Biometric Systems
Lesson 6: Face recognition – 2D

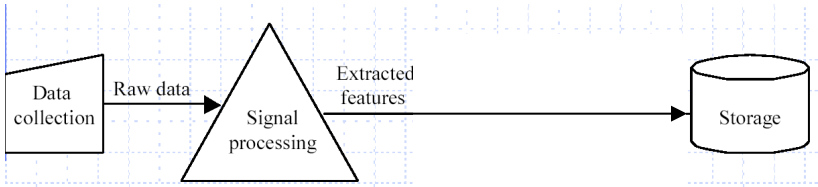
Maria De Marsico
demarsico@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Dipartimento di
Informatica

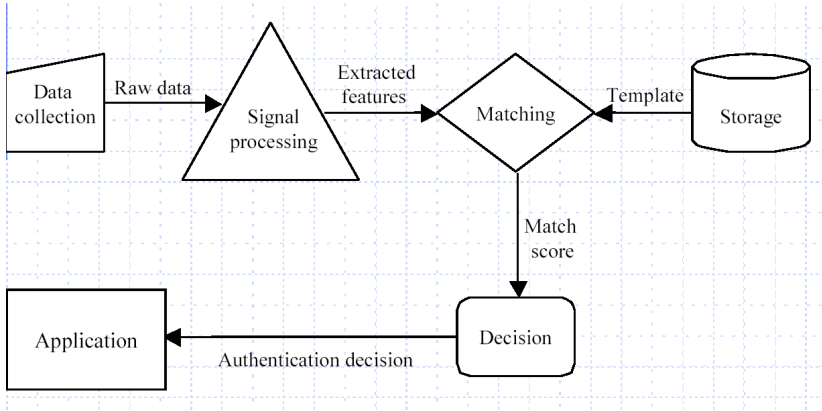
Structure of a face recognizer

- We associate a set of features to the identity of a subject
  - Data collection and feature extraction
- The extracted model (template) is stored in a daabase or on a mobile holder (smart card)
- This process can even be performed in bunches (batch enrollment)



Enrollment

- The same enrollment procedure extracts the template from the face of the subject to recognize (verification or identification)
  - Data collection and feature extraction
- The extracted model (template) is searched for in a database or on a mobile holder (smart card)
- A matching criterion determines if the subject must be accepted



Testing

Two main issues

- It is necessary to identify representative as well as discriminative features, allowing a «manageable» representation in a related feature space: it is very difficult to obtain a linear separations among classes, as well as convex classes, but we aim at obtaining the «less complex» representation possible → normalization (both geometrical and photometric = «canonical» forms)

- According to the selected features, it is necessary to build a robust classifier able to generalize training results → to correctly recognize even a never seen variation of an object.

Possible face representations

- An image I (possibly a face image) can be considered as a two-dimensional function I(x, y) defined in the Cartesian plane, which associates a value (e.g., a grey level between 0 and 255 or a RGB color with each channel ranging from 0 to 255) to each position (x, y) in the plane.

- An image I is usually stored as a matrix of size w × h in which each element (single number or tuple) represents the value of a pixel.

- Alternatively, an image can be represented as a one-dimensional vector of n = w × h pixels, for example by concatenating rows of the image one after the other.

- More in general, an image I can be represented as a point in a multi-dimensional space.

Possible face spaces

- Image space:
  - An image of size w×h is represented by a n-dimensional vector, with n = w×h
  - Considerable problems of dimensionality (***curse of dimensionality***)
  - Typically techniques for dimensionality reduction are adopted

- Feature space:
  - Gabor filters
  - Discrete Cosine Transform (DCT)
  - Local Binary Pattern (LBP) Operator
  - Fractal encoding (e.g., Partitioned Iterated Function Systems – PIFS)

Note

Curse of dimensionality

- The term *curse of dimensionality* was coined by Richard E. Bellman when considering problems in dynamic optimization.
- When the dimensionality increases, the volume of the space increases so fast that the available data become sparse.
- Sparsity is problematic for any method that requires statistical significance: in order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality.
- Data classification often relies on detecting areas where objects form groups with similar properties: in high dimensional data all objects appear to be sparse and somehow dissimilar, and this prevents data organization from being efficient.

Note (continued)

- Some consequences:
  - Machine learning requires learning a possibly infinite distribution from a finite number of data samples; in a high-dimensional feature space with each feature having a number of possible values, an enormous amount of training data are required to ensure that there are several samples with each combination of values; with a fixed number of training samples, the predictive power reduces as the dimensionality increases (*Hughes effect*).
  - When a measure such as a Euclidean distance is defined using many coordinates, there is little difference in the distances between different pairs of samples → problems in nearest neighbor search and k-nearest neighbor classification
- Dimensionality reduction methods such as Principal Component Analysis (PCA) aim at reducing high-dimensional data sets to lower-dimensional data without significant information loss, by identifying the most informative dimensions → relevant subspaces

A statistical method in detail: PCA

- **Principal component analysis** (**PCA**) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (some redundancy is involved) into a set of values of linearly uncorrelated variables (no redundancy) called **principal components**.
- The number of principal components is less than or equal to the number of original variables → the *n*-dimensional feature space is projected onto a *k*-dimensional subspace.
- The transformation 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 each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to (i.e., uncorrelated with) the preceding components.
- For an image space, the principal components are orthogonal when they are the **eigenvectors** of the **covariance** matrix, which is symmetric.
- In signal processing it is also often referred to as the discrete Karhunen–Loève transform (KLT).

A statistical method in detail: PCA

- The optimal *k* is computed starting from a set of *m n*-dimensional samples making up the training set ***TS*** = $\{\mathbf{x}_i \in \mathcal{R}^n | i=1,\ldots m\}$

- Given ***TS***, we compute the mean vector

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

- We can now compute the covariance matrix for the vector set ***TS:***

$$\mathrm{C} = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{x})(x_i - \bar{x})^T$$

Note on matrix multiplication

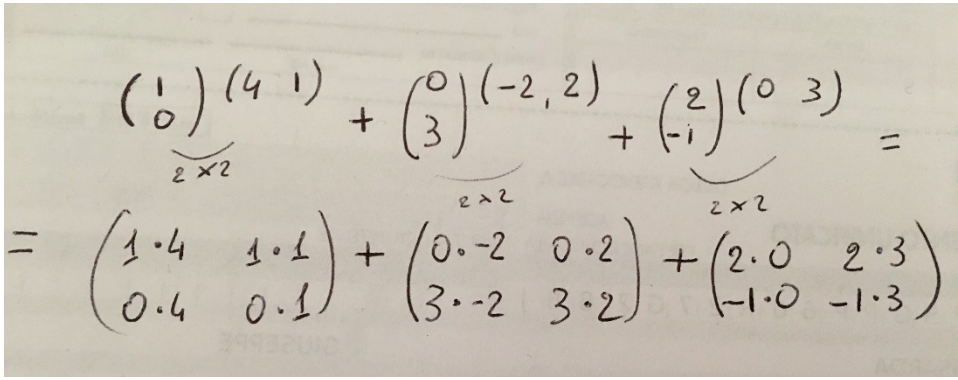
$$C = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{x})(x_i - \bar{x})^T$$

$$C = \frac{1}{m}(x \quad - \bar{x})(x \quad - \bar{x})^T$$

$$AB = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 3 & -1 \end{pmatrix}\begin{pmatrix} 4 & 1 \\ -2 & 2 \\ 0 & 3 \end{pmatrix} =$$

$$= \begin{pmatrix} 1\cdot 4 + 0\cdot(-2) + 2\cdot 0 & 1\cdot 1 + 0\cdot 2 + 2\cdot 3 \\ 0\cdot 4 + 3\cdot(-2) + (-1)\cdot 0 & 0\cdot 1 + 3\cdot 2 + (-1)\cdot 3 \end{pmatrix} =$$

$$= \begin{pmatrix} 4+0+0 & 1+0+6 \\ 0-6+0 & 0+6-3 \end{pmatrix} = \begin{pmatrix} 4 & 7 \\ -6 & 3 \end{pmatrix}$$



A statistical method in detail: PCA

- The size of the covariance matrix C is $n \times n$

- The new *k*-dimensional space is defined by the projection matrix whose columns are the *k* eigenvectors of C corresponding to the *k* highest eigenvalues of C.

- The eigenvector corresponding to the highest eigenvalue identifies the direction of maximum variation of data → highest information

- Eigenvalues represent variances along eigenvectors.

A statistical method in detail: PCA

- Sirovich and Kirby where the first to use Principal Component Analysis (PCA) for face recognition.
- They demonstrated that any particular face can be efficiently represented along the eigenpictures coordinate space, and that can be approximately reconstructed by using just a small collection of eigenpictures and the corresponding projections ('coefficients') along each eigenpicture.

- PCA → Image space for face representation

PCA and Eigenfaces

- Starting from Sirovich and Kirby's approach to representation, Turk and Pentland realized that projections along eigenpictures can be used as classification features to recognize faces.
- They developed a face recognition system that builds eigenfaces, corresponding to eigenvectors associated with the dominant eigenvalues of the covariance matrix built with linearized known faces (patterns).
- The system recognizes particular faces by comparing their projections along the eigenfaces to those of the face images of the known individuals.
- The eigenfaces define a feature space that drastically reduces the dimensionality of the original space, and face identification is carried out in this reduced space.
- PCA → Holistic classification

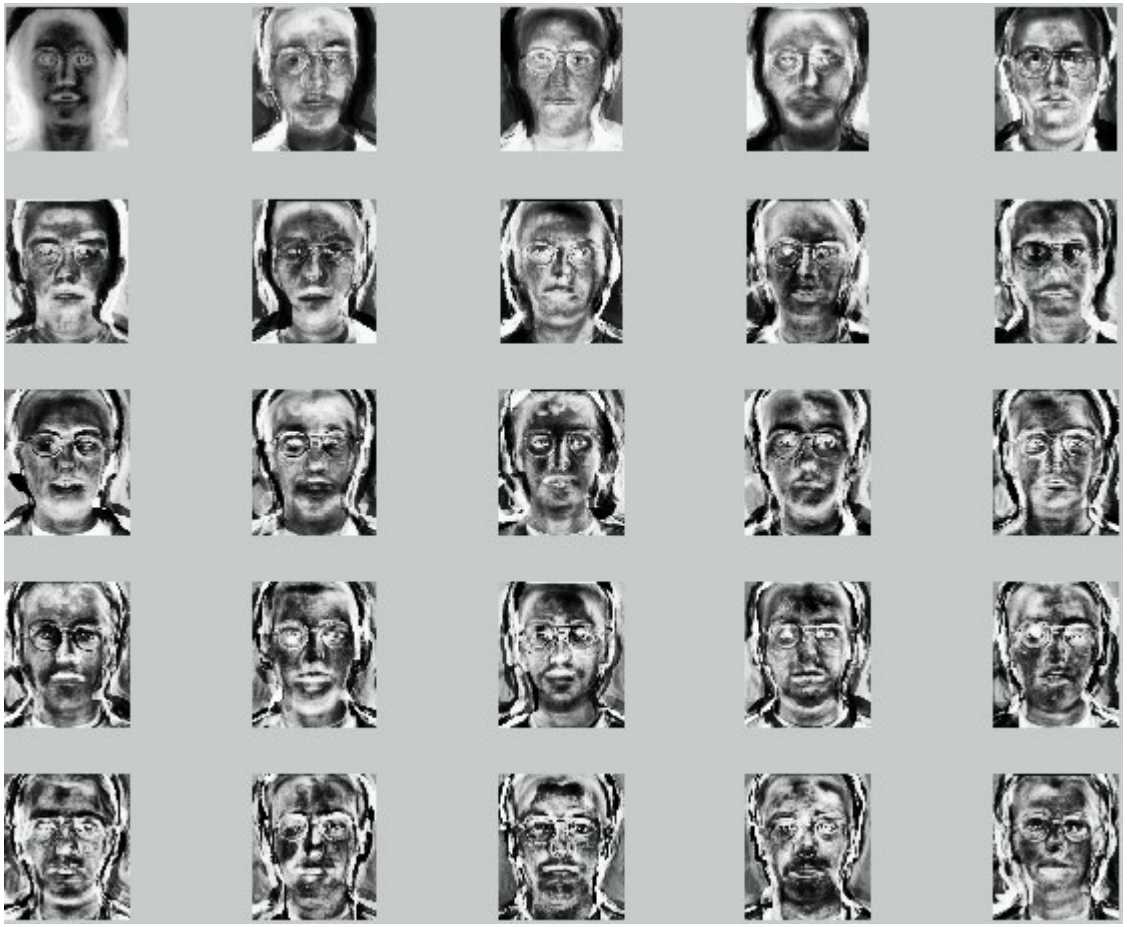
PCA and Eigenfaces

Example training set

PCA and Eigenfaces

Mean image from the training set

PCA and Eigenfaces



Eigenfaces from the training set

Projection

- Projection is performed by multiplying the trasposed projection matrix ${\varphi_k}^T$ by the original vector to which its mean was first substracted.

$$Proj\colon \Re^n \rightarrow \Re^k$$

$$Proj(x) = {\varphi_k}^T (x - \bar{x})$$

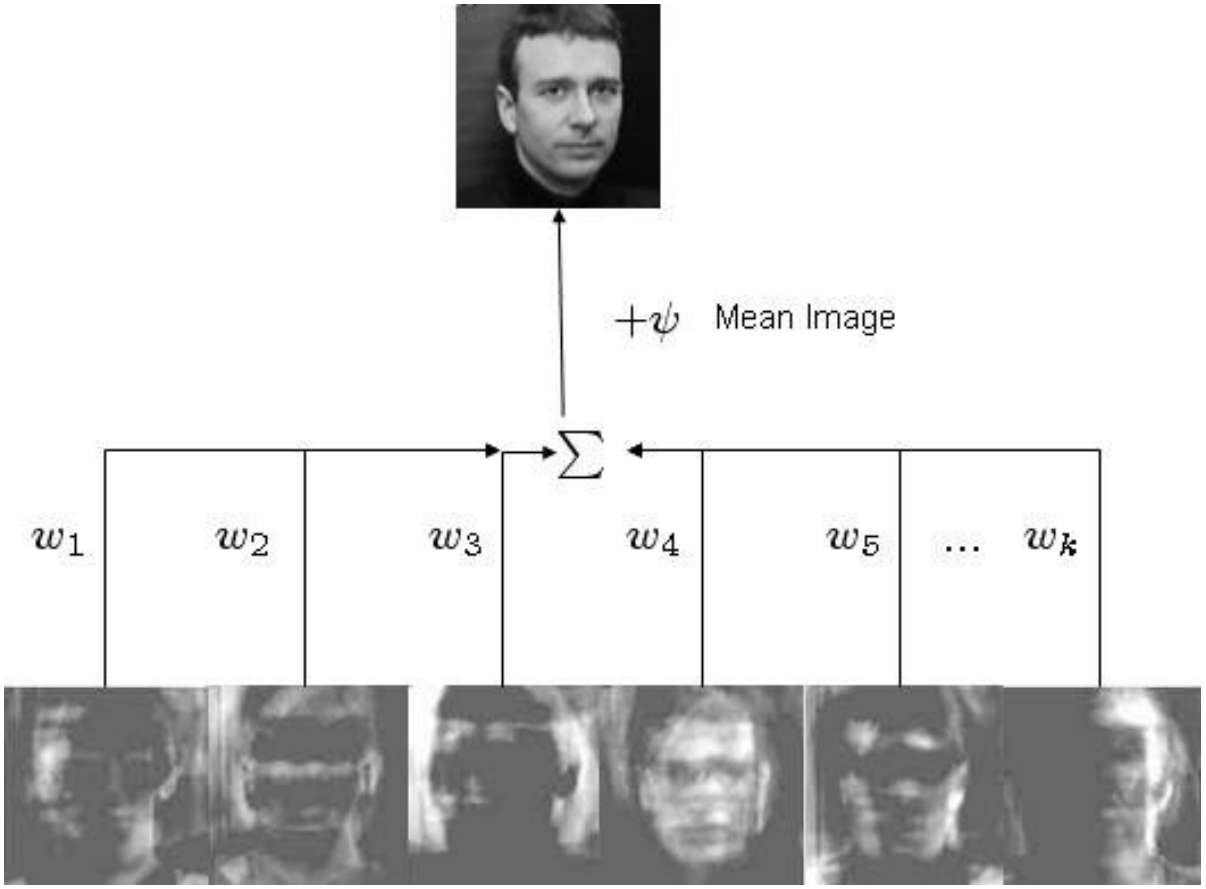
PCA and Eigenfaces

- Basic idea: perform the search in a subspace k-dimensional KL subspace which has been built after faces of the training set.

