
BIMA - rapport

RECONNAISSANCE DE VISAGES PAR EIGENFACES

Kim-Anh Laura NGUYEN
Arij RIABI
M1 DAC
Promo 2018-2019

Enseignant : Arnaud DAPOGNY

L'objectif du TME est d'étudier les propriétés de la méthode de reconnaissance de visages « eigenfaces ». Nous développerons un système capable :

- d'identifier un visage parmi une base de données de visages de référence
- de déterminer si une image contient un visage présent dans la base de données
- de reconnaître si une image représente un visage ou non

Nous disposons de la base de données de visages Yale Face Database, sur laquelle nous appliquerons les outils développés. Cette base contient 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 la variation de l'expression du visage et la variation des accessoires.

Chaque visage sera représenté par un vecteur de caractéristiques supposées invariantes pour une même personne, et différentes d'une personne à l'autre. Il sera considéré comme un vecteur dans un espace ayant autant de dimensions que de pixels dans l'image (4096 dans notre cas). Un ensemble de visages sera stocké dans une matrice dont chaque colonne est un visage. En général, les images seront contenues dans une matrice \mathbf{X} de taille $d \times n$:

$$\mathbf{X} = [x_1, \dots, x_n]$$

Cette base \mathbf{X} sera divisée en deux groupes : l'un sera utilisé comme jeu d'entraînement, \mathbf{X}^{train} de taille $d \times N_{train}$, l'autre comme jeu de test, \mathbf{X}^{test} de taille $d \times N_{test}$.

Exercice 1 - Chargement de la base et centrage des visages

Dans cette partie, nous calculons et affichons le "visage moyen" (ou moyenne des visages) de la base de visages de références \mathbf{X}^{train} ainsi que quelques visages et leurs visages centrés associés. Les fonctions utilisées (calcul du visage moyen x_{moy} et centrage des visages) sont contenues dans l'annexe A. Les résultats sont présentés dans la figure 1.

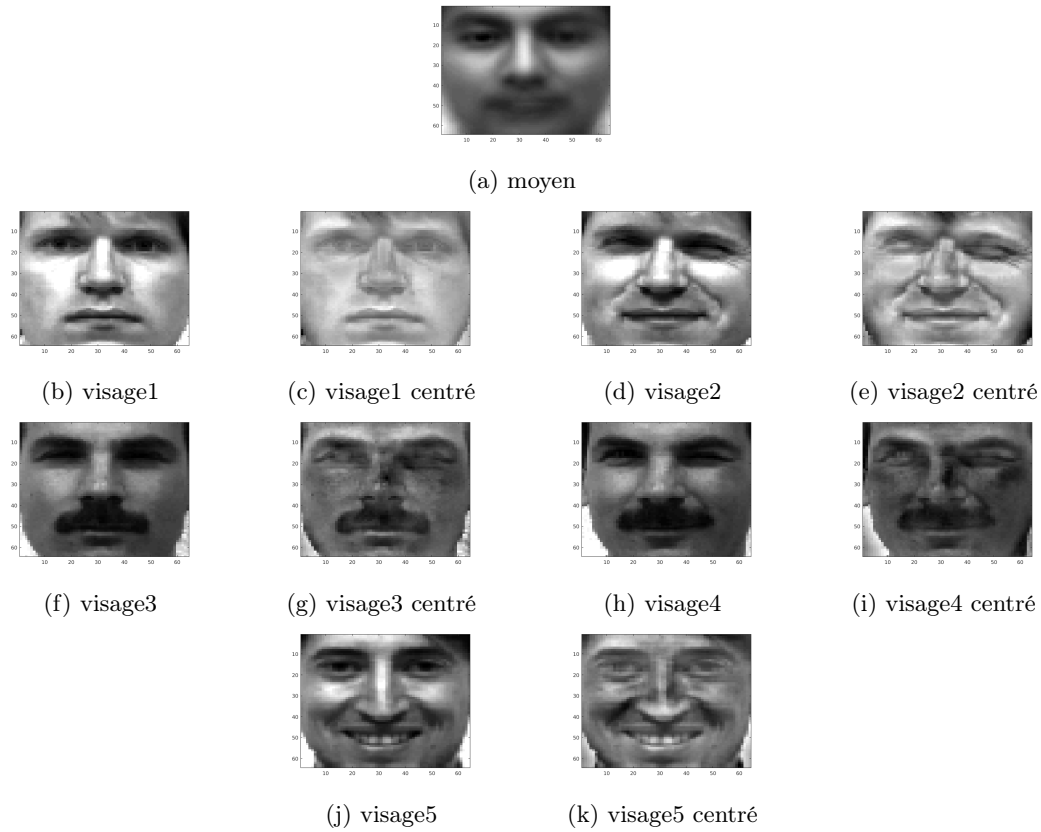


FIGURE 1 – Visage moyen et centrage de la base d'apprentissage

Exercice 2 - Calcul des Eigenfaces (ACP)

Pour résoudre le problème de la reconnaissance de visages, nous allons utiliser la méthode des *eigenfaces*, une approche de type "image" basée sur l'Analyse en Composantes Principales (ACP).

