

Cours de Biométrie

Chapitre 2: Reconnaissance du Visage

Djellali H.
Master 1 RSI,
UBMA Université Algérie
Mars 2020

Sommaire

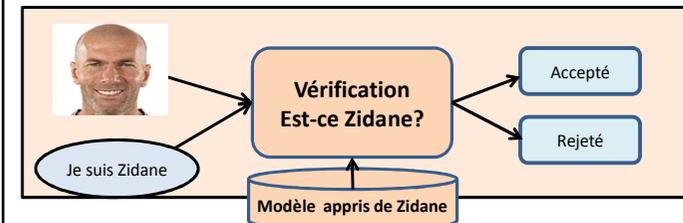
- Introduction
- Détection et Reconnaissance Automatique du Visage
- Les Applications
- Les Techniques d'Identifications
- ACP et Les Visages Propres
- Fisher Face
- Analyse Discriminante Linéaire (LDA)
- Local Binary Patterns LBP Histogramme

Reconnaissance Automatique du Visage

- La Reconnaissance automatique a pour but de reconnaître l'identité d'un individu automatiquement à partir de l'image du visage.
- Deux type de Reconnaissance
 - Identification
 - Verification

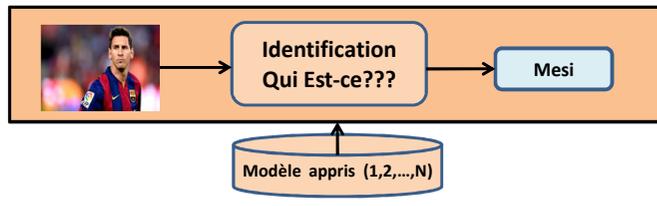
Vérification

- **Vérification:** La personne qui veut accéder au système sécurisé proclame son identité.
- L'algorithme de vérification compare l'image en entrée avec le modèle pré-appris de cet personne.
- Il prend un décision binaire (oui/ non).



Identification

- **Identification:** La personne qui veut accéder au système sécurisé ne proclame pas son identité.
- L'algorithme d'identification compare l'image en entrée avec tous les modèle pré-appris de toutes les personnes enregistrées. Il obtient un score Il prend une décision binaire (oui/ non).



Etapes Principales

- Etape 1:
 - Détection de Visage par Haar Cascade et Integral Images
 - Le but : Appliquer la détection de visage pour récupérer la région d'intérêt (le visage de la personne)
- Etape2: **Reconnaissance de Visages:**
 - 2.1 Apprentissage
 - 2.1.1 Préparer la base d'images et étiquettes
 - 2.1.2 Apprentissage par Analyse en Composantes principales
 - 2.1.3 Apprentissage par Analyse Discriminante Linéaire (fisher Face)
- 3. Etape 3: La Reconnaissance:
 - La Prédiction des visages inconnus
 - Evaluation des Résultats

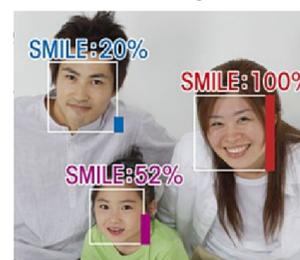
Applications de la Reconnaissance des visages

- Digital photography
- Surveillance



Applications de la Reconnaissance des visages

- Digital photography
- Surveillance
- Album organization
- Emotion Recognition

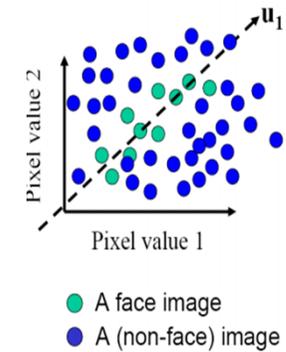


Les Visages

- L'espace des visages
- Une image est un point dans un espace a grande dimension.
- – Si l'image est representé en niveaux de gris
NxM image est un point dans $R(NM)$
- – Exemple. 100x100
- image = 10,000 dim

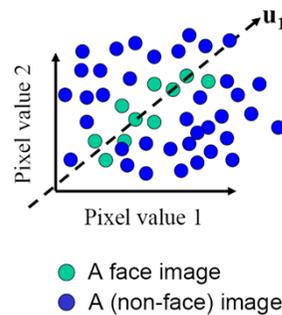
Analyse en Composante Principale

- Peu de vecteurs de grande dimension correspondent à des images de visages valides.
- On veut modéliser ce sous espace d'images de visages.



