

Apprentissage et reconnaissance de visage

Objectif

Identifier un visage parmi une base de données de visages de référence.

Compétences visées

1. Donner des fonctions statistiques permettant l'analyse de données
2. Introduire une méthode d'apprentissage pour la reconnaissance de visages

Note : ce TD est inspiré du travail de Thomas Robert, enseignant à l'Université de la Sorbonne, Paris.

1 Présentation de la méthode

A partir d'une image d'un visage, l'objectif est de déterminer l'identité de la personne correspondante. On dispose d'une base de données de visages de toutes les personnes connues par le système. Un vecteur de caractéristiques est associé à chaque visage. Ces caractéristiques sont supposées être invariantes pour une même personne, et différentes d'une personne à l'autre. La reconnaissance consiste alors à comparer le vecteur de caractéristiques du visage à reconnaître avec celui de chacun des visages de la base. Ceci permet de retrouver la personne ayant le visage le plus ressemblant, qui est celui dont le vecteur est le plus similaire.

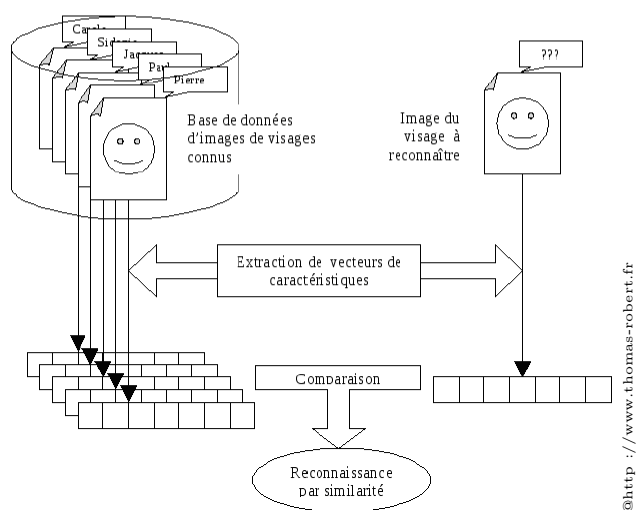


FIGURE 1 – Principe de reconnaissance de visage

1.1 Base d'images

On utilise la base d'images **yalefaces**¹. Dans cette base, les visages ont tous été traités afin de les recaler et rogner à la taille 64×64 pixels, de sorte que les images puissent être comparées pixel à pixel. Cette base contient

1. Base de visages développée à l'université de Yale, <http://vision.ucsd.edu/content/yale-face-database>

120 images en niveaux de gris, représentant les visages de 15 individus. Il y a 8 images par individu, chacune correspondant à une catégorie d'images variant selon les critères suivants :

- variation de l'expression du visage : normal, sad, sleepy, surprised, wink, happy
- variation des accessoires : glasses, noglasses

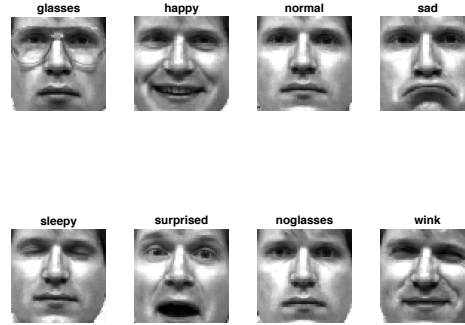


FIGURE 2 – 8 catégories de visage

Une image de visage est une matrice de 64×64 pixels. Pour faciliter les traitements effectués pour extraire les caractéristiques associées au visage, la matrice 64×64 est mise sous la forme d'un vecteur de dimension 4096×1 (voir figure 3).