- Each gallery image is projected onto the KL subspace → the projection coefficients make up the feature vector.

- Each probe image is projected onto the subspace to be recognized, and classification is based on a simple nearest neighbor criterion : the face to be recognized is associated with the face of the gallery to which corresponds the minimum distance in the subspace (possibly subject to a threshold).

PCA and Eigenfaces



Example backprojection

PCA: problems and a solution

- PCA is good for representation, but lacks true discriminative power.

- Eigenfaces separation (scattering) that is maximized depends not only on inter-class separation (truly useful for classification) but also on the intra-class differences …

- When there are significant PIE variations, PCA retains them and therefore the similarity in the space of the faces is not necessarily determined the identity of individuals → the classes may be not separate correctly.

- To solve the above problems Belhumer et al. (1997) proposed **Fisherfaces,** an application of Fisher's linear discriminant (FLD), more often mentioned in the context of Linear Discriminant Analysis (LDA).

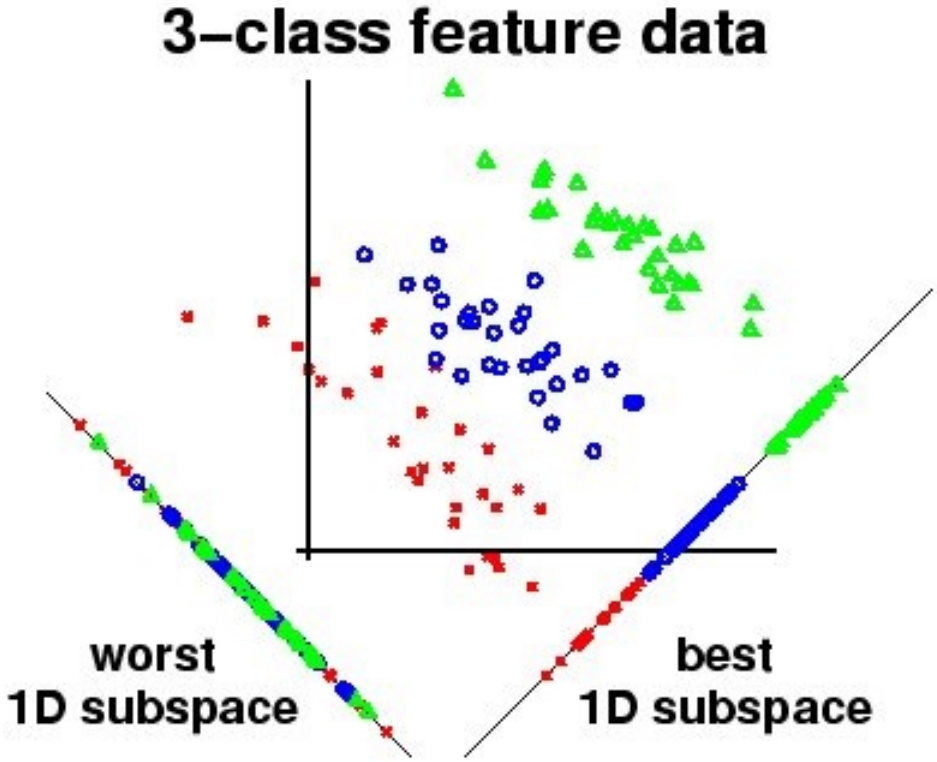
LDA: goal

- LDA is a technique of linear supervised (samples in the training set are labeled) dimensionality reduction whose objective is to maximize the separation between the classes.

- The space transformation is determined on the basis of an optimization criterion which has the aim of maximizing the separation between the classes in the reduced space

LDA: goal

- In the figure below, the reduction is from two to one dimension.
- It is not sufficient to reduce from two to one axis: axes are rotated so to maximize separation among classes



- Note that the projected values produce complete separation on the transformed axis, whereas there is overlap on both the original X and Y axes.

Figure by Schwardt and du Preez

A possible approach

- We start from a set of m n-dimensional samples making up the training set $\boldsymbol{TS} = \{\mathbf{x}_i \in \mathcal{R}^n | i=1,\ldots m\}$

- Differently from PCA, the training set is partitioned according to the known class labels of the s classes $\boldsymbol{PTS} = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_s\}$ where each class $\mathbf{P}_i$ has cardinality m_i

- We seek to obtain a scalar y by projecting the samples x onto a line such that

$$y = w^T x$$

A possible approach

- Without loss of generality, we can start from considering two classes $\mathbf{P}_1$, and $\mathbf{P}_2$ with respectively m_1 and m_2 vectors

- Let us consider the mean vectors of each class in the two spaces

$$\mu_i = \frac{1}{m_i}\sum_{j=1}^{m_i} x_j$$

and

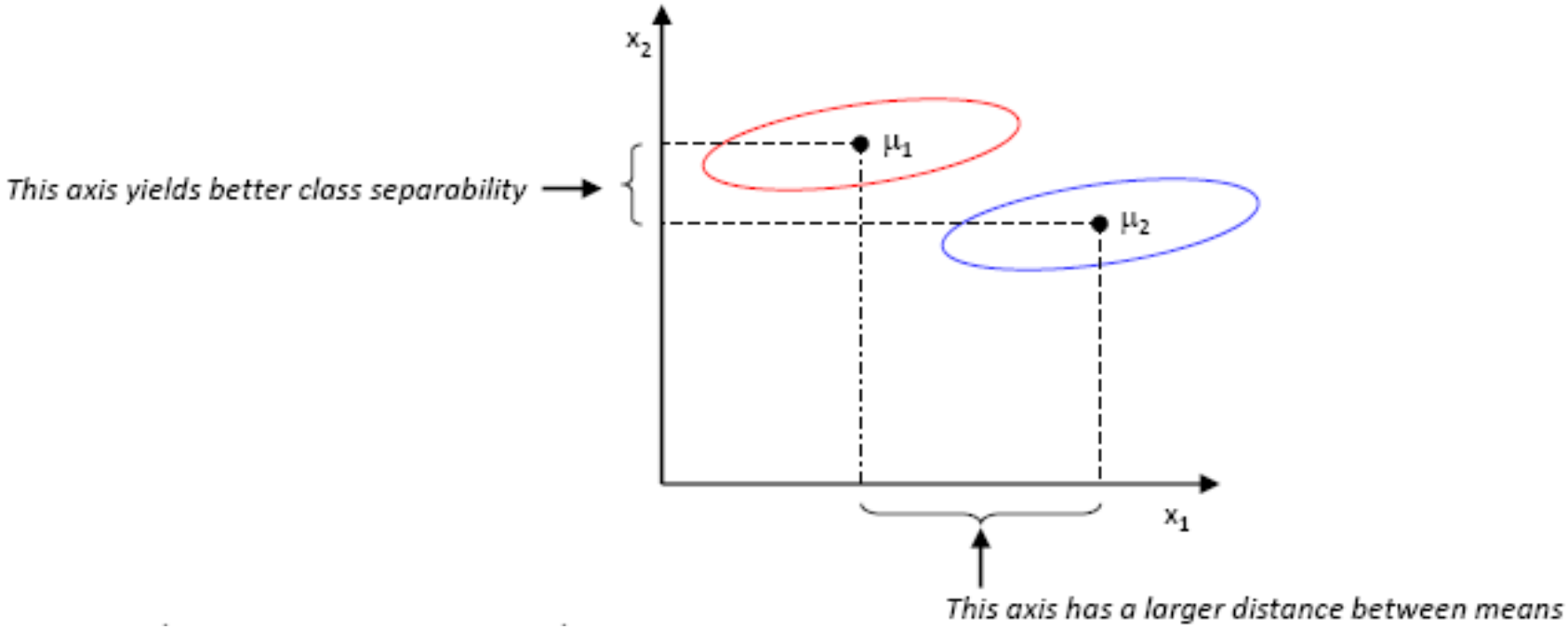
$$\tilde{\mu}_i = \frac{1}{m_i}\sum_{j=1}^{m_i} y_j = \frac{1}{m_i}\sum_{j=1}^{m_i} w^T x_j = \mathrm{w}^\mathsf{T}\mu_i$$

- We could choose the distance between the projected means as the objective function (to maximize)

$$J(w) = |\tilde{\mu}_1 - \tilde{\mu}_2| = |\mathrm{w}^\mathsf{T}(\mu_1 - \mu_2)|$$

A possible approach

- The distance between projected means is not a good measure since it does not account for the standard deviation within classes (scatter).



Scattering matrices

- Better results are obtained by maximizing the ratio of between-class variance to within-class variance starting from **scattering** matrices.
  - **Within-class scatter matrix S_w**: indicates how the vectors of each class are scattered with respect to its center
  - **Between-class scatter matrix S_b**: indicates how the centers o much the classes are scatt

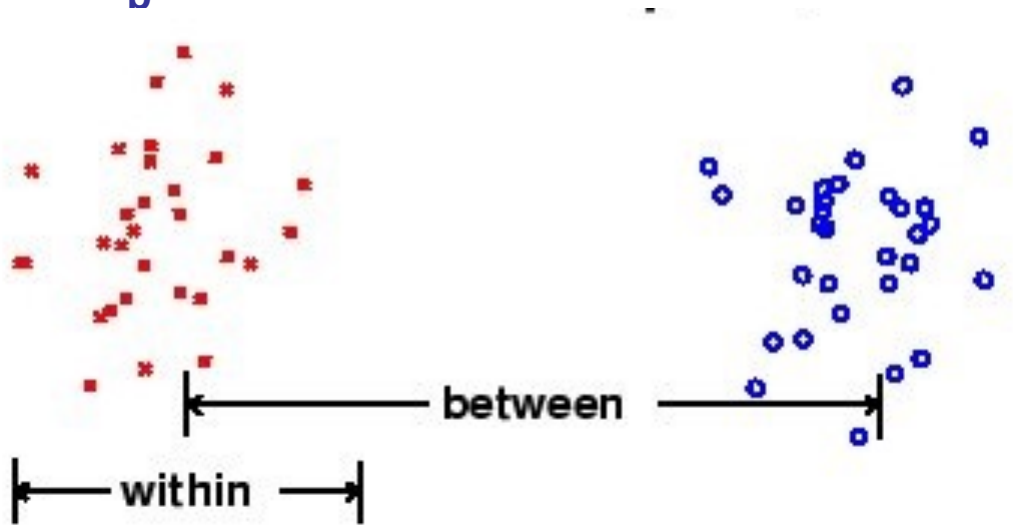


Figure by Ludwig Schwardt and Johan du Preez

Scatter matrices

- We start from a set of *m* *n*-dimensional samples making up the training set $\boldsymbol{TS} = \{\mathbf{x}_i \in \mathcal{R}^n | i=1,\ldots m\}$

- Differently from PCA, the training set is partitioned according to the known class labels of the s classes $\boldsymbol{PTS} = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_s\}$ where each class $\mathbf{P}_i$ has cardinality m_i

- For each class $\mathbf{P}_i$, we compute the mean vector (a «centroid» for each class similarly to PCA on the whole set) and the mean of means («centroid» of «centroids»)

$$\mu_i = \frac{1}{m_i}\sum_{j=1}^{m_i} x_j \qquad \mu_{TS} = \frac{1}{m}\sum_{i=1}^{s} m_i \mu_i$$

- We compute the covariance matrix for the vector set $\mathbf{P}_i$ (for each class similarly to PCA on the whole set):

$$\mathrm{C_i} = \frac{1}{m_i}\sum_{j=1}^{m_i}(x_j - \mu_i)(x_j - \mu_i)^T$$

Scatter matrices

- **Within-class scatter matrix S_w**:
  :

$$S_W = \sum_{i=1}^{s} m_i \mathrm{C_i}$$

- **Between-class scatter matrix S_b**:

$$\mathrm{S_B} = \sum_{i=1}^{s} m_i (\mu_i - \mu_{TS})(\mu_i - \mu_{TS})^T$$

(for two classes it is also defined as $\mathrm{S_B} = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$)

- The same definitions hold in the new space with $\widetilde{S_w}$ amd $\widetilde{S_b}$ with $y=w^Tx$ and scatter of projected class P_i defined as

$$\tilde{s}_i^2 = \sum_{y \in P_i} (y - \tilde{\mu}_i)^2$$

- where the quantity given by the sum of $\tilde{s}_i^2$ over all classes is called the within-class scatter of the projected examples