The space of all face images

- Eigenface ou Visage propre :
- construire Un sous espace linéaire qui explique les variations dans l'ensemble des images (visages)



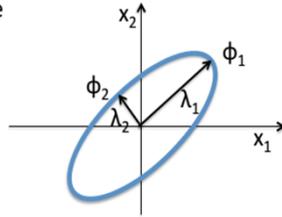
PCA

- Principal Component Analysis (PCA)
- P. Belhumeur, J. Hespanha, and D. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection". IEEE Transactions on pattern analysis and machine intelligence 19 (7): 711. 1997.
- Turk and Pentland, Eigenfaces for Recognition, Journal of Cognitive Neuroscience 3 (1): 71-86

Formuler PCA

- If x is Gaussian with covariance Σ , the equiprobability contours are the ellipses whose

- Principal components ϕ_i are the eigenvectors of Σ
- Principal lengths λ_i are the eigenvalues of Σ



- by computing the eigenvalues we know the data is
 - Not flat if $\lambda_1 \approx \lambda_2$
 - Flat if $\lambda_1 \gg \lambda_2$

Algorithme d'apprentissage ACP

► Given sample $\mathcal{D} = \{x_1, \dots, x_n\}$, $x_i \in \mathcal{R}^d$

- compute sample mean: $\hat{\mu} = \frac{1}{n} \sum_i(x_i)$
- compute sample covariance: $\hat{\Sigma} = \frac{1}{n} \sum_i(x_i - \hat{\mu})(x_i - \hat{\mu})^T$
- compute eigenvalues and eigenvectors of $\hat{\Sigma}$

$$\hat{\Sigma} = \Phi \Lambda \Phi^T, \Lambda = \text{diag}(\sigma_1^2, \dots, \sigma_n^2) \Phi^T \Phi = I$$

- order eigenvalues $\sigma_1^2 > \dots > \sigma_n^2$
- if, for a certain k , $\sigma_k \ll \sigma_1$ eliminate the eigenvalues and eigenvectors above k .

Le Test ACP

► Given principal components $\phi_i, i \in 1, \dots, k$ and a test sample $T = \{t_1, \dots, t_n\}$, $t_i \in \mathcal{R}^d$

- subtract mean to each point $t'_i = t_i - \hat{\mu}$
- project onto eigenvector space $y_i = A t'_i$ where

$$A = \begin{bmatrix} \phi_1^T \\ \vdots \\ \phi_k^T \end{bmatrix}$$

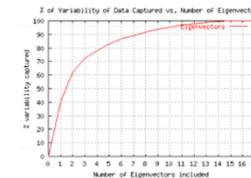
- use $T' = \{y_1, \dots, y_n\}$ to estimate class conditional densities and do all further processing on y .

Inspired by N. Vasconcelos



Rule of thumb for finding the number of PCA components

- A natural measure is to pick the eigenvectors that explain p% of the data variability
 - Can be done by plotting the ratio r_k as a function of k

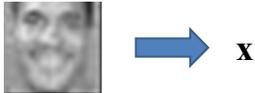


$$r_k = \frac{\sum_{i=1}^k \lambda_i^2}{\sum_{i=1}^n \lambda_i^2}$$

- E.g. we need 3 eigenvectors to cover 70% of the variability of this dataset

Starting idea of “eigenfaces”

1. L'image du visage est vectorisé



2. Reconnaître un visage par les plus proches voisins



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

The space of all face images

- l'image du visage vue comme un vecteur a deux dimensions de valeur de pixel, les images de visages sont de hautes dimensions
 - 100x100 image = 10,000 dimensions
 - Lenteur et grands espace de stockage
- Mais peu 10,000-dimensional vectors sont des images valides de visages. On veut modéliser l'espace des visages



Analyse en composante Principale(PCA)

- Given: N data points $\mathbf{x}_1, \dots, \mathbf{x}_N$ in \mathbb{R}^d
- On veut trouver un nouvel ensemble de caractéristiques qui sont en combinaison linéaire avec les caractéristiques originales .

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

($\boldsymbol{\mu}$: La moyenne)

Choisir le vecteur unité \mathbf{u} dans \mathbb{R}^d qui capture la plus grande variance des données.