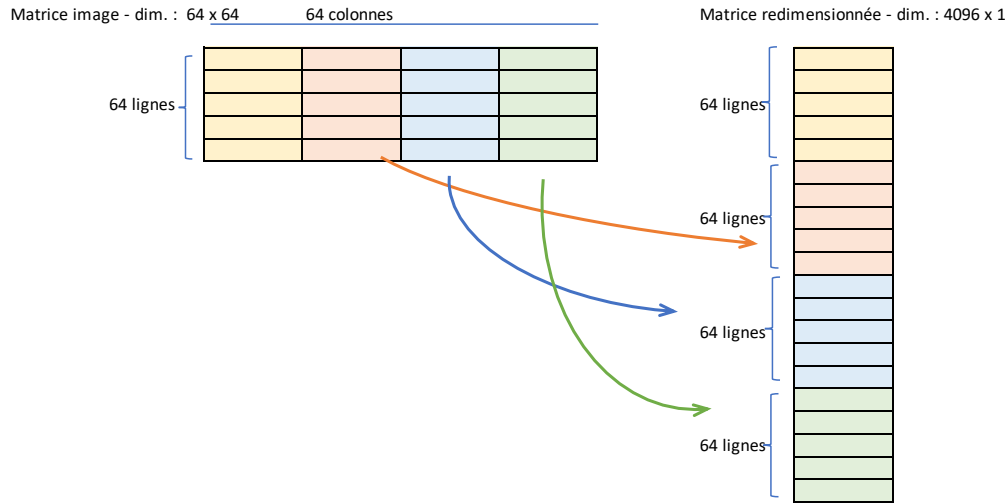


FIGURE 3 – Redimensionnement de la matrice

La matrice \mathbf{X} est composée de 2 matrices, $\mathbf{X}_{\text{train}}$ et \mathbf{X}_{test} . La matrice $\mathbf{X}_{\text{train}}$, de dimension $4096 \times N_{\text{train}}$, contient des images utilisées comme jeu d'entraînement ($N_{\text{train}} = 90$ images d'entraînement). L'autre matrice, \mathbf{X}_{test} , de taille $4096 \times N_{\text{test}}$, contient des images utilisées comme jeu de test ($N_{\text{test}} = 30$ images de test).

1.2 Principe de la reconnaissance de visages

On calcule le visage "moyen", x_{moy} , qui correspond à la valeur moyenne des visages d'entraînement, sorte de résumé de l'ensemble des visages. Une image de visage, x , peut être exprimée comme le résumé de l'ensemble des visages (visage moyen) plus ses caractéristiques propres (combinaison linéaire de vecteurs caractéristiques u_h) :

$$x = x_{\text{moy}} + \sum_h a_h u_h \quad (1)$$

où a_h représente le poids du vecteur caractéristique d'indice h dans le visage x .



FIGURE 4 – Projection d'un objet 3D en 2D sur différents plans

2 Travail demandé

2.1 Chargement de la base et centrage des visages

1. Importez le fichier `YaleFaces.mat`, disponible sur Arche.
 2. Calculez le visage moyen x_{moy} du jeu d'entraînement \mathbf{X}_{train} (x_{moy} de taille 4096×1), à l'aide de la fonction `mean`.
 3. Centrez les visages ($\mathbf{X}_c = \mathbf{X}_{train} - x_{moy}$) du jeu d'entraînement (\mathbf{X}_c de taille 4096×90).
 4. Créez une fonction qui permet de recomposer la matrice image d'un visage. Cette fonction prend une matrice x de dimension 4096×1 en argument d'entrée et retourne une matrice de dimension 64×64 .
- A l'aide de votre fonction, affichez le visage moyen, ainsi que quelques visages du jeu d'entraînement accompagnés des visages centrés associés. Utilisez la fonction `mat2gray` pour convertir la matrice de type `double` contenant les images vers le type "image en niveau de gris". Voici un exemple de résultat attendu :



FIGURE 5 – Visage moyen, visages d'origines et visages centrés associés

2.2 Calcul des vecteurs caractéristiques

Pour obtenir la combinaison linéaire des vecteurs caractéristiques de l'équation (1), il faut mesurer la ressemblance d'un visage de la base avec les autres de la base. La matrice de covariance permet de mesurer cette ressemblance. Elle est donnée par le produit matriciel $\mathbf{X}_c \times \mathbf{X}_c^T$, de taille 4096×4096 .

La matrice \mathbf{X}_c de taille 4096×90 intervenant dans la mesure de ressemblance, doit être réécrite sous la forme (décomposition en valeurs singulières) :

$$\mathbf{X}_c = \mathbf{U} \times \mathbf{S} \times \mathbf{V}^T \quad (2)$$

où \mathbf{U} et \mathbf{V} sont des matrices orthonormales ($\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{Id}^{4096}$ et $\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{Id}^{90}$). La matrice \mathbf{S} de dimension 4096×90 a toutes ses composantes nulles sauf celles sur la diagonale principale. Cette décomposition présente les propriétés suivantes :

- les colonnes de \mathbf{V} sont les vecteurs propres de $\mathbf{X}_c^T \mathbf{X}_c$;
- les colonnes de \mathbf{U} sont les vecteurs propres de $\mathbf{X}_c \mathbf{X}_c^T$;
- la diagonale de \mathbf{S} contient les valeurs singulières de \mathbf{X}_c , égales à la racine carrée des valeurs propres λ_k de $\mathbf{X}_c \mathbf{X}_c^T$.