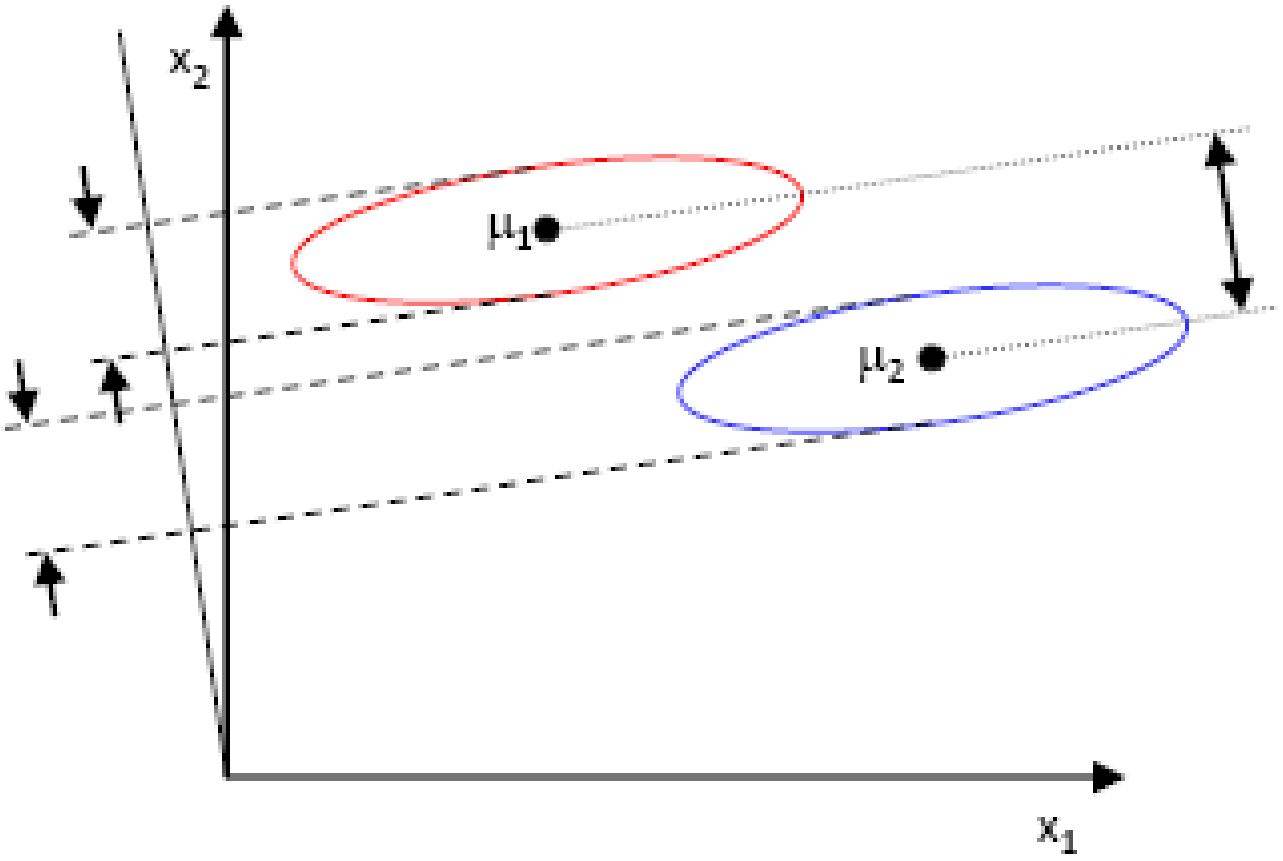
Fisher's solution

- Fisher suggested maximizing the difference between the means, normalized by a measure of the within-class scatter
- With two classes , the Fisher linear discriminant is defined as the linear function $w^T x$ that maximizes the criterion function

$$J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

- We are looking for a projection where examples from the same class are projected very close to each other and, at the same time, the projected means are as farther apart as possible

Fisher's solution



Fisher solution

- The difference between the projected means can be expressed in terms of the means in the original feature space

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (w^T\mu_1 - w^T\mu_2)^2 = w^T \underbrace{(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T}_{S_B} w = w^T S_B w$$

- Similarly, the scatter of the projection y can then be expressed as a function of the scatter matrix in feature space x

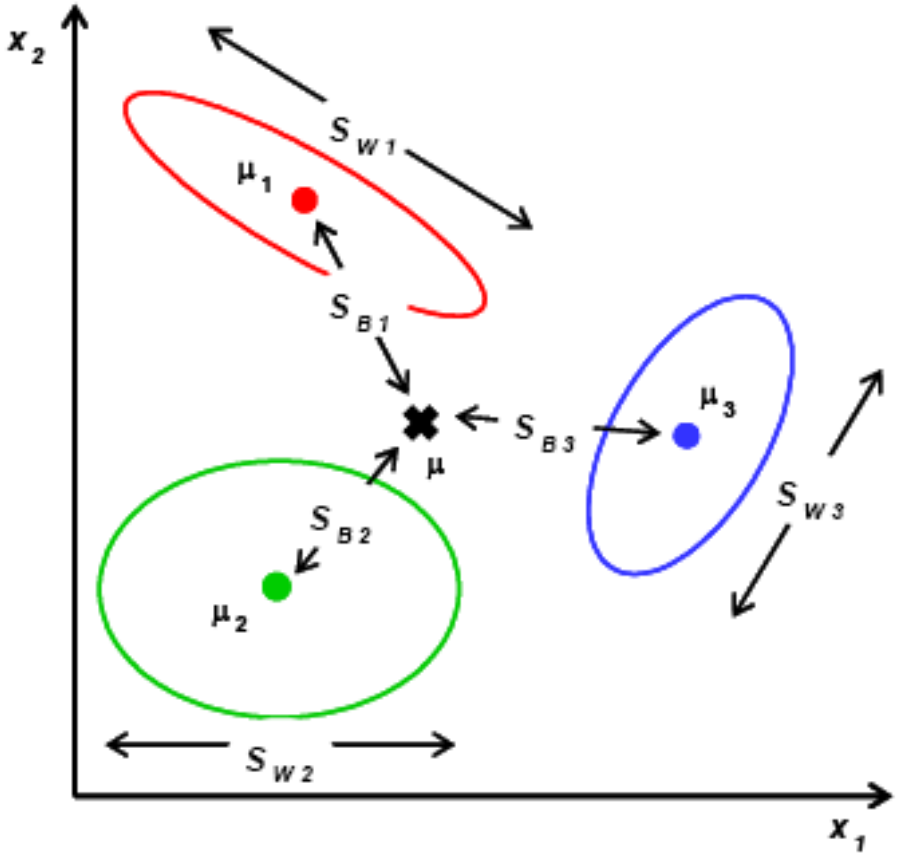
$$\tilde{s}_i^2 = \sum_{y\in\omega_i}(y - \tilde{\mu}_i)^2 = \sum_{x\in\omega_i}(w^T x - w^T\mu_i)^2 = \\ = \sum_{x\in\omega_i} w^T(x - \mu_i)(x - \mu_i)^T w = w^T S_i w$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^T S_W w$$

- Therefore the Fisher criterion requires to maximize

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

Fisher for more classes



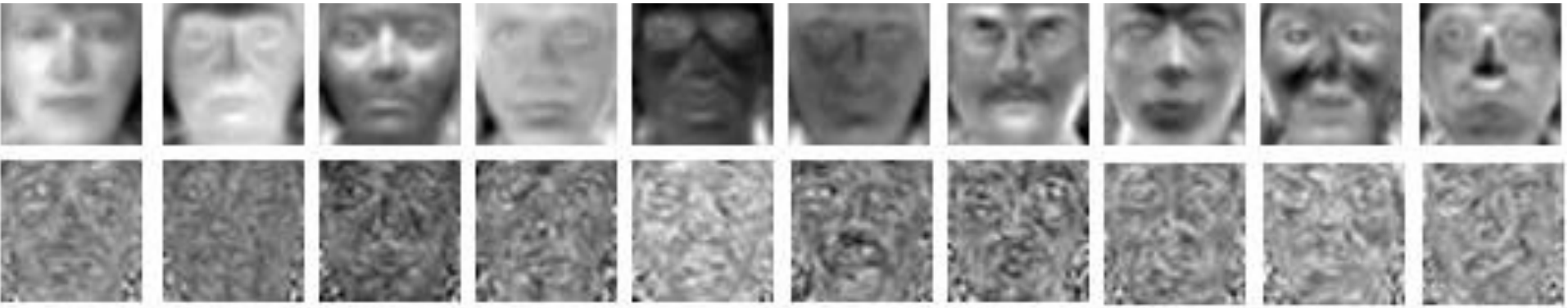
$$J(W) = \frac{|\tilde{S}_B|}{|\tilde{S}_W|} = \frac{|W^T S_B W|}{|W^T S_W W|}$$

We seek the projection matrix W^* that maximizes this ratio

It can be shown that the optimal projection matrix $W*$ is the one whose columns are the eigenvectors corresponding to the largest eigenvalues of the following generalized eigenvalue problem

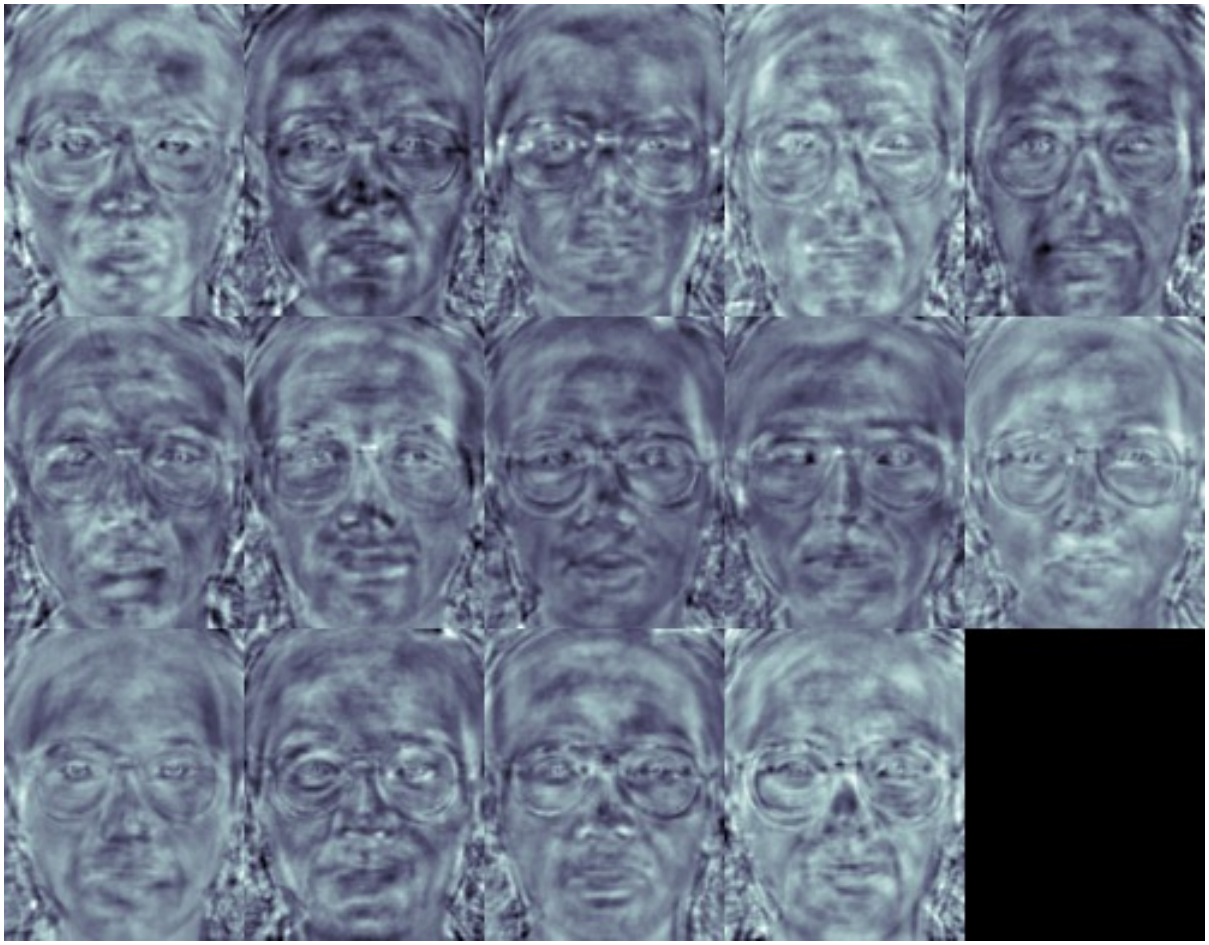
$$W^* = [w_1^* | w_2^* | \dots w_{C-1}^*] = \arg\max \frac{|W^T S_B W|}{|W^T S_W W|} \Rightarrow (S_B - \lambda_i S_W) w_i^* = 0$$

Fisherfaces



Eigenfaces (upper row) and Fisherfaces (lower row) for the same set of images from Yale database

Fisherfaces



From an OpenCV tutorial

Feature spaces

- The relevant features of the image of a face can be extracted by applying image filters or transforms, different kinds of operators, each suited to detect particular properties.

- When the dimensionality of the extracted feature vectors is high, they can undergo a process of dimensionality reduction using one of the previously described techniques.

- We can only mention some of the most popular operators … considering also their variations they are orders of magnitude beyond the capacity of a course!

Which features? An interesting experiment: Bubbles

a.

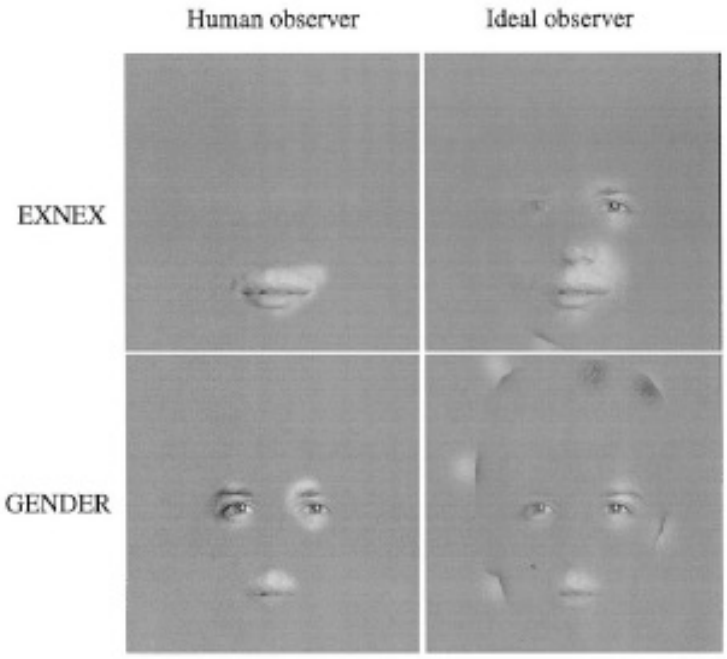
b.

A human observer can easily can make subtle judgments of gender, identity, age and expression, based on the same visual input, though requiring different information from it. Which information?