Le principe est d'extraire les caractéristiques d'un visage quelconque $x \in \mathbb{R}^d$ par une méthode mathématique de réduction de dimensionnalité basée sur l'ACP. Les eigenfaces sont définies comme les axes principaux obtenus en effectuant cette analyse. Il s'agit ensuite de modéliser la différence du visage x par rapport au visage moyen x_{moy} par un ensemble limité de K images, $\mathbf{W}_K = [u_1, \dots, u_K]$, appelées eigenfaces. Le visage x est donc exprimé comme le visage moyen auquel s'ajoute une combinaison linéaire d'eigenfaces :

$$x = x_{moy} + \sum_{h=1}^K a_h u_h + \epsilon$$

où a_h représente le poids de l'eigenface d'indice h dans le visage x , et ϵ représente l'erreur entre x et son approximation par les eigenfaces.

À partir de $\mathbf{X}_c = [x_1 - x_{moy}, \dots, x_n - x_{moy}]$, l'ensemble des images centrées de la base d'apprentissage, nous mettons au point la fonction **eigenfaces** (annexe B) qui calcule la matrice des eigenfaces, les vecteurs propres de la matrice de covariance $\mathbf{X}_c \times \mathbf{X}_c^T$, ainsi que leurs valeurs propres associées. Nous normalisons ces dernières afin que leur somme soit égale à 100%.

La figure 2 contient le visage moyen ainsi que les 15 premières eigenfaces et leurs valeurs propres associées.

Les eigenfaces forment un ensemble représentant, avec un nombre de dimensions réduit, les visages utilisés pour construire la matrice de variance co-variance.

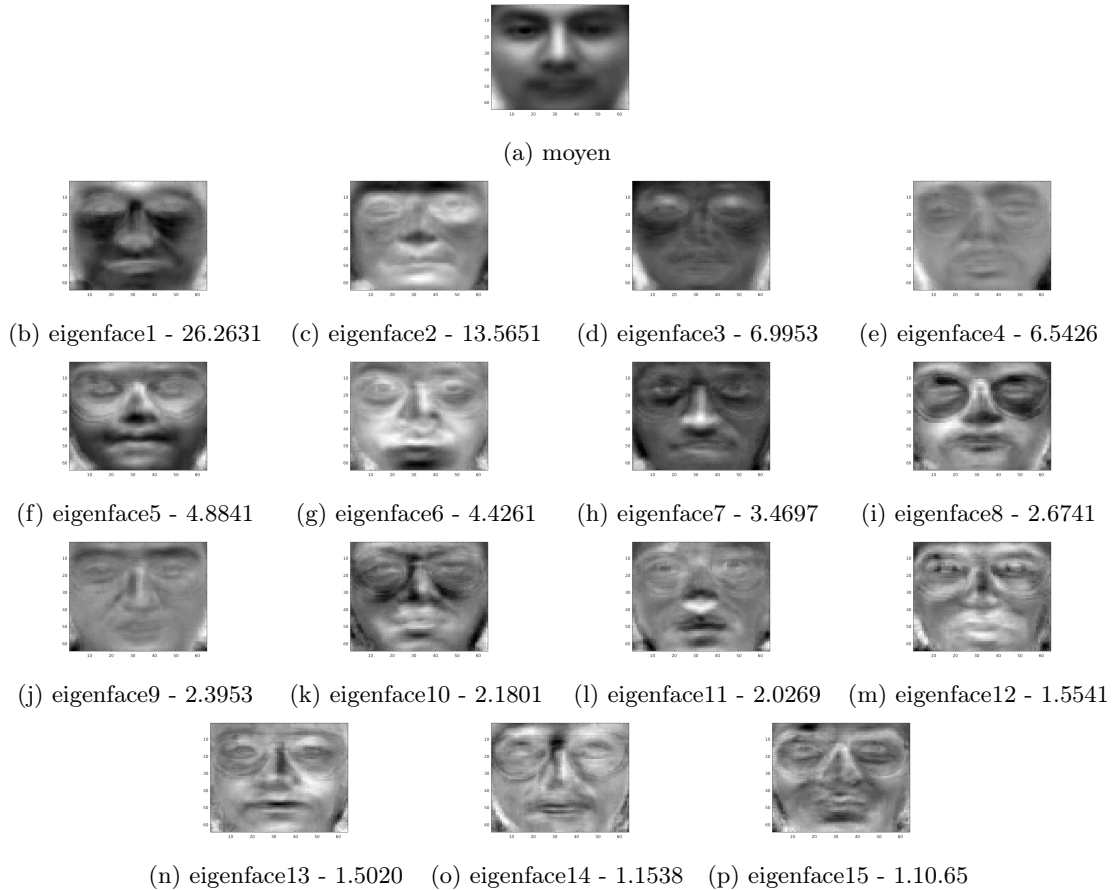


FIGURE 2 – Visage moyen et 15 premières eigenfaces ainsi que leurs valeurs propres associées

Afin d'évaluer le taux de variation capturée par les K premières eigenfaces, nous traçons la courbe de la somme cumulée des valeurs propres normalisées (figure 3).