Sous Matlab, pour effectuer la décomposition en valeurs singulières, la commande suivante est utilisée :

```
[u, s, v] = svd(Xc, 0)
```

Cette commande renvoie des matrices \mathbf{U} , \mathbf{S} et \mathbf{V} de taille 4096×90 , 90×90 et 90×90 .

1. A partir des images centrées de la base d'entraînement, calculez la matrice \mathbf{U} contenant les vecteurs caractéristiques recherchés et les valeurs propres λ_k associées. Les valeurs propres représentent l'importance du

vecteur caractéristique. Plus la valeur propre est élevée, plus le vecteur de caractéristique associé apportera des informations sur le visage.

2. Calculez le pourcentage associée à chaque valeur propre par rapport à la somme des valeurs propres.

3. Tracez la courbe de la somme cumulée des valeurs propres en pourcentage, afin de voir combien de variation est capturée par les K premiers vecteurs caractéristiques.

Pour $K = 15$, quel est le pourcentage cumulé des valeurs propres ?

Combien de vecteurs caractéristiques K sont nécessaires pour obtenir une "bonne" reconstruction (au moins 75% de l'information) ?

2.3 Projection dans le sous-espace des visages

Une fois que la forme (1) a été déterminée (parties 2.1 et 2.2), la phase de projection et de comparaison avec les vecteurs caractéristiques peut démarrer. Dans la suite, un nombre réduit K de vecteurs caractéristiques sera utilisé pour représenter un visage.

Chaque visage possède plusieurs représentations :

- son image d'origine, représentée par le vecteur x (4096×1) ;
- les coordonnées de l'image projetée z dans la base des vecteurs caractéristiques ($K \times 1$) :

$$z = \mathbf{U}_K^T (x - x_{moy}) \quad (3)$$

- son image reconstruite \hat{x} à partir des vecteurs caractéristiques (4096×1) :

$$\hat{x} = x_{moy} + \mathbf{U}_K z \quad (4)$$

L'erreur de reconstruction e est définie comme la distance entre une image et l'image reconstruite associée :

$$e = \|x - \hat{x}\|_2 = \sqrt{\sum_p (x(p) - \hat{x}(p))^2} \quad (e \text{ de taille } 1 \times 1) \quad (5)$$

1. A partir d'une image x de la base d'entraînement et de l'image moyenne x_{moy} , calculez les coordonnées de l'image projetée z .
2. A partir de z , reconstruisez l'image \hat{x} .
3. Calculez l'erreur de reconstruction entre x et \hat{x} .
4. Affichez l'image d'origine ainsi que l'image reconstruite pour différentes valeurs de K (par exemple, $K = 5, 10, 30, 50, 90$).
6. A partir d'une image x de la base de test, effectuez la même procédure (calcul des coordonnées de l'image projetée, reconstruction de l'image d'origine et calcul de l'erreur de reconstruction).

2.4 Identification de visages

Une fois les données projetées, il est possible de réaliser les comparaisons et d'identifier les visages. A chaque visage d'entraînement x_k^{train} est associée une identité, sous la forme d'un numéro $id_{train}(k)$. L'objectif dans cette partie est d'identifier un visage x^{test} à partir des visages d'entraînement. La méthode la plus simple consiste à comparer la projection z^{test} du visage test x^{test} avec la projection z_k^{train} de chaque image d'entraînement x_k^{train} . La dissimilitude entre les deux est quantifiée par la distance E_k :

$$E_k = \|z^{test} - z_k^{train}\|_2 \quad (6)$$

En évaluant cette distance pour chaque visage d'entraînement, on peut déterminer le visage d'entraînement x_k^{train} le plus proche du visage test x^{test} . Son identifiant $id_{train}(k)$ permet alors la reconnaissance du visage de test.

1. Calculez la distance entre les visages de test et d'entraînement. La matrice contenant la distance sera de dimension 30×90 , pour $K = 30$ vecteurs caractéristiques conservés.
2. A partir de la distance calculée à la question précédente et du vecteur id_{train} , calculez le vecteur id_{test} (dimension 30×1) donnant l'indice du visage d'entraînement le plus proche de chaque visage de test, pour $K = 30$.
3. Recalculez le vecteur id_{test} pour différentes valeurs de K . Quelle valeur de K permet d'avoir un compromis acceptable entre bonne reconnaissance et temps de calcul faible ?