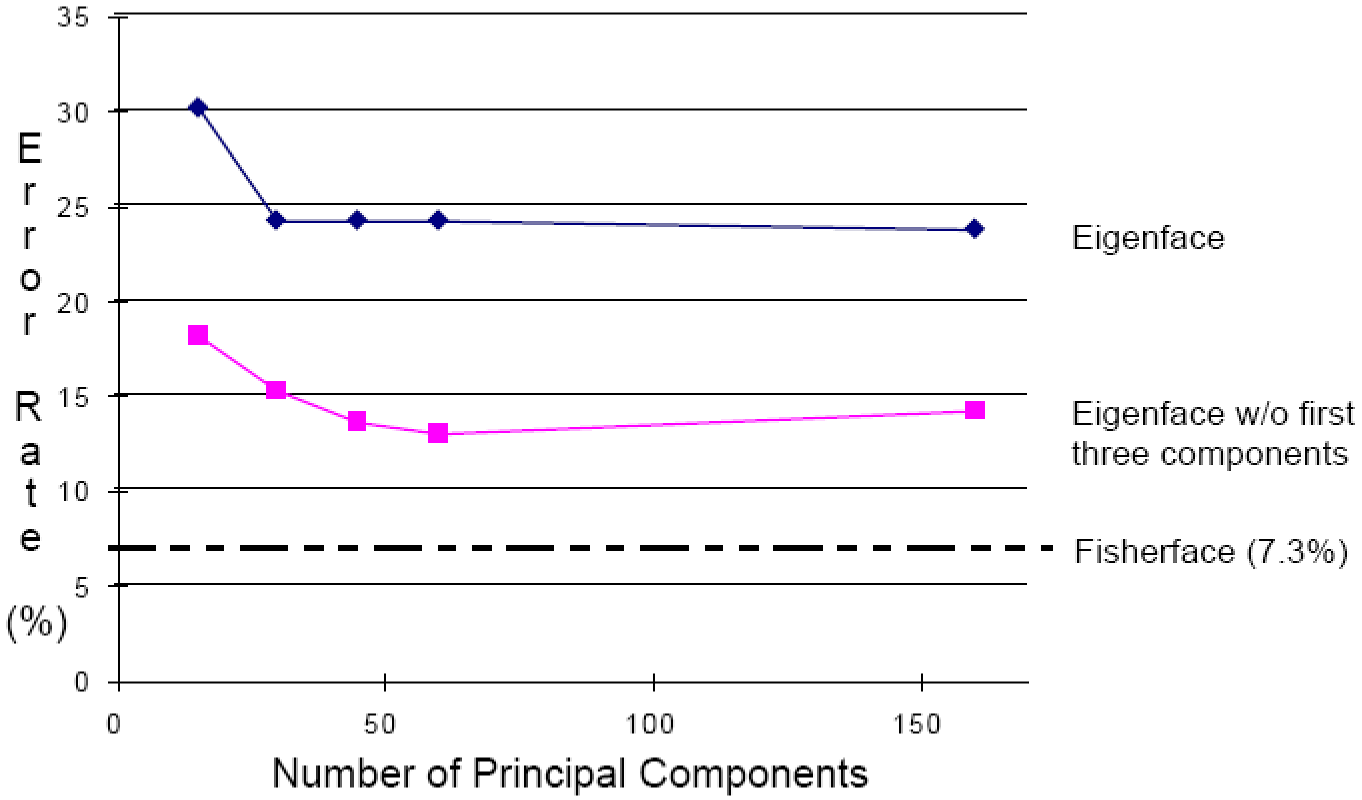
Results: Eigenface vs. Fisherface

- Input: 160 images of 16 people
- Train: 159 images
- Test: 1 image

- Variation in Facial Expression, Eyewear, and Lighting

Reference: [Eigenfaces vs. Fisherfaces, Belheumer et al., PAMI 1997]