Analyse en composante Principale(PCA)

- Direction qui maximise la variance des données projetées:

$$\begin{aligned} \text{Maximize} \quad & \frac{1}{N} \sum_{i=1}^N \underbrace{\mathbf{u}^T(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{u}^T(\mathbf{x}_i - \boldsymbol{\mu}))^T}_{\text{Projection of data point}} \quad \text{subject to } \|\mathbf{u}\|=1 \\ & = \mathbf{u}^T \left[\frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \right] \mathbf{u} \\ & \quad \underbrace{\hspace{10em}}_{\text{Covariance matrix of data}} \\ & = \mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u} \end{aligned}$$

The direction that maximizes the variance is the eigenvector associated with the largest eigenvalue of $\boldsymbol{\Sigma}$

Implementation

- La matrice de Covariance est enorme (N^2 for N pixels)
- But typically # examples $\ll N$
- Astuce
 - X est une matrice des données d'apprentissage normalisées
 - Solve for vecteurs propre u of XX^T au lieu de $X^T X$
 - Then $X^T u$ est le vecteur propre de covariance $X^T X$
 - Necessaire de normalizer (pour avoir unit length vector)

Eigenfaces (PCA on face images)

1. Calculer la matrice de covariance de l'image du visage
2. Calculer les composantes principales visages propre ("eigenfaces")
 - K eigenvectors avec les plus grandes valeurs propres
3. Représenter toutes les images des visages dans le dataset comme une combinaison linéaire des visages propre (eigenfaces)
 - Appliquer les K plus proche voisins sur ces coefficients

De l'article de : M. Turk and A. Pentland,
[Face Recognition using Eigenfaces](#), CVPR 1991

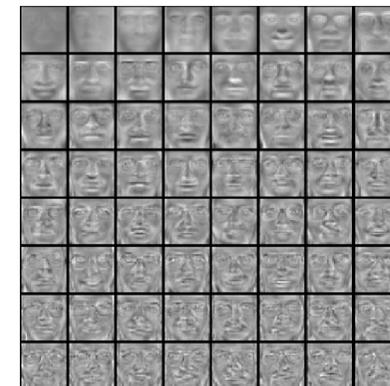
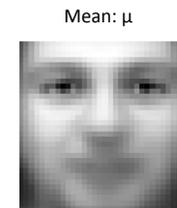
Eigenfaces example

- Training images
- x_1, \dots, x_N



Exemple de Visage propre (Eigenfaces)

Top vecteurs propre : u_1, \dots, u_k



Visualization of eigenfaces

Principal component (vecteur propre = eigenvector) u_k

$\mu + 3\sigma_k u_k$

$\mu - 3\sigma_k u_k$

Representation and reconstruction

- Face x in "face space" coordinates:

$$x \rightarrow [u_1^T(x - \mu), \dots, u_k^T(x - \mu)]$$

$$= w_1, \dots, w_k$$

Representation et reconstruction

- Face x in "face space" coordinates:

$$x \rightarrow [u_1^T(x - \mu), \dots, u_k^T(x - \mu)]$$

$$= w_1, \dots, w_k$$

Reconstruction:

$$\hat{x} = \mu + w_1 u_1 + w_2 u_2 + w_3 u_3 + w_4 u_4 + \dots$$

Eigenfaces

Calculé a partir de 400 face images de ORL face database

Reconstruction



Après calcul des visages propres (eigenfaces) utilisant 400 images de visages de ORL face database

Note

Preserver la variance (minimizing MSE) ne mène pas



Nécessairement à une bonne reconstruction.

References

- Turk and Pentland, Eigenfaces for Recognition, Journal of Cognitive Neuroscience 3 (1): 71-86.
- P. Belhumeur, J. Hespanha, and D. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection". IEEE Transactions on pattern analysis and machine intelligence 19 (7): 711-726. 1997.

L'algorithme Visage Propre Apprentissage

Vectoriser chaque image

1. Align training images x_1, x_2, \dots, x_N



Note that each image is formulated into a long vector!

2. Compute average face $\mu = \frac{1}{N} \sum x_i$

3. Compute the difference image (the centered data matrix)

$$X_c = \begin{bmatrix} | & & | \\ x_1 & \dots & x_n \\ | & & | \end{bmatrix} - \begin{bmatrix} | & & | \\ \mu & \dots & \mu \\ | & & | \end{bmatrix}$$

$$= X - \mu 1^T = X - \frac{1}{n} X 1 1^T = X \left(I - \frac{1}{n} 1 1^T \right)$$

L'algorithme Visage Propre Apprentissage (suite)

4. Compute the covariance matrix

$$\Sigma = \frac{1}{n} \begin{bmatrix} | & & | \\ x_1^c & \dots & x_n^c \\ | & & | \end{bmatrix} \begin{bmatrix} - & x_1^c & - \\ \vdots & \vdots & \vdots \\ - & x_n^c & - \end{bmatrix} = \frac{1}{n} X_c X_c^T$$