These two images were normalized as for pose, illumination and haistyle

Bubbles are devised by Gosselin & Schyns (2001)

Which features? An interesting experiment: Bubbles

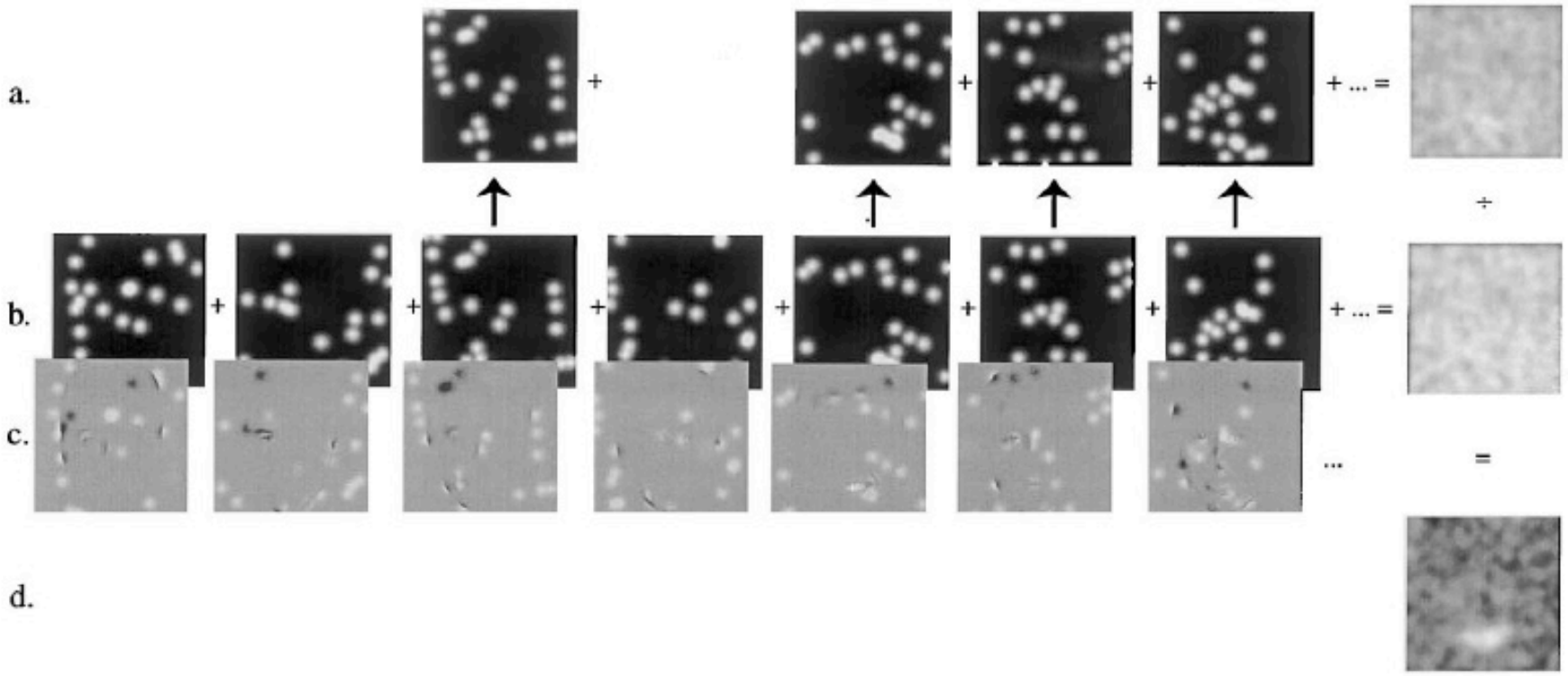


One of the experiments in Bubbles.

This figure illustrates diagnostic face information in an experiment for judging whether a face is expressive or not (EXNEX task), or its gender (GENDER task). The pictures are the outcome of Bubbles in the categorizations of the experiment on human (left column) and ideal observers (right column).

Ideal observer in this experiment will capture all the regions of the image that have highest local variance between the considered categories (male vs. female, and neutral vs. expressive). This ideal considers the stimuli as images (not faces composed of eyes, a nose and a mouth, as humans do) and it might not necessarily be sensitive to the regions that humans find most useful (the diagnostic regions), but rather to the information that is mostly available in the data set for the task at hand.

Which features? An interesting experiment: Bubbles



This figure illustrates Bubbles in experiment for the EXNEX task. (a) The bubbles leading to a correct categorization are added together to form the CorrectPlane (the rightmost greyscale picture); (b) all bubbles (those leading to a correct and incorrect categorizations) are added to form TotalPlane (the rightmost greyscale picture); (c), examples of experimental stimuli as revealed by the bubbles of (b). It is illustrative to judge whether each sparse stimulus is expressive or not. ProportionPlane (d) is the division of CorrectPlane with TotalPlane. Note the whiter mouth area (the greyscale has been renormalized to facilitate interpretation).

A state-of-the art approach: Wavelet Transforms

- “For a better understanding of the need for the WT, let's look at the FT more closely. FT (as well as WT) is a reversible transform ... However, … no frequency information is available in the time-domain signal, and no time information is available in the Fourier transformed signal. The natural question that comes to mind is that is it necessary to have both the time and the frequency information at the same time?

- Recall that the FT gives the frequency information of the signal, which means that it tells us how much of each frequency exists in the signal, but it does not tell us **when in time** these frequency components exist. This information is not required when the signal is so-called **stationary**.

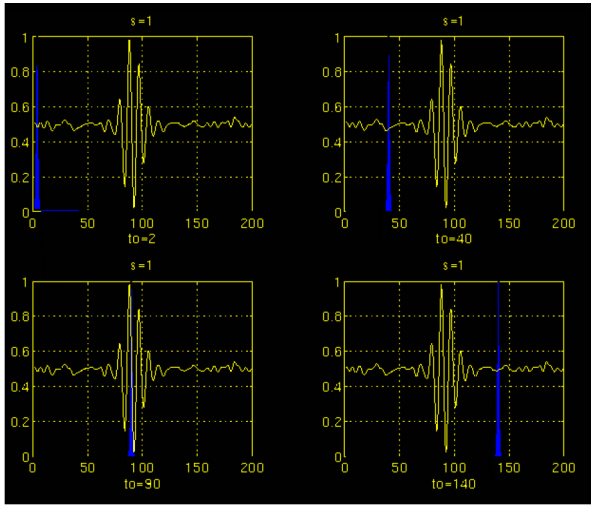
- … the frequency content of stationary signals do not change in time. In this case, one does not need to know **at what times frequency components exist** , since **all frequency components exist at all times.**

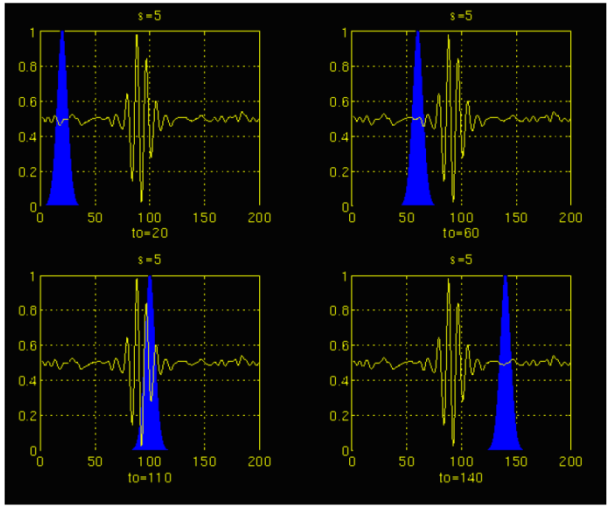
- … the spectrums of two entirely different signals [may] look very much alike… Recall that the FT gives the spectral content of the signal, but it gives no information regarding **where in time those spectral components appear**. Therefore, … FT can be used for non-stationary signals, if we are **only interested** in **what** spectral components exist in the signal, but **not interested where** these occur. ” (Wavelet Tutorial by Polikar)

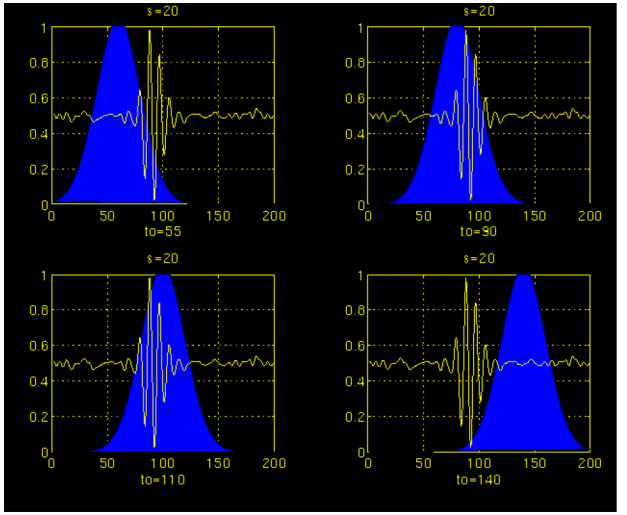
Wavelet transforms

- Wavelet transform is capable of providing the time and frequency information simultaneously, hence giving a time-frequency representation of the signal.

- Very trivial definition: a “mother wavelet” is shifted in time-domain, and the process is repeated with different scales (inversely proportional to frequencies).







The term **wavelet** means a **small wave** . The smallness refers to the condition that this (window) function is of finite length (it has a **compact** support, which means that it vanishes
outside of a finite interval). The **wave** refers to the condition that this function is oscillatory. The term **mother** implies that the functions with different region of support that are used in the transformation process are derived from one main function, or the mother wavelet. In other words, the mother wavelet is a **prototype** for generating the other window functions.

Wavelet transforms

$$CWT_x^{\psi}(\tau, s) = \Psi_x^{\psi}(\tau, s) = \int x(t).\psi_{\tau,s}^{*}(t)\, dt$$

$f^*(x)$ is the complex coniugate of complex function $f(x)$

$$\psi_{\tau,s} = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right)$$

Wavelet algorithms process data at different scales or resolutions. If we look at a signal with a large “window”, we would notice gross features. Similarly, if we look at a signal with a small “window”, we would notice small features. The result in wavelet analysis is to see “both the forest and the trees”.

Temporal analysis is performed with a contracted, high-frequency version of the prototype wavelet, while frequency analysis is performed with a dilated, low-frequency version of the same wavelet.

In two dimensions:

$$\psi_\theta(b_x, b_y, x, y, x_0, y_0) = \frac{1}{\sqrt{b_x b_y}}\psi_\theta\left(\frac{x-x_0}{b_x} + \frac{y-y_0}{b_y}\right)$$

where $\psi_\theta(x, y)$ is the 2-D mother wavelet, with b_x and b_y the scaling parameters, x_0 and y_0 the spatial shifting, and θ the orientation parameter.

Except for spatial shifting, Gabor filters are compatible with this expression.

An example of filters: Gabor filters

- From WIKIPEDIA:

"In image processing, a **Gabor filter**, named after Dennis Gabor, is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave.

Simple cells in the visual cortex of mammalian brains can be modeled by Gabor functions. Thus, image analysis with Gabor filters is thought to be similar to perception in the human visual system."

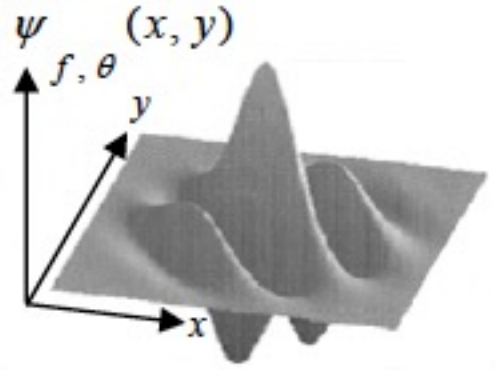
An example of filters: Gabor filters

- A Gabor filter works as a bandpass filter for the local spatial frequency distribution, achieving an optimal resolution in both spatial and frequency domains. $(i(2\pi f x_{\theta_n}))$
- The 2D Gabor filter $\psi_{f,\theta}(x,y)$ can be represented as a complex sinusoidal signal modulated by a Gaussian kernel function as

$$\psi_{f,\theta}(x,y) = \exp\left[-\frac{1}{2}\left\{\frac{x_{\theta_n}^2}{\sigma_x^2}+\frac{y_{\theta_n}^2}{\sigma_x^2}\right\}\right]\exp\left(i\left(2\pi f x_{\theta_n}\right)\right)$$

where, $\begin{bmatrix} x_{\theta_n} \\ y_{\theta_n} \end{bmatrix} = \begin{bmatrix} \sin\theta_n & \cos\theta_n \\ -\cos\theta_n & \sin\theta_n \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$

σ_x, σ_y are the standard deviations of the Gaussian envelope along the x- and y-dimensions, f is the central frequency of the sinusoidal plane wave, and θ_n the orientation. The rotation of the x-y plane by an angle θ_n will result in a Gabor filter at the orientation θ_n. The angle θ_n is defined by:



$$\theta_n = \frac{\pi}{p}(n-1),$$

for n=1,2,…,p and $p \in \mathrm{N}$, where p denotes the number of orientations.

An example of filters: Gabor filters

- The filter has a real and an imaginary component representing orthogonal directions. The two components may be formed into a complex number or used individually.

Complex

$$\psi_{f,\theta}(x,y) = \exp\left[-\frac{1}{2}\left\{\frac{x_{\theta_n}^2}{\sigma_x^2}+\frac{y_{\theta_n}^2}{\sigma_x^2}\right\}\right]\exp\left(i\left(2\pi f x_{\theta_n}\right)\right)$$

Real

$$\psi_{f,\theta}(x,y) = \exp\left[-\frac{1}{2}\left\{\frac{x_{\theta_n}^2}{\sigma_x^2}+\frac{y_{\theta_n}^2}{\sigma_x^2}\right\}\right]\cos(2\pi f x_{\theta_n})$$

Imaginary

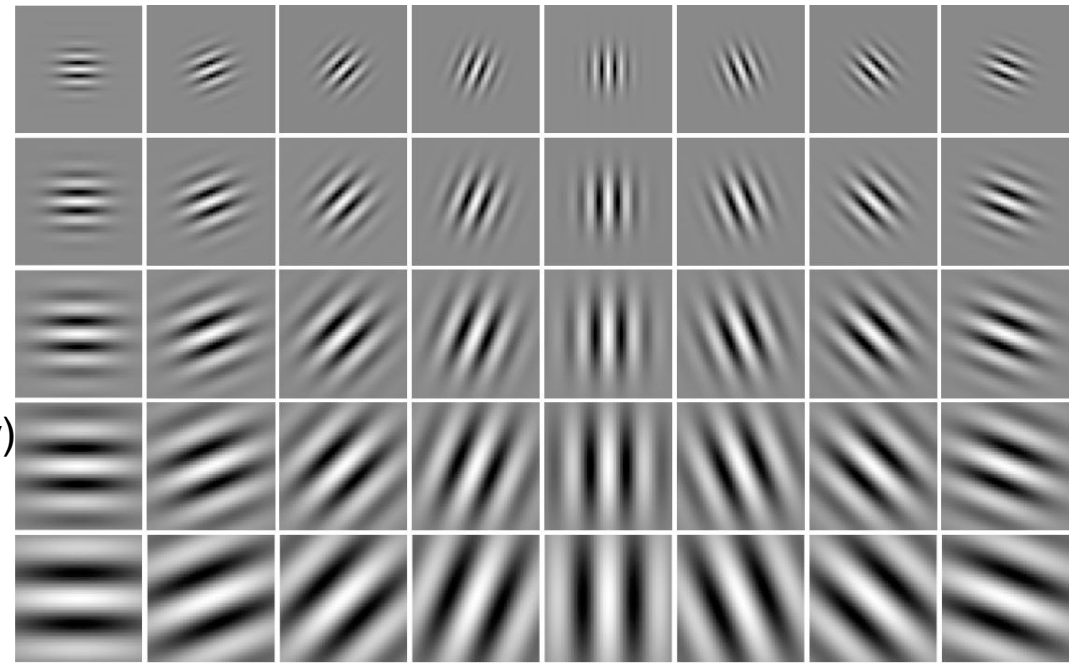
$$\psi_{f,\theta}(x,y) = \exp\left[-\frac{1}{2}\left\{\frac{x_{\theta_n}^2}{\sigma_x^2}+\frac{y_{\theta_n}^2}{\sigma_x^2}\right\}\right]\sin(2\pi f x_{\theta_n})$$