Eigenfaces vs. Fisherfaces



Reference: [Eigenfaces vs. Fisherfaces, Belheumer et al., PAMI 1997]()

Sparse Representation

- Linear generative model:

$$\mathbf{y} = A\mathbf{x} + \boldsymbol{\varepsilon}$$

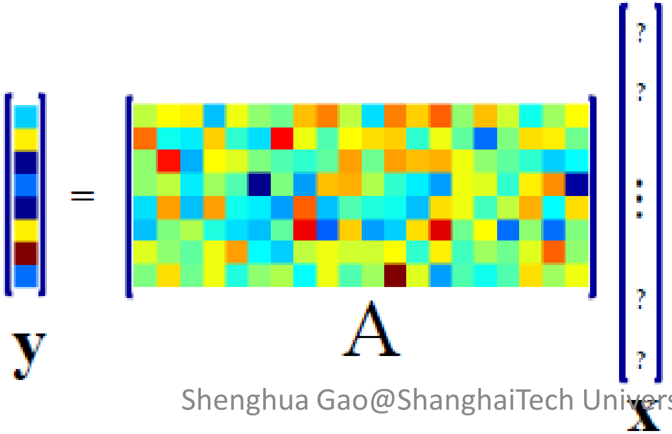
m-dimensional observations

matrix of n basis vectors or features

unknown sparse coefficients

noise

- **Objective**: Estimate the *sparse* $\mathbf{x}$ assuming $n >> m$

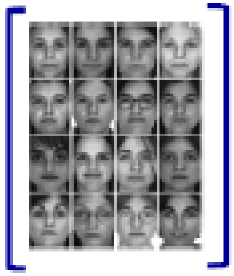


underdetermined system

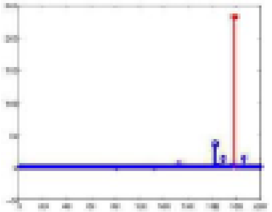
Face recognition

Generative model for faces, given a database of images from k subjects

=



$\times$



$+$

$\boldsymbol{y} \in \mathbb{R}^m$

Test image

$\boldsymbol{A} = [\boldsymbol{A}_1 \mid \boldsymbol{A}_2 \mid \cdots \mid \boldsymbol{A}_k]$

Combined training dictionary

$\boldsymbol{x} \in \mathbb{R}^n$

coefficients

$\boldsymbol{e} \in \mathbb{R}^m$

corruption, occlusion

[W., Yang, Ganesh, Sastry, Ma '09]

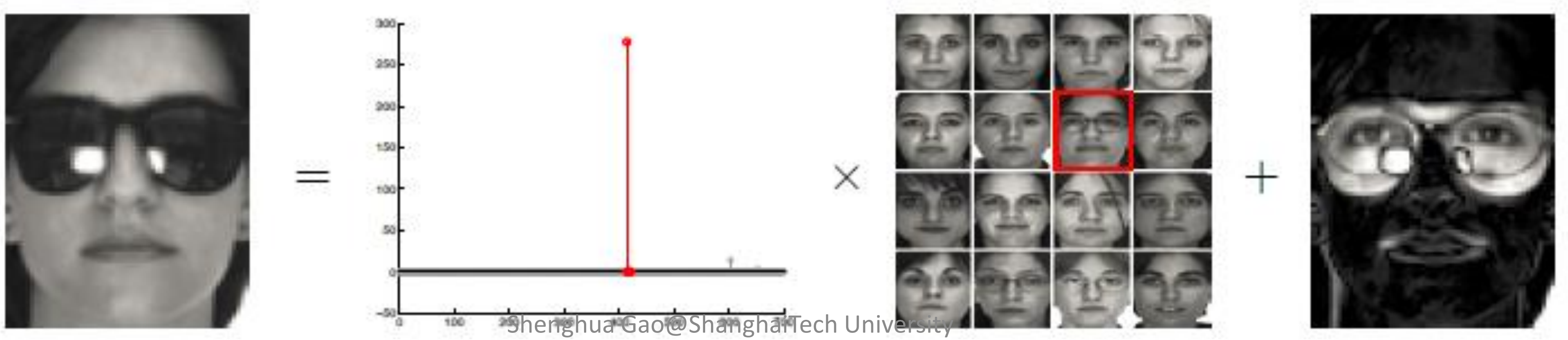
Sparse Representation for Face Recognition