5. Compute the eigenvectors of the covariance matrix Σ
6. Compute each training image x_i 's projections as

$$x_i \rightarrow (x_i^c \cdot \phi_1, x_i^c \cdot \phi_2, \dots, x_i^c \cdot \phi_K) \equiv (a_1, a_2, \dots, a_K)$$

7. Visualize the estimated training face x_i

$$x_i \approx \mu + a_1 \phi_1 + a_2 \phi_2 + \dots + a_K \phi_K$$

L'algorithme Visage Propre Apprentissage (suite)



6. Compute each training image x_i 's projections as

$$x_i \rightarrow (x_i^c \cdot \phi_1, x_i^c \cdot \phi_2, \dots, x_i^c \cdot \phi_K) \equiv (a_1, a_2, \dots, a_K)$$

7. Visualize the estimated training face x_i

$$x_i \approx \mu + a_1 \phi_1 + a_2 \phi_2 + \dots + a_K \phi_K$$

Algorithme Visage Propre : Test

- Le Test

1. Take query image t
2. Project y into eigenface space and compute projection

$$t \rightarrow ((t - \mu) \cdot \phi_1, (t - \mu) \cdot \phi_2, \dots, (t - \mu) \cdot \phi_K) \equiv (w_1, w_2, \dots, w_K)$$

3. Compare projection w with all N training projections

- Simple comparison metric: Euclidean
- Simple decision: K-Nearest Neighbor

(note: this "K" refers to the k-NN algorithm, is different from the previous K's referring to the # of principal components)

Reconstruction et erreur

- K=4



- K=100



Sélectionner les K premiers visages propres réduit la dimension

Résumé visage Propre

Avantages

- Non-iterative, globally optimal solution

Limitations

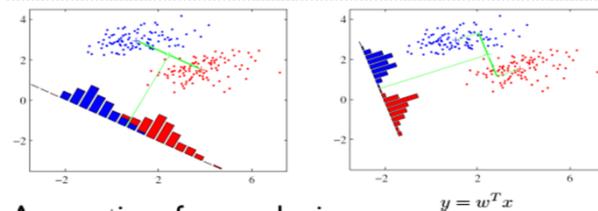
- PCA projection is **optimal for reconstruction** from a low dimensional basis, but **may NOT be optimal for discrimination...**

Reconnaissance de Visages

- Autres Approches seront présentés dans la suite de ce cours:
- Analyse discriminante linéaires
- Fisherface :
 - Réduire la dimension et trouver la meilleure séparation entre différentes classes)
- Les Formes Binaires Locales : les informations sur la texture. (Local binary patterns) LBP

ALD

Linear Discriminant Analysis



- Assumptions for new basis:
 - ☑ Maximize distance between projected class means
 - ☑ Minimize projected class variance

Comparaison

- PCA: réduction de dimension et préservant la variance dans l'espace de grande dimension
- LDA: réduction de dimension en préservant la discrimination entre classes.

A lire

- C. Bishop, “Neural Networks for Pattern Recognition”, Oxford University Press, 1998, Chapter 1.
- Forsyth and Ponce, Chap 22.3 (through 22.3.2-
-eigenfaces)
- Turk, M. and Pentland, A. *Eigenfaces for recognition*. Journal of Cognitive Neuroscience, 1991
- Viola, P. A. and Jones, M. J. (2004). Robust real-time face detection. *IJCV*, 57(2), 137–154.

Liens utiles

- Steve Seitz, CSE [455/576](#), previous quarters
- Fei-Fei, Fergus, Torralba, [CVPR'2007 course](#)
- Efros, [CMU 16-721](#) Learning in Vision
- Freeman, [MIT 6.869](#) Computer Vision: Learning
- Linda Shapiro, CSE 576, [Spring 2007](#)

Application en Python et opencv

- Reconnaissance des Visages par:
 - 1- Analyse en Composante Principale EigenFace
 - 2- Fisher Face basée Analyse Discriminante Linéaire (LDA)
 - initialiser le reconnaisseur
 - `recognizer = cv2.face.FisherFaceRecognizer_create()`
 - 3- Local Binary Patterns
- ```
print("Initialize Face Recognizer")
recognizer=cv2.face.LBPHFaceRecognizer_create()
apprentissage en python
recognizer.train(x_train, np.array(y_labels))
#enregistrer le modele appris:
recognizer.save("facetrainner.yml")
La prediction:
id_conf = recognizer.predict(imroi_gray)
```