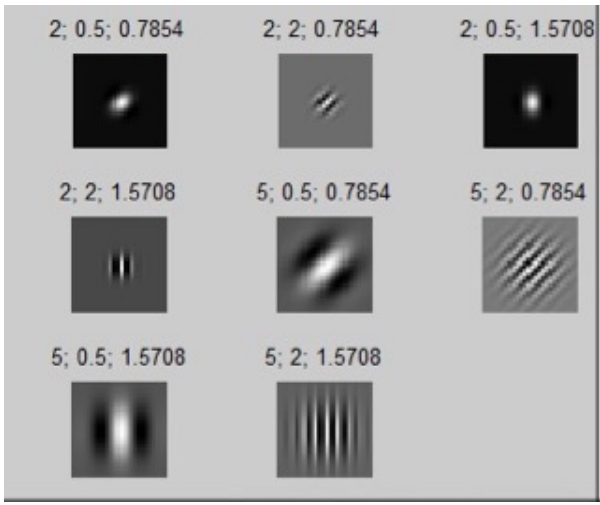
An example of filters: Gabor filters

- A feature vector is obtained by convolution of an image with a Gabor filter bank (different orientations and different scales)

orientation

scale (spatial frequency)





Each filter is specified giving the tuple (f, θ, s_x, s_y) with the last two possibly equal

Real part of filters

Figure from Bhuiyan & Liu (2007)

An example of filters: Gabor filters

$f(x,y)$ be the intensity at the coordinate (x,y) in a gray scale face image, its convolution with a Gabor filter $\psi_{f,\theta}(x,y)$ is defined as:

$$g_{f,\theta}(x,y) = f(x,y) \otimes \psi_{f,\theta}(x,y)$$

response to each Gabor kernel filter representation is a complex function with a real part $\Re\{g_{f,\theta}(x,y)\}$ and an imaginary part $\Im\{g_{f,\theta}(x,y)\}$. The magnitude response $\|g_{f,\theta}(x,y)\|$ is expressed as:

$$\|g_{f,\theta}(x,y)\| = \sqrt{\Re^2\{g_{f,\theta}(x,y)\} + \Im^2\{g_{f,\theta}(x,y)\}}$$

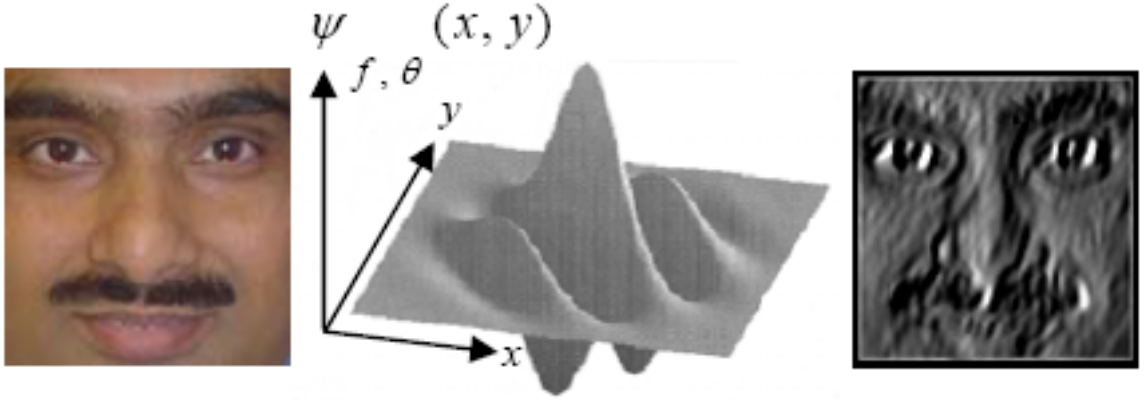


Fig. 3 Convolutic ult of a face image with a Gabor filter. (a) Face image, (b) Gabor filter (f=0.19,θ=3π/4,σ_x=σ_y=3), (c) Output of Gabor

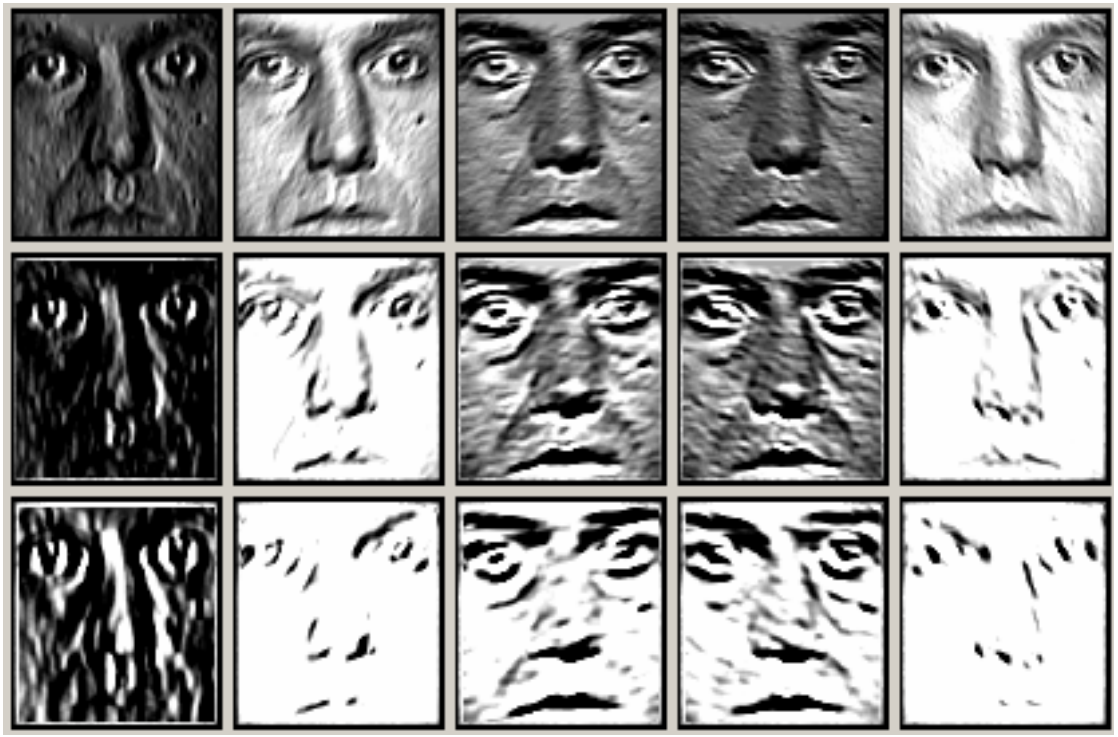
The magnitudes of responses

make up a possible feature vectr

$$\|g_{f,\theta}(x,y)\|$$

Figure and text from Bhuiyan & Liu (2007)

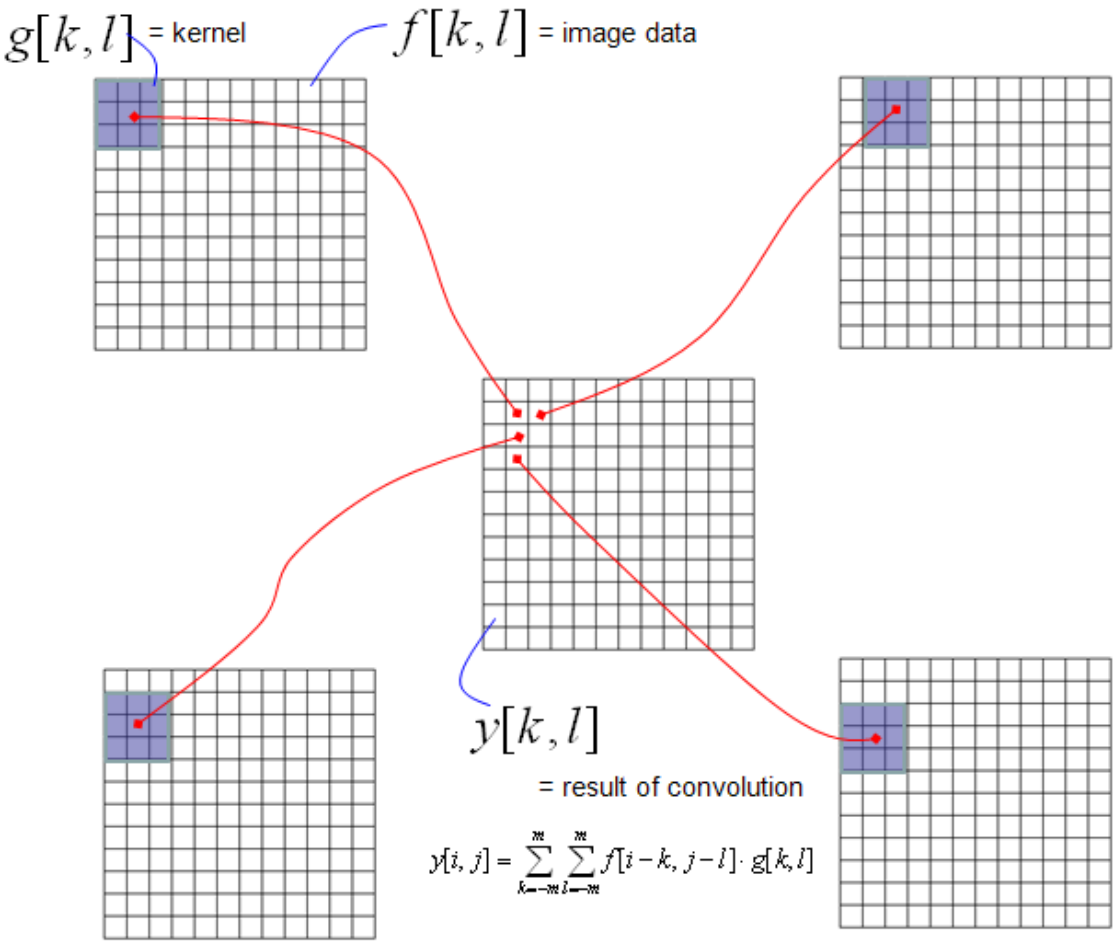
An example of filters: Gabor filters



2 D Gabor functions enhance edge contours. This corresponds to enhancing eye, mouth, and nose edges, and also moles, dimples, scars, etc.

If the convolution is performed on all the pixel of the image and we take the overall result, dimensionality is hight (size of the image). In alternative it can be performed on a regular grid of points, or on salinet face regions only

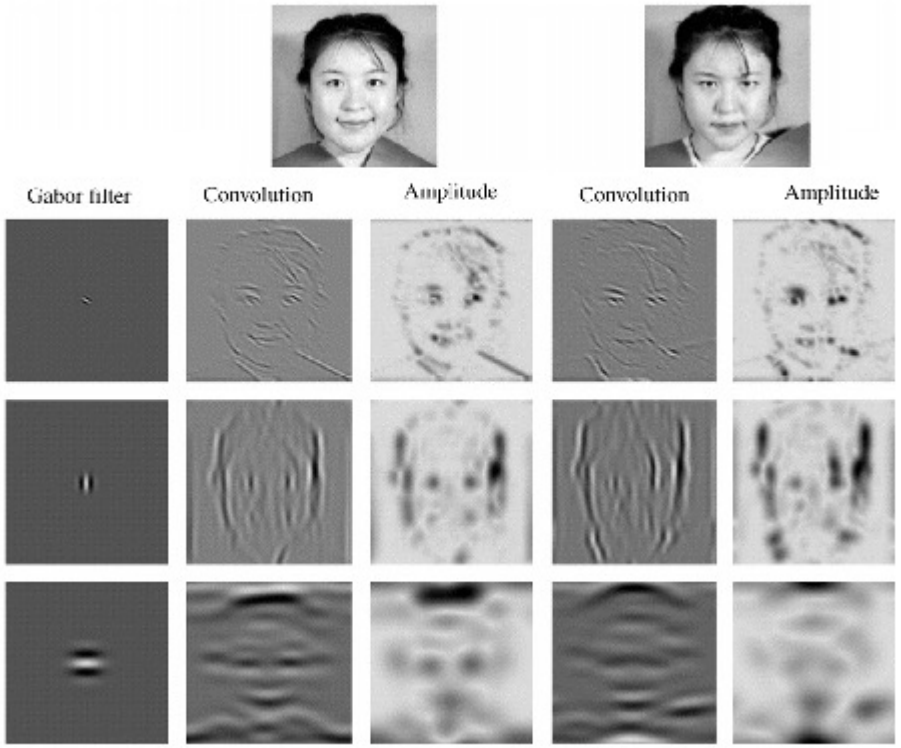
Note: Image Convolution



Where to perform convolution

- Overall image:

– the resulting feature vector has a huge size → dimensionality reduction



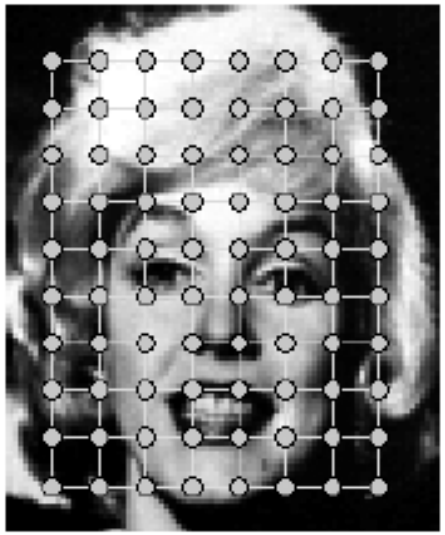
Where to perform convolution

- Fixed grid

– we can miss relevant points

- Grid defined «by hand» such that vertices are close to important facial features such as the eyes

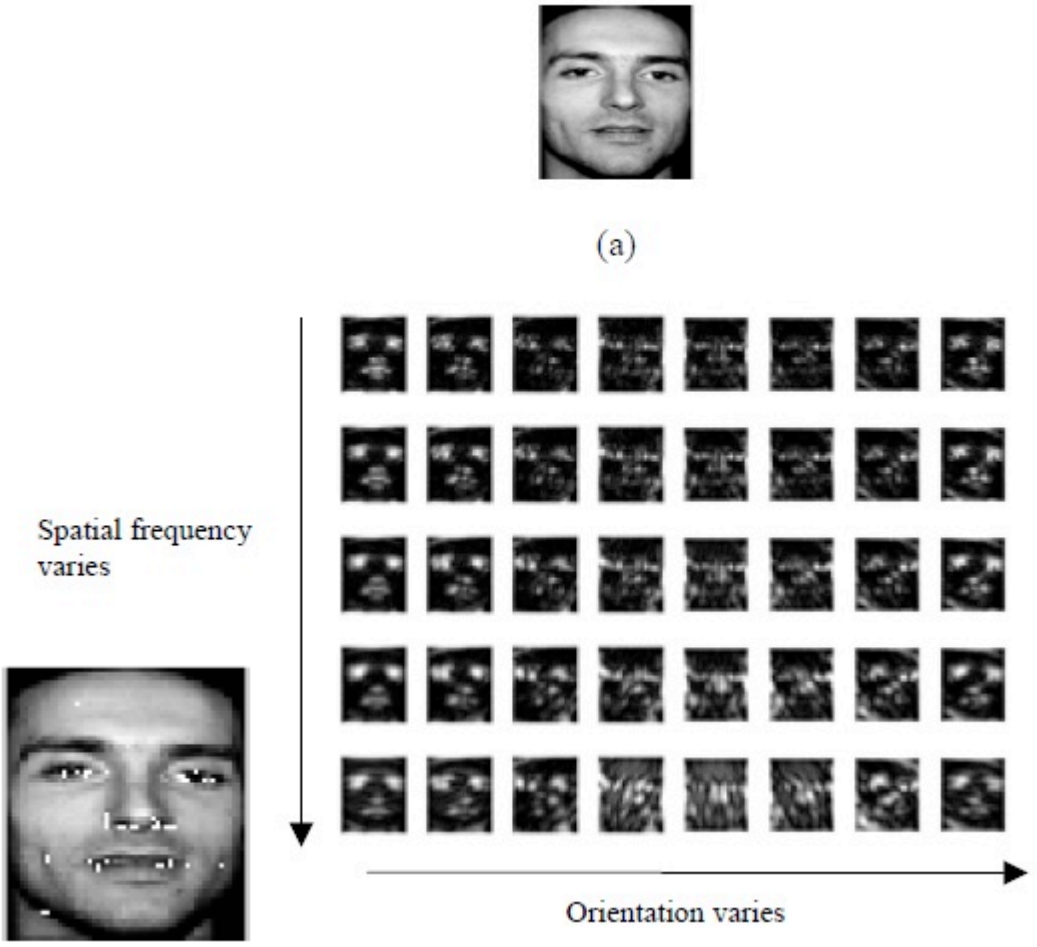
– by hand …



Where to perform convolution

- Selecting peaks (high energized points) of the Gabor wavelet responses as feature points

– alignment for matching?