Nous constatons qu'avec $K = 30$ eigenfaces, nous capturons un taux de variation de 90%, ce qui permet d'avoir une "bonne" reconstruction des visages de test.

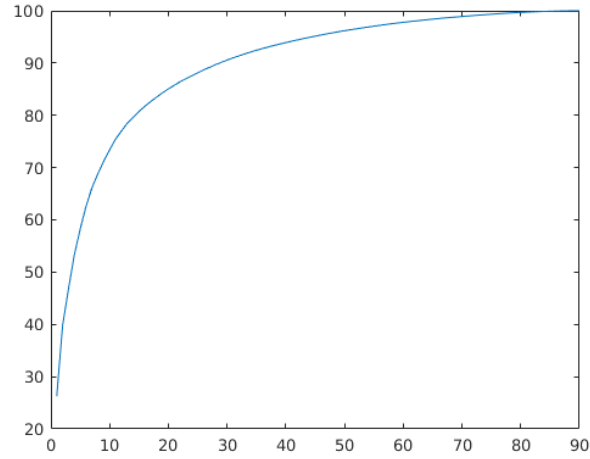


FIGURE 3 – Somme cumulée des valeurs propres

Exercice 3 - Projection dans le sous-espace des visages

Dans la suite, nous utiliserons un nombre réduit d'eigenfaces afin de modéliser l'espace de visages représenté sous la forme de la base $\mathbf{W}_K = [u_1, \dots, u_K]$ des K premiers vecteurs propres.

La projection d'une image x dans le sous-espace des visages consiste à soustraire le visage moyen x_{moy} puis effectuer le produit scalaire de l'image obtenue avec chaque eigenface $a_h, h \in \{1; K\}$. Ceci donne les coordonnées de l'image test dans le sous-espace des visages, qui est de dimension K :

$$z = \mathbf{W}_K^\top \times (x - x_{moy})$$

Son image reconstruite dans l'espace d'origine \mathbb{R}^n est :

$$\tilde{x} = x_{moy} + \sum_h a_h u_h = x_{moy} + \mathbf{W}_K \times z$$

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

$$E^{recons}(x) = \|x - \tilde{x}\|_2 = \sqrt{\sum_p (x(p) - \tilde{x}(p))^2}$$

Nous mettons au point une fonction **calculeProj** qui, à partir d'une image x et de l'image moyenne x_{moy} de \mathbf{X}^{train} , calcule les coordonnées z dans le sous-espace \mathbf{W}_K des visages.

Nous écrivons également une fonction **reconstruction**, qui à partir de la projection z d'une image dans le sous-espace des visages de dimension K , calcule les coordonnées de l'image projetée \tilde{x} dans l'espace \mathbb{R}^n de départ.

Nous implémentons aussi une fonction **erreur_Reconstruction** qui calcule l'erreur de reconstruction entre x et \tilde{x} , et **affiche_Reconstruction**, qui affiche l'image initiale et la reconstruction obtenue pour $K \in \{5, 10, 25, 50, 90\}$.

Ces fonctions sont contenues dans l'annexe C.

Nous utilisons ces fonctions pour projeter et reconstruire plusieurs images de la base de référence et de test. Les figures 4, 5, et 6 montrent les résultats de la reconstruction pour, respectivement, l'image 50, 55 de \mathbf{X}^{train} , et 17 de \mathbf{X}^{test} .

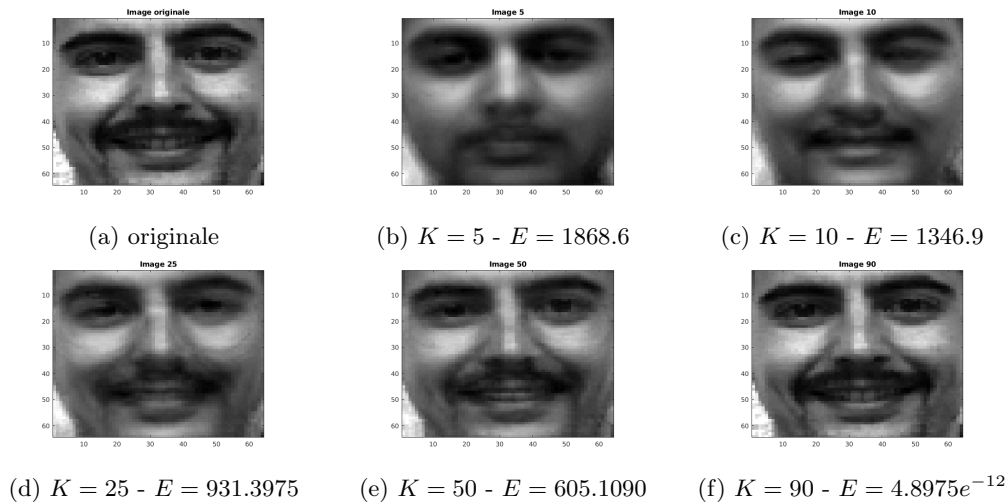


FIGURE 4 – Reconstruction de l'image 50 de la base d'apprentissage pour différentes valeurs de K