- For face recognition, “If sufficient training samples are available from each class, it would be possible to represent a test sample as a linear combination of those training samples from the same class”.

train: $A_i = [a_{i,1}, \ldots, a_{i,n_i}] \in \mathbb{R}^{d \times n_i}$ $A = [A_1, \ldots, A_N] \in \mathbb{R}^{d \times \sum_{i=1}^{N} n_i}$ test: y

$$min \quad \|\alpha\|_1 \quad s.t. \quad \|y - A\alpha\|_2 \leq \epsilon$$

$\alpha_i = [\alpha_{i,1}, \ldots, \alpha_{i,n_i}]$ $\quad r_i(y) = \|y - A_i\alpha_i\|_2$ $\quad$ class label of $y := arg\ min_i\ \{r_1(y), \ldots, r_N(y)\}$



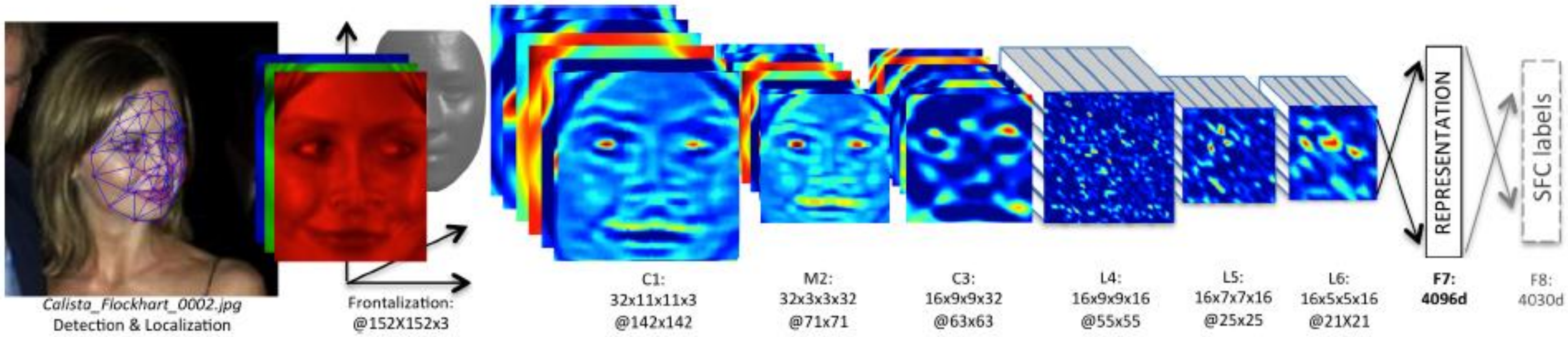
Deep Learning Based Face Recognition

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

Following slides adapted from Daphne Tsatsoulis

Face Alignment

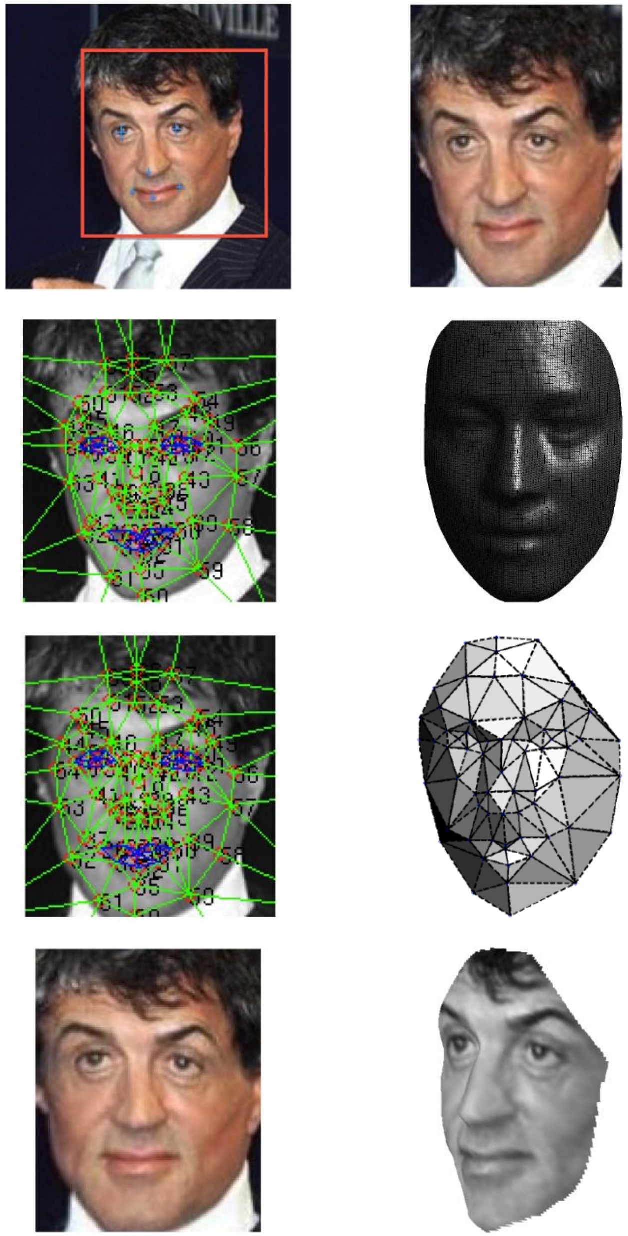
1. Detect a face and 6 fiducial markers using a support vector regressor (SVR)

2. Iteratively scale, rotate, and translate image until it aligns with a target face

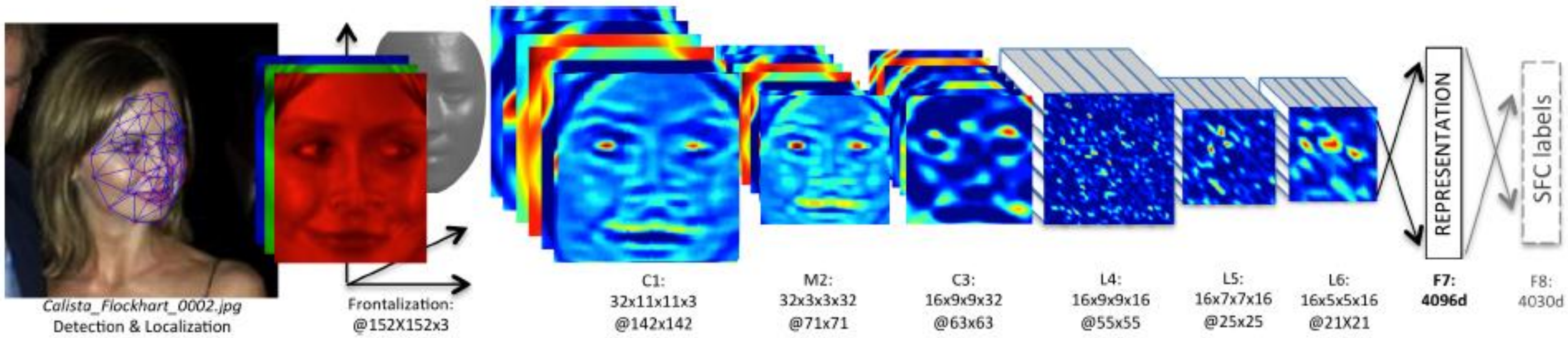
3. Localize 67 fiducial points in the 2D aligned crop

4. Create a generic 3D shape model by taking the average of 3D scans from the USF Human-ID database and manually annotate the 67 anchor points

5.Fit an affine 3D-to-2D camera and use it to direct the warping of the face



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

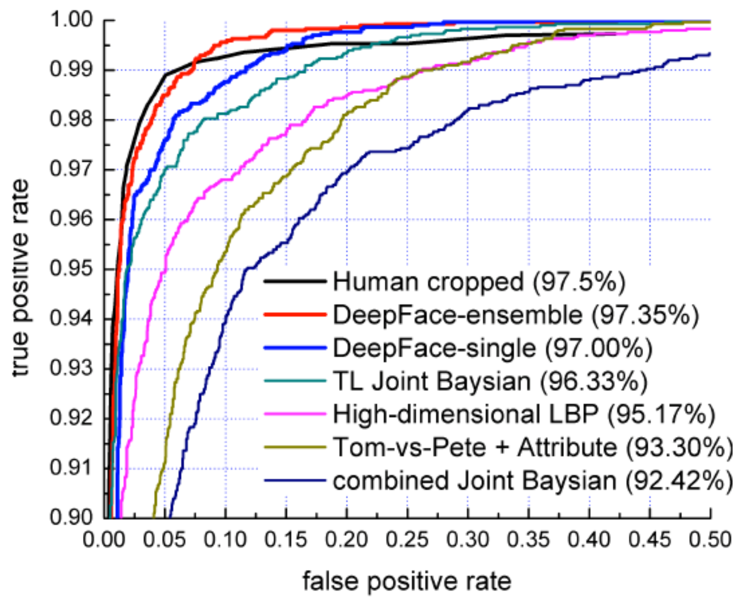
- Train to identify face among set of possible people

Verification is done by comparing features at last layer for two faces

Experiments: Datasets

- **LFW Dataset**
  - 13,323 webphotos of 5749 celebrities divided into 6000 face pairs in 10 splits
  - Performance based on mean recognition accuracy using
    - A) Restricted Protocol: Only same and not-same labels are used in training
    - B) Unrestricted Protocol: Given some identity information so many more training pairs can be added to the training set
    - C) Unsupervised Protocol: No training whatsoever is performed on LFW images
- **YTF Dataset**
  - 3425 YouTube videos of 1595 people, 5000 video pairs and 10 splits

Results: Labeled Faces in the Wild Dataset



Method	Accuracy $\pm$ SE	Protocol
Joint Bayesian [6]	0.9242 $\pm$0.0108	restricted
Tom-vs-Pete [4]	0.9330 $\pm$0.0128	restricted
High-dim LBP [7]	0.9517 $\pm$0.0113	restricted
TL Joint Bayesian [5]	0.9633 $\pm$0.0108	restricted
DeepFace-single	**0.9592** $\pm$0.0029	unsupervised
DeepFace-single	**0.9700** $\pm$0.0028	restricted
DeepFace-ensemble	**0.9715** $\pm$0.0027	restricted
DeepFace-ensemble	**0.9735** $\pm$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)

Face Verification

Shenghua Gao

Face Verification

- Face Identification: Who it is? 1 vs. N
- Face verification: are they the same person? 1 vs. 1

http://vis-www.cs.umass.edu/lfw/results.html#Human

Face Verification

- Two-steps:
  - Extract features: represent each face as one feature vector
  - Calculate the distance
    - Imposter Pair: distance is larger than a threshold
    - Genuine pair: distance is smaller than a threshold

- Procedure: Training samples contain Imposter and Authentic pairs
  - Training: learn a distance threshold, and/or a feature extractor, or/and a distance metric;
  - Testing: extract features, calculate distance, and make the decision

Siamese Network: feature learning

Learning a Similarity Metric Discriminatively, with Application to Face Verification

Sumit Chopra Raia Hadsell Yann LeCun

Courant Institute of Mathematical Sciences
New York University
New York, NY, USA
{sumit, raia, yann}@cs.nyu.edu

Siamese Network

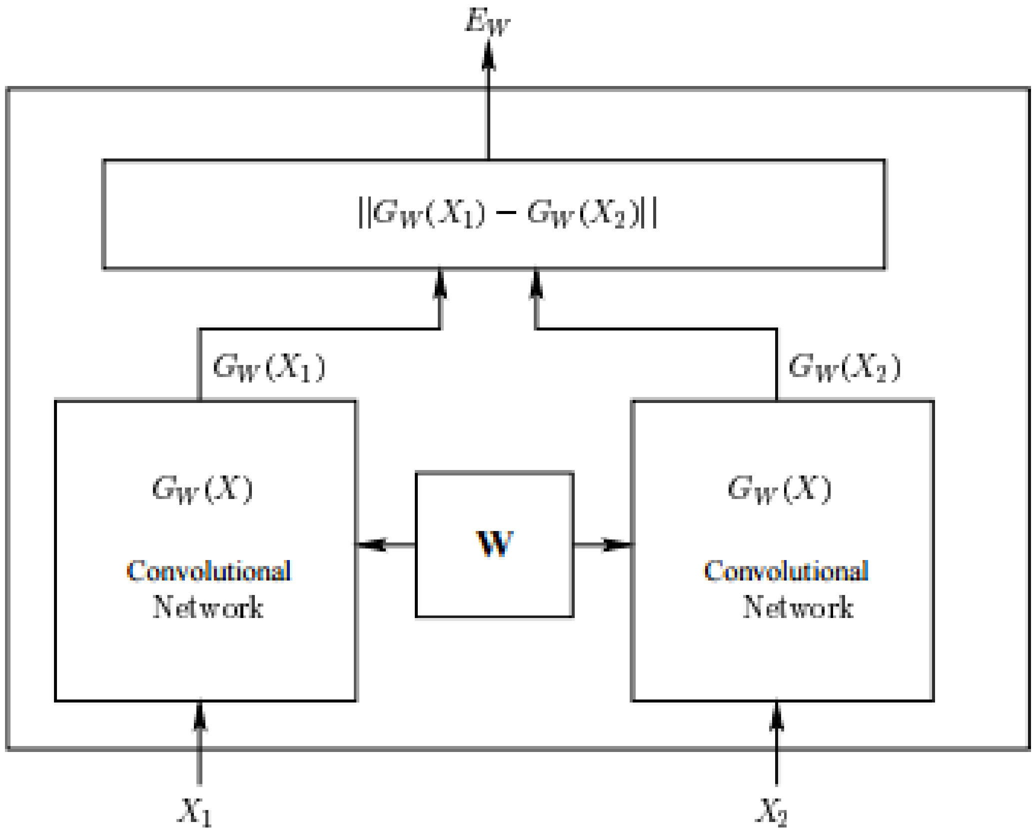
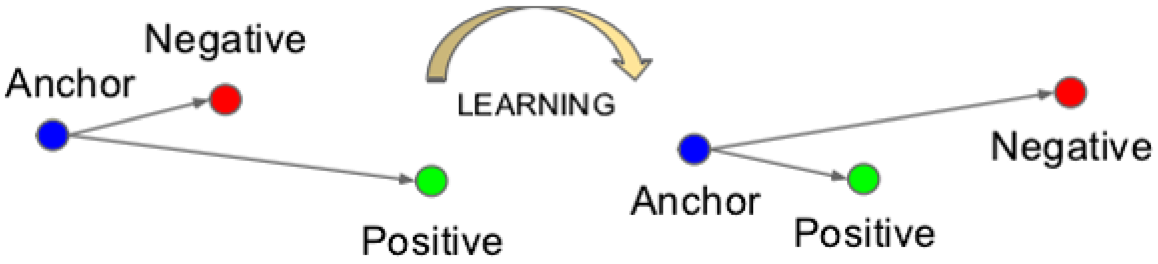


Figure 1. Siamese Architecture.

FaceNet: A Unified Embedding for Face Recognition and Clustering

Florian Schroff[1], Dmitry Kalenichenko[1], James Philbin[1] ({fschroff, dkalenichenko, jphilbin}@google.com)
[1]Google Inc.



$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \ \forall \left(x_i^a, x_i^p, x_i^n\right) \in \mathcal{T} \ , \quad (1)$$

The loss that is being minimized is then $L =$

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+ \ . \quad (2)$$

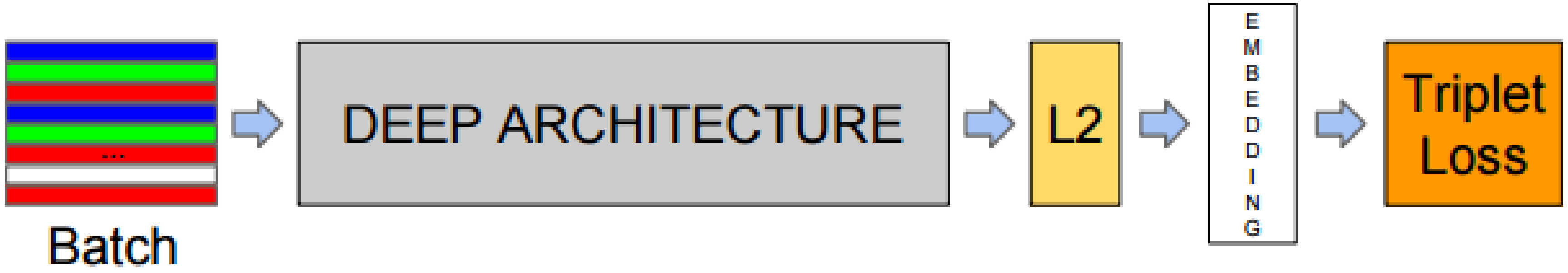


Figure 3: **Model structure.** Our network consists of a batch input layer and a deep CNN followed by L_2 normalization and the triplet loss.

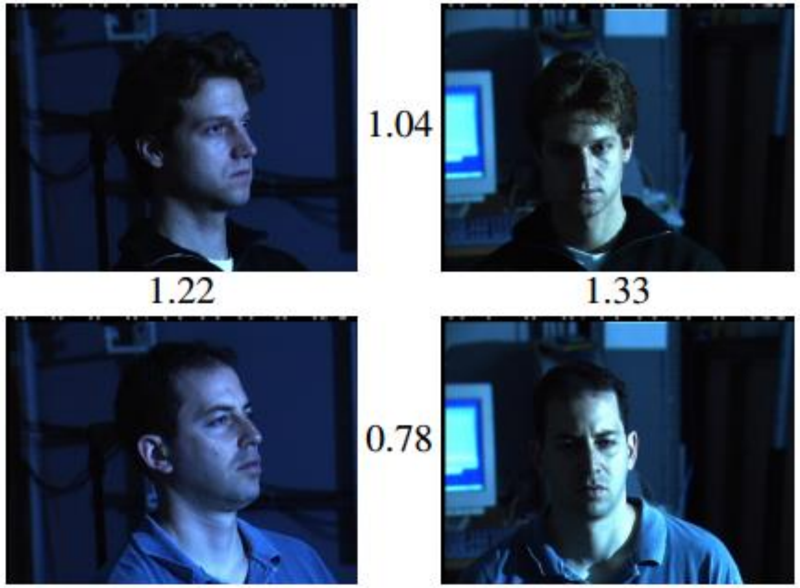


Figure 2: **Illumination and Pose invariance.** This figure shows the output distances of FaceNet between pairs of faces of the same and a different person in different pose and illumination combinations. A distance of 0.0 means the faces are identical, 4.0 corresponds to the opposite spectrum, two different identities. You can see that a threshold of 1.1 would classify every pair correctly.