An application of Gabor filter: Elastic Bunch Graph Matching (EBGM)

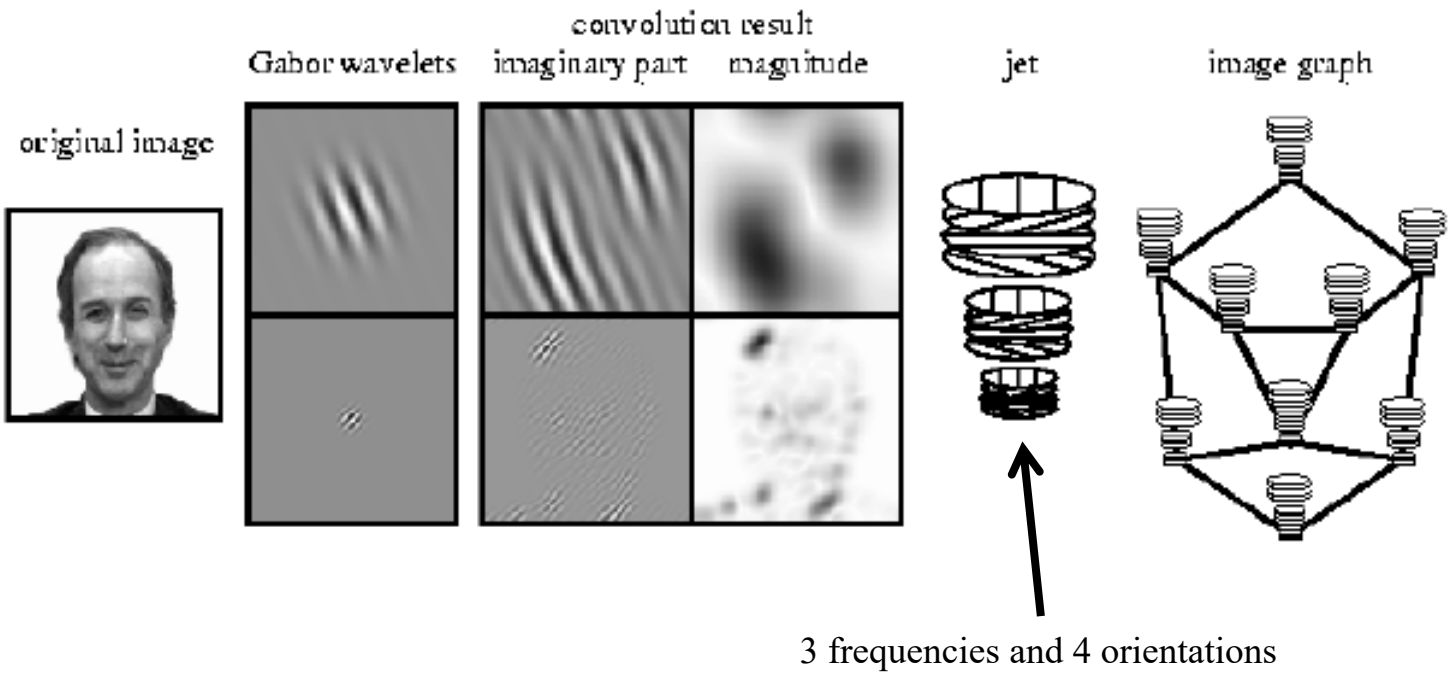
- The basic object representation is a labeled graph, defined as bunch graph, one for each pose, which collect class-specific information

- Edges are labeled with distance information and nodes are labeled with Gabor wavelet responses locally collected in jets.

- Bunch graphs are stacks of a moderate number (e.g., 70 in experiments by Wiskott et al.) of different faces, jet-sampled in an appropriate set of fiducial points (placed over eyes, mouth, contour, etc.).

EBGM

- A jet describes a small patch of grey values in an image I around a given pixel p = (x; y). It is based on a wavelet transform (5 frequencies × 8 orientations) providing a set of 40 coefficients for one image point.
- A sparse collection of such jets together with some information about their relative location constitutes an image graph, used to represent an object, such as a face



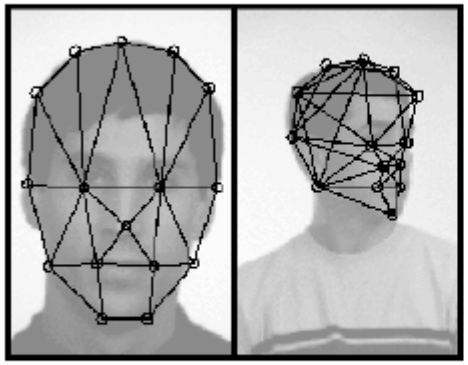
EBGM

- A bunch graph is created in two stages.
  - Its qualitative structure as a graph (a set of nodes plus edges) as well as the assignment of corresponding labels (jets and distances) for one initial image is designer-provided
  - The bulk of the bunch graph is extracted semi-automatically from sample images by matching the embryonic bunch graph to them, less and less often intervening to correct incorrectly identified fiducial points.

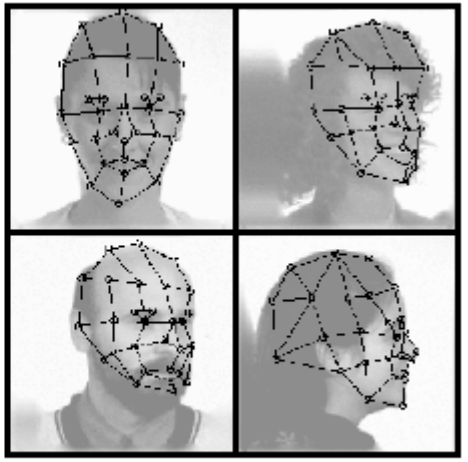
EBGM

- Individual faces: a labeled graph G representing a face consists of N nodes on a set of fiducial points (e.g. the pupils, the corners of the mouth, the tip of the nose, the top and bottom of the ears, etc.) and E edges between them. The nodes are labeled with jets and the edges are labeled with distances. This face image graph is object-adapted, since the nodes are selected from face-specific points (fiducial points)

- Graphs for different head pose differ in geometry and local features. Although the fiducial points refer to corresponding object locations, some may be occluded, and jets as well as distances vary due to rotation in depth.
- To be able to compare graphs for different poses, pointers are manually defined to associate corresponding nodes in the different graphs.



grids for face finding

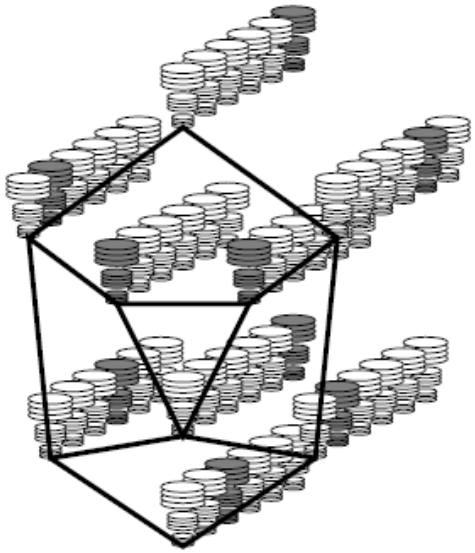


grids for face recognition

EBGM

- To find ducial points in new faces, one needs a general representation.
- The general representation should cover a wide range of possible variations in the appearance of faces, such as differently shaped eyes, mouths, or noses, different types of beards, variations due to sex, age, race, etc.
- A representative set of individual model graphs is combined into a stack-like structure, called a **face bunch graph** (FBG). Each model has the same grid structure and the nodes refer to identical fiducial points.
- A set of jets referring to one fiducial point is called a bunch.
  - an eye bunch may include jets from closed, open, female, and male eyes, e

bunch



face bunch graph

- During matching the most appropriate jet is selected in each bunch
- A simple graph G is matched with a FBG B

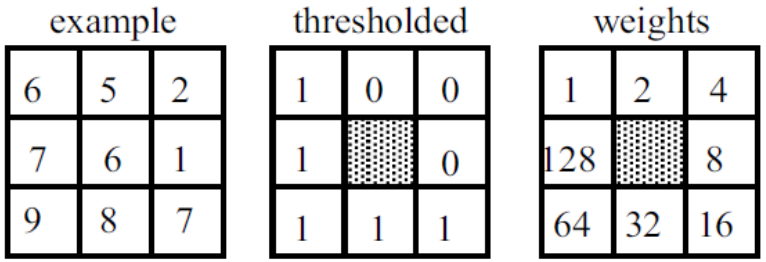
$$S(G,B) = \frac{1}{N}\sum_{n}\max_{m}(S_{\phi}(J_n^G, J_n^{B_m})) - \frac{\lambda}{E}\sum_{e}\frac{\left(\Delta\vec{x}_e^{\,G} - \Delta\vec{x}_e^{\,B}\right)^2}{\left(\Delta\vec{x}_e^{\,B}\right)^2}$$

Feature (Jets) comparison term.

Metric (distances) comparison term. (Distortions)

A texture-based operator: Local Binary Pattern (LBP)

- The **LBP** operator was presented by Ojala, Pietikainen and Maenpaa in 2002
- LBP works pixel by pixel. For each pixel, the basic version of the operator considers its neighborhood of size 3×3 and assigns to each pixel therein a binary value (0 or 1).
- Let p_c be a pixel and p a pixel in its neighborhood. The binary value is assigned to p by comparing its value with that of the pixel p_c : if p has a value greater or equal to that of p_c then p is assigned the value 1, otherwise the value 0. This operation is also called tresholding by the authors. The string of 0-1 provides a binary value.
- The histogram of values is an image feature.
- Furthermore, a kind of contrast measure referred to the central pixel can be computed, subtracting the average value of neighbors with a higher or equal value from the average vale of neighbors with a lower value.

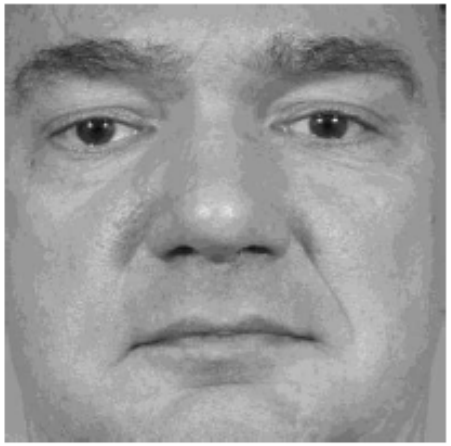


Pattern = **11110001**

LBP = 1 + 16 +32 + 64 + 128 = **241**

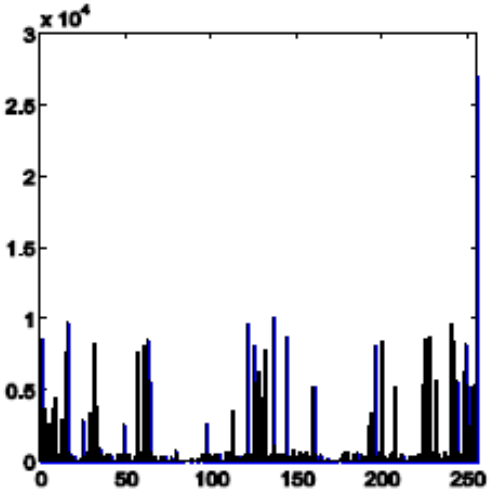
C = (6+7+8+9+7)/5 - (5+2+1)/3 = **4.7**

LPB



Input image

LBP image

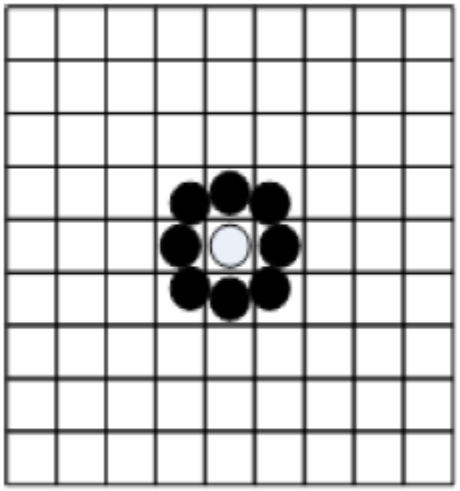


LBP histogram

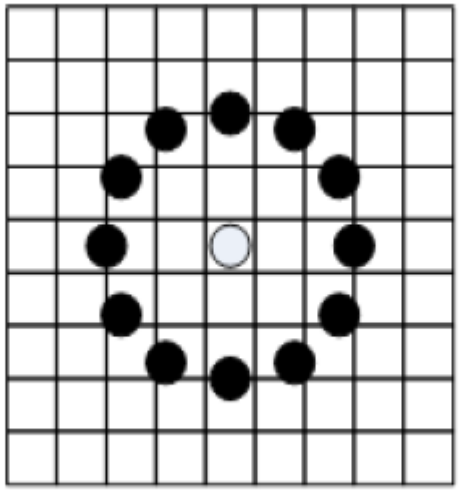
Image obtained by substituting the original pixel values with binary values obtained by LBP

LBP

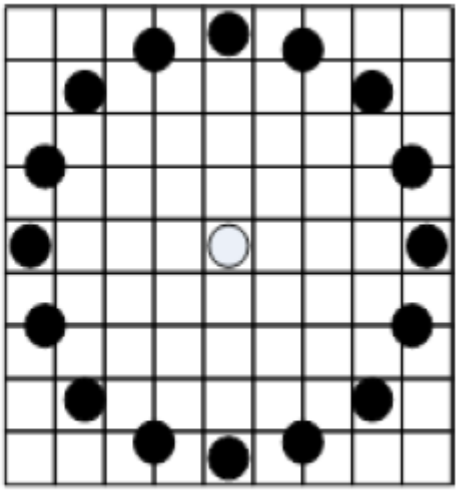
- The original method has been modified to handle interval of different radius.
- Two parameters: number of points P (possibly interpolated), interval radius R



P = 8, R = 1.0



P = 12, R = 2.5



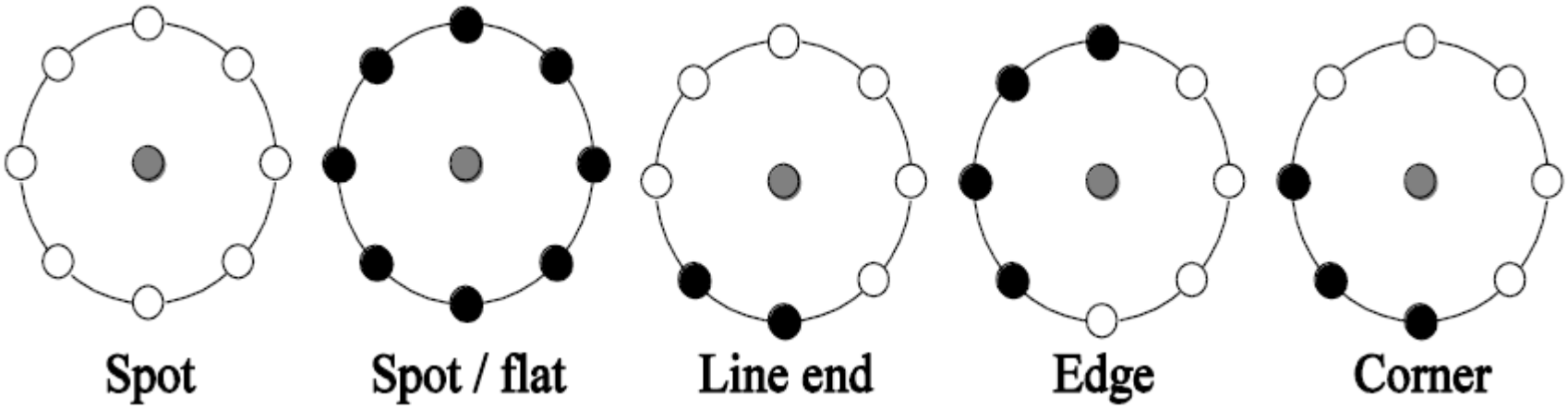
P = 16, R = 4.0

The value of the LBP code of a pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad s(x) = \begin{cases} 1, if\ x \geq 0; \\ 0, otherwise. \end{cases}$$

LBP

- A pattern is called uniform when, if considered in a circular fashion, it contains at most two transitions 0-1 or 1-0. For example the patterns
  10000011, 11110000, 00000000 are uniform.
- Uniform patterns are useful to save memory, since they are only P× (P-1) +2 over a total of P^2 (all non-uniform patterns are assigned to a single bin).
- Uniform patterns are relevant since they identify significant structures



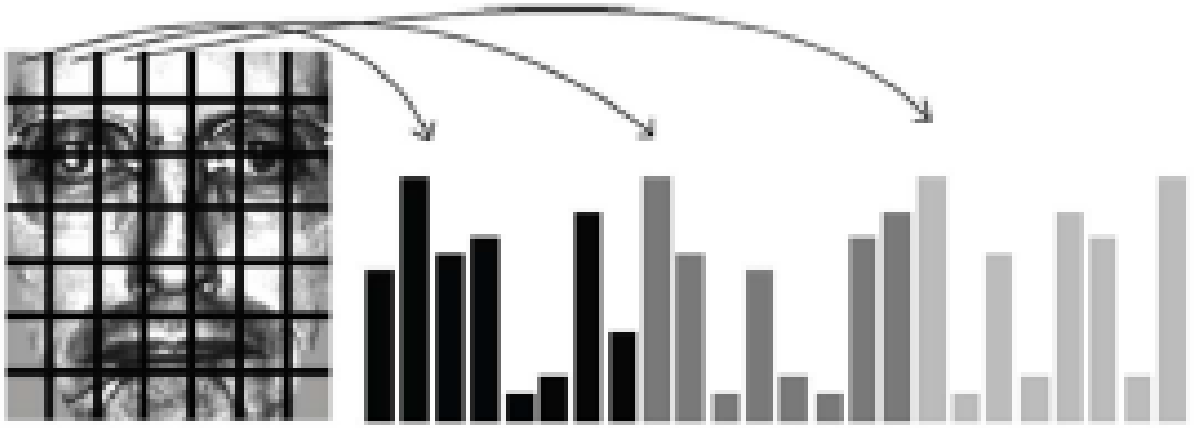
LBP

- The feature vector associated with an image is a histogram (possibly normalized) calculated as follows:

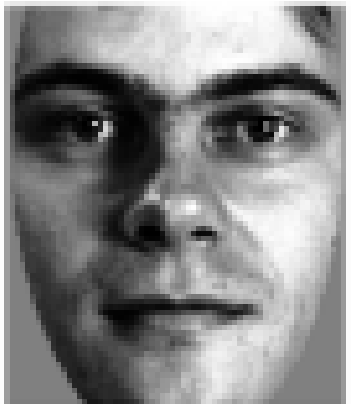
– The image is partitioned into subwindows, e.g., according to a square grid of k×k elements.

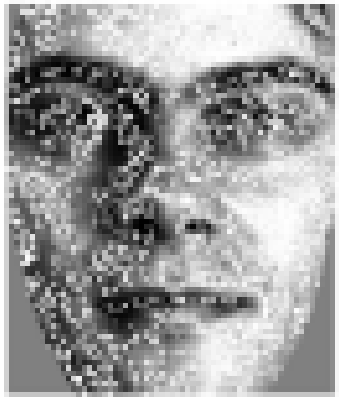
– For each sub-window a histogram is constructed in which each bin is associated with a specific pattern (the number of bins depend of the LBP type)

– The final feature vector is obtained by concatenating the histograms calculated for all



LBP



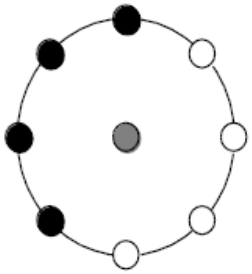


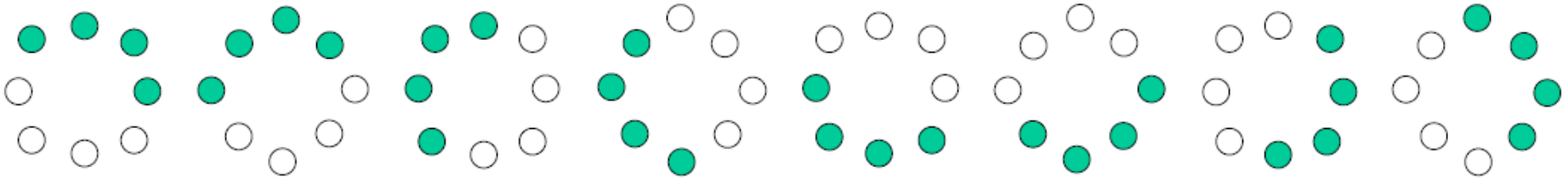
Pixels associated with uniform patterns

Pixels associated with non-uniform patterns

LBP

- Rotation modifies the LBP code





$$LBP^{ri}_{P,R} = min\{ROR(LBP_{P,R}, i) \mid i = 0,1,...,P-1\} \quad (8)$$

where $ROR(x,i)$ performs a circular bit-wise right shift on the P-bit number x i times. In terms of image pixels simply corresponds to rotating the neighbor set clockwise so many times that a maximal number of the most significant bits, starting from g_{P-1}, are 0.

A possible classification of recognition systems

- Global (holistic) appearance methods: based on the whole appearance of face. For example:
  - Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA): linear methods also used for dimensionality reduction → a face is represented as a linear combination of a «basis» of vectors or as an addition of subcomponents.
  - Some Artificial Intelligence (AI) approaches: neural Networks
  - Sparse representations: the «basis» is the whole set of training images.

Advantages and Disadvantages

- They do not destroy any of the information in the images by concentrating on only limited regions or points of interest
- Several of these algorithms have been modified and/or enhanced to compensate for PIE variations, and dimensions

- Most of these approaches start out with the basic assumption that all the pixels in the image are equally important.
- These techniques are computationally expensive
- They require a high degree of correlation between the test and training images.
- They do not perform effectively under large variations in PIE as well as scale, etc.

A possible classification of recognition systems

- Local or feature-based methods: based on relevant points or on local characteristics of single zones, possibly anatomically significant. For example:
  - Elastic Bunch Graph Matching (EBGM)
  - Local Binary Patterns (LBP): method used for computing textures

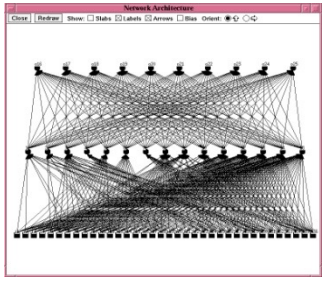
Advantages and Disadvantages

- The extraction of the feature points precedes the analysis done for matching the image to that of a known individual, therefore such methods are relatively robust to position variations in the input image.
- Feature-based schemes can be made invariant to size, orientation and/or lighting.
- They allow a compact representation of the face images and high speed matching.
- The implementer of any of these techniques has to make arbitrary decisions about which features are important.
- If the feature set lacks discrimination ability, no amount of subsequent processing can compensate for that intrinsic deficiency.

Systems in literature: intensity based

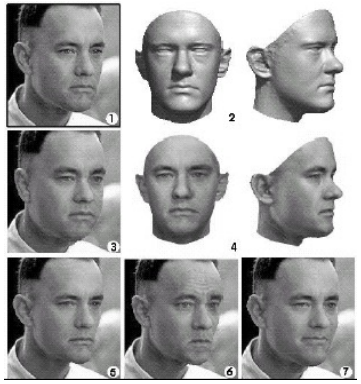
Image Based

- Eigenfaces
- ICA
- Neural Networks



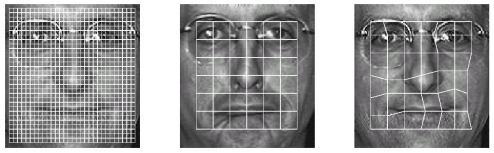
3D

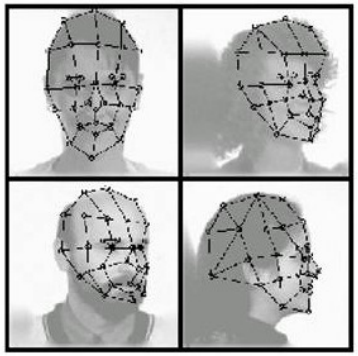
- 3D Morphable Models



Feature Based

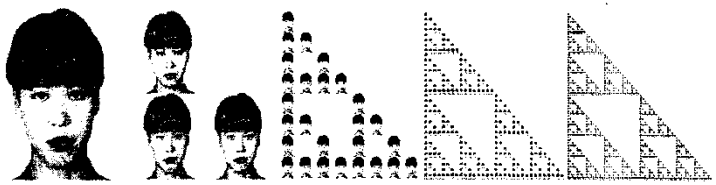
- Elastic Graph Matching

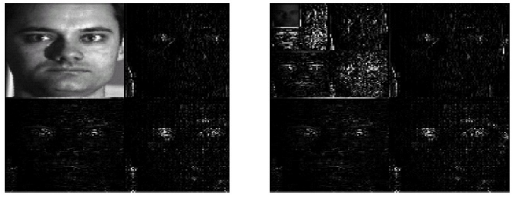




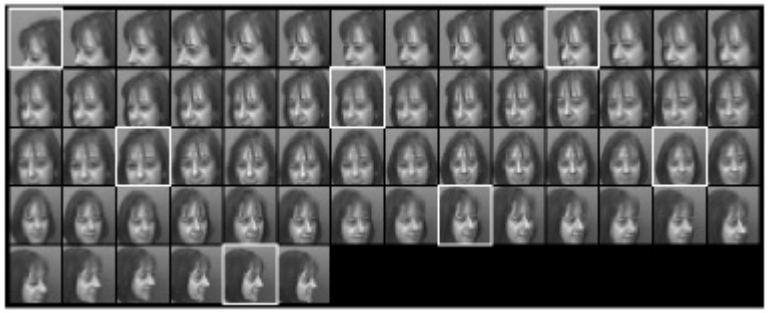
Hybrid

- Fractals
- Wavelets





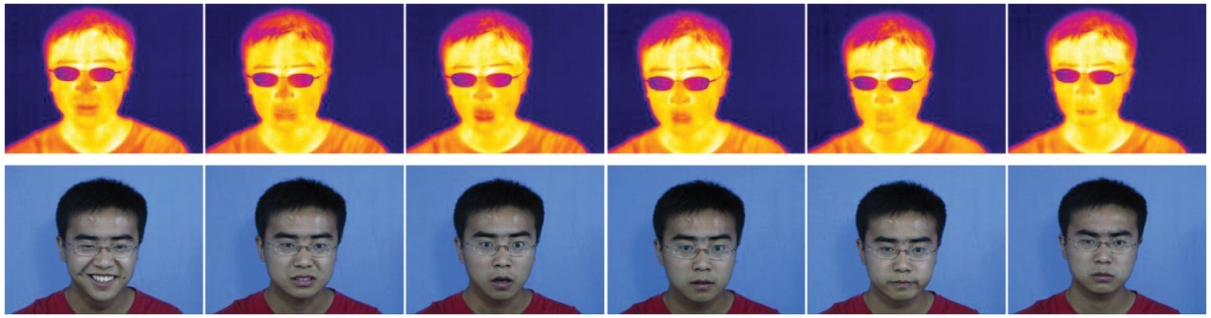
Systems in literature: more systems



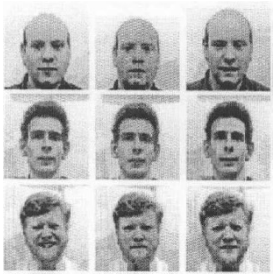
Face recognition from video sequences

Face Recognition from Other Sensory Inputs

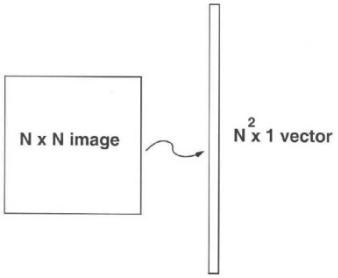
- Infra-red images
- Thermal images



Linear systems - Eigenfaces



Input images



Linearization

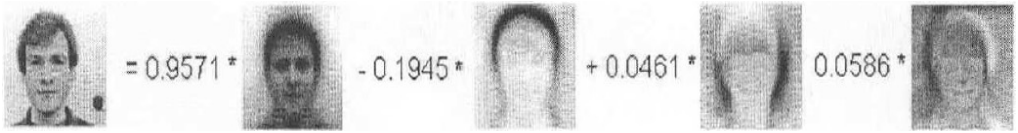
$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$$

Mean value over all faces

$$\Phi_i = \Gamma_i - \Psi$$

We subtract the mean from each vector

We solve a specific linear system to extract a set of reference faces



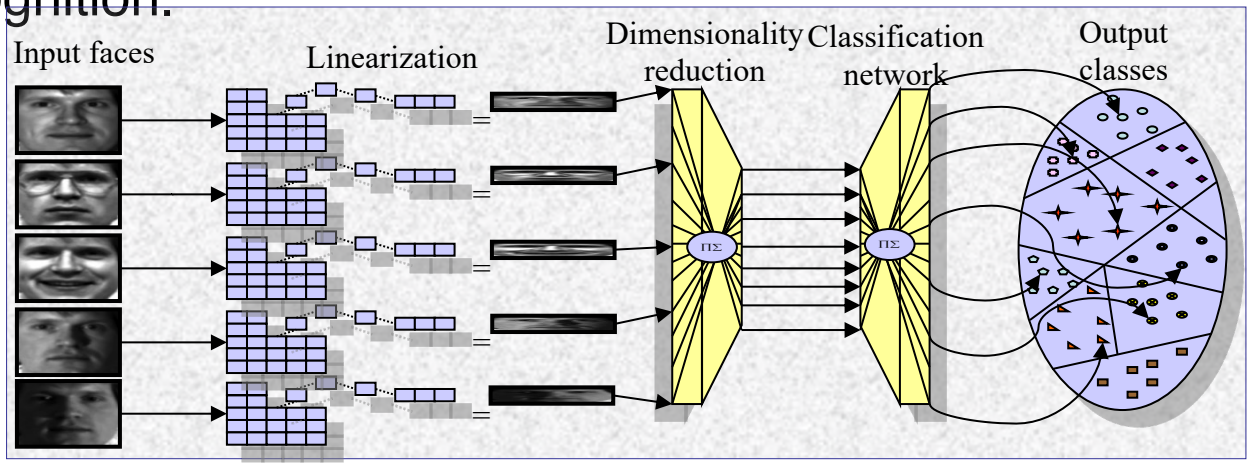
Each face is represented as a combination of the reference faces (is projected onto the space identified by the reference faces as axes). The coefficients used in the representation make up the feature vector.

Advantages and Disadvantages

- Identification phase is fast.
- If eigenvectors are preserved it is possible to reconstruct the initial information.
- Training phase is slow.
- If a significant number of new subjects is added it is necessary to retrain the system.
- The system is highly sensible to illumination and pose variations, to occlusions…

Neural Networks

- A neural network aims at simulating the way brain neurons work.
- Each neuron is represented by a mathematical function based on probabilities.
- In order to recognize faces, an optimal choice might be to use a neuron for each pixel. It is clear that this approach uses too many neurons.
- The solution is to use a neural network to « summarize » the image in a smaller vector. A second network performs the actual recognition.



Advantages and Disadvantages

- This approach reduces the ambiguity among subject belonging to similar classes.
- With suitable tricks they are robust to occlusions.
- They require more than one image for training.
- Some networks are subject to a number of problems:
  - Overfitting: the network has the same dimension of the input.
  - Overtraining: the network looses the ability to generalize.
  - Database size: when the number of subjects increases, they become inefficient.

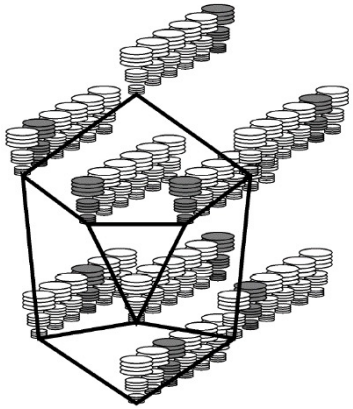
Notes

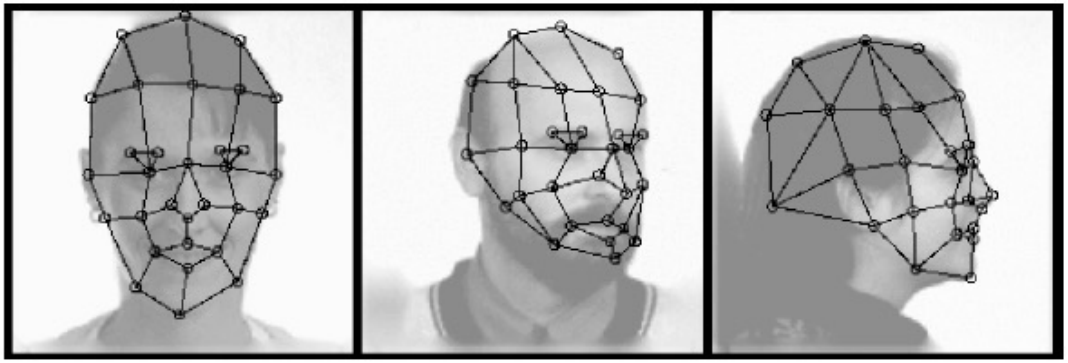
OVERFITTING. In statistics and machine learning, **overfitting** occurs when a statistical model describes random error or noise instead of the underlying relationship. Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. A model that has been overfit will generally have poor predictive performance, as it can exaggerate minor fluctuations in the data. (From Wikipedia)

OVERTRAINING. Overtraining occurs when the system “memorizes” patterns and thus looses the ability to generalize. It is an important factor in prediction systems as their primary use is to predict (or generalize) on input data that it has never seen.

Systems based on Graphs

- These systems use filters and localization functions to localize a set of reference points on the face.
- These points are connected by weighted archs so to create a graph.
- Each face is associated with a graph: matching two faces menas matching two graphs.





Advantages and Disadvantages

- They are robust with respect to pose variations.
- They are robust with respect to illumination variations.
- They do not require retraining the system.
- Training is slow.
- Testing is very slow because it requires graph matching (NP-Hard).

Thermogram

- The face image is acquired through a thermic sensor.
- The sensor detects temperature variations from face skin.
- The image is segmented and indexed.

(a) (b)

Advantages and Disadvantages

- They are robust with respect to illumination variations.
- They are robust with respect to time variations.
- They are efficient both indoor and outdoor.

- They require expensive capture devices.
- Capture devices are too sensitive to subject movements and provide a quite low resolution.
- They are affected by emotional state of the subject.
- A glass between the subject and the capture device makes the capture operation quite ineffective.

The deep revolution: DepFace by Facebook

- Deep networks soon changed face recognition applications too
- **DeepFace** is the facial recognition system used by Facebook for tagging images.

- It was proposed by researchers at Facebook AI Research (FAIR) at the 2014 IEEE Computer Vision and Pattern Recognition Conference (CVPR)

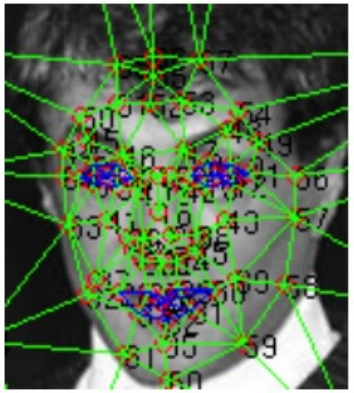
The deep revolution: DepFace by Facebook

- **DeepFace** includes an alignment step.
- The goal of this alignment part is to generate a frontal face from the input image that may contain faces from different pose and angles. The method proposed uses 3D frontalization of faces based on the fiducial (face feature points) to extract the frontal face.
- 1) Given an input image, identify the face using **six fiducial points**. These six fiducial points are 2 eyes, tip of the nose and 3 points on the lips. These feature points are used to detect faces in the image.

The deep revolution: DepFace by Facebook

- 2) Generates the 2D-face image cropped from the original image using 6 fiducial points.

- 3) Apply the 67 fiducial point map with their corresponding Delaunay Triangulation on the 2D-aligned cropped image. Generate a 3D-model using a generic 2D to 3D model generator.



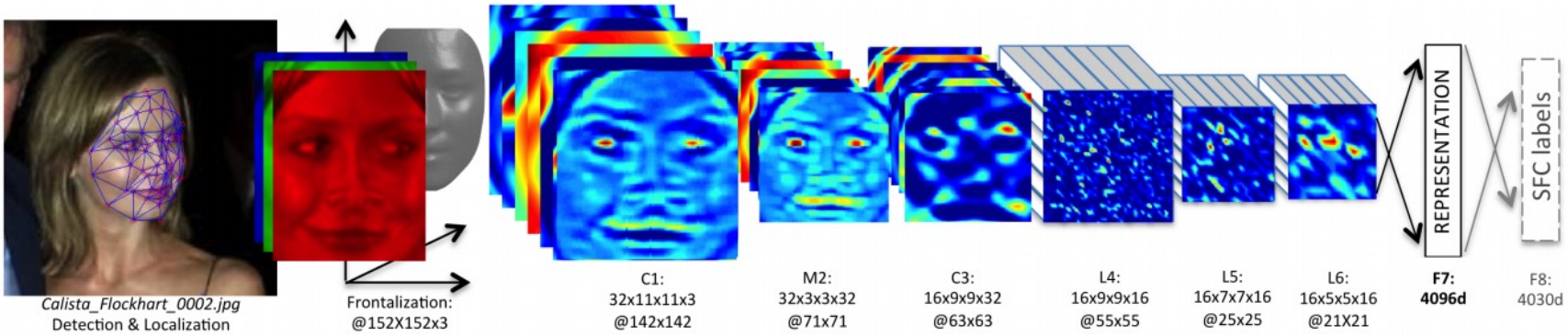
The deep revolution: DepFace by Facebook

- …
- …

- The final stage is ***frontalization of alignment***.

The deep revolution: DepFace by Facebook

- The aligned faces are the input for the deep architecture



The deep revolution: FaceNet by Google

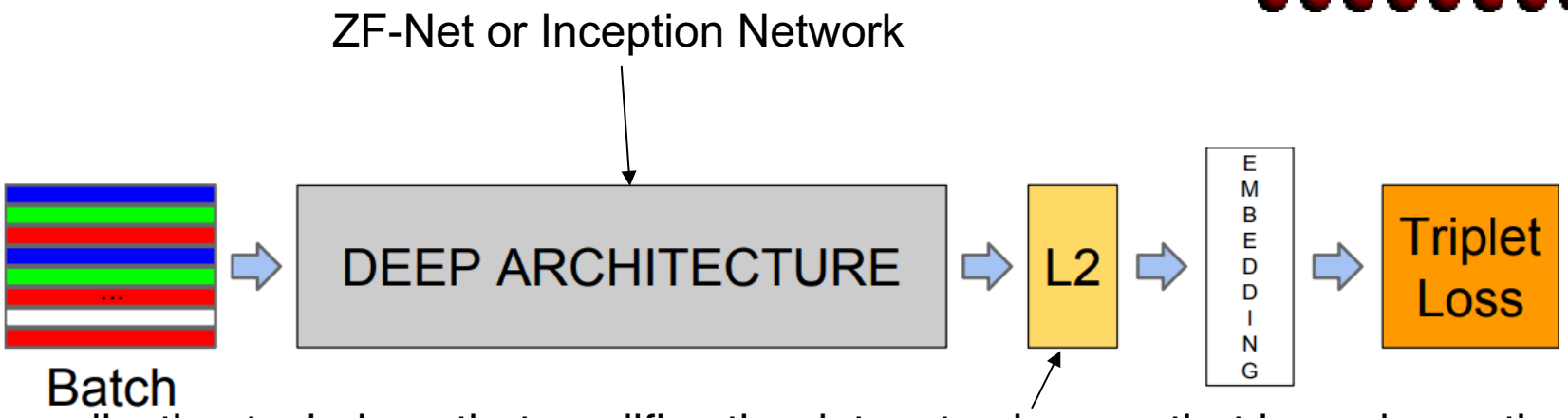
- **FaceNet** is the name of the facial recognition system that was proposed by Google Researchers in 2015 (*FaceNet: A Unified Embedding for Face Recognition and Clustering*)

- It achieved state-of-the-art results on the most challenging face recognition datasets such as Labeled Faces in the Wild (LFW) and Youtube Face Database.

- The proposed approach generates a high-quality face mapping from the images using deep learning architectures such as **ZF-Net** and **Inception Network**.

- Then it uses a method called **triplet loss** as a loss function to train this architecture.

FceNet architecture in more detail



A normalization technique that modifies the dataset values so that in each row the sum of the squares will always be up to 1. It is also called least squares. In other words, it rescales the feature vectors so that their L2 norm is 1 (each vector element is divided by the vector L2 norm).

The L2 norm calculates the distance of the vector coordinate from the origin of the vector space. As such, it is also known as the Euclidean norm as it is calculated as the Euclidean distance from the origin.

Example: we count the number of times each word appears in each document. Two documents might appear different simply because they have different lengths (the longer document contains more words). If we are interested in the *meaning* of the document, the length doesn't contribute to this. Normalizing lets us consider the frequency of words relative to each other, while removing the effect of total word count.

Triplet loss

The goal of **Triplet loss** function is to make the squared distance between two image embeddings of the same identity small despite independently of image condition and pose, whereas the squared distance between two images of different identities is large

We want to make sure that the embedding of the *anchor* image of a person is closer to the embedding of a positive image (image of the same person) as compared to the embedding of a negative image (image of another person)

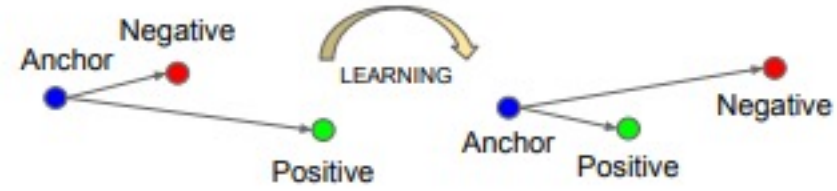
$$\|f\left(x_i^a\right) - f\left(x_i^p\right)\|_2^2 + \alpha < \|f\left(x_i^a\right) - f\left(x_i^n\right)\|_2^2$$

$$\forall \left(f\left(x_i^a\right), f\left(x_i^p\right), f\left(x_i^n\right)\right) \in \top$$

where α is the margin that is enforced to differentiate between positive and negative pairs and T is the image space.
Therefore the loss function is defined as the following :

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

Triplet mining



- The effective triplets are those with positive embeddings vert far from the anchor and negative ones very close

- We have the problem of **triplet mining**

Some references

- Abate, A. F., Nappi, M., Riccio, D., & Sabatino, G. (2007). 2D and 3D face recognition: A survey. *Pattern Recognition Letters*, *28*(14), 1885-1906.
- L. Sirovich and M. Kirby, "Low-dimensional Procedure for the Characterization of Human Faces," *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, Vol.4, pp.519-524, 1987.
- M. Turk and A. Pentland, "Eigenfaces For Recognition," *Journal Of Cognitive Neuroscience*, Vol.3, pp.71-86, 1991.
- Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *19*(7), 711-720.
- Gosselin, F., & Schyns, P. G. (2001). Bubbles: a technique to reveal the use of information in recognition tasks. *Vision research*, *41*(17), 2261-2271.
- Bhuiyan, A. A., & Liu, C. H. (2007). On face recognition using gabor filters. *World academy of science, engineering and technology*, *28*. Available at http://www.waset.org/publications/10902

Some references

- R. Polikar. The Wavelet Tutorial - Part I. http://users.rowan.edu/~polikar/WAVELETS/WTpart1.html (http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html)
- Wiskott, L., Fellous, J. M., Kuiger, N., & Von Der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *19*(7), 775-779.
- Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*(7), 971-987.
- Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *28*(12), 2037-2041.
- Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1701-1708).
- Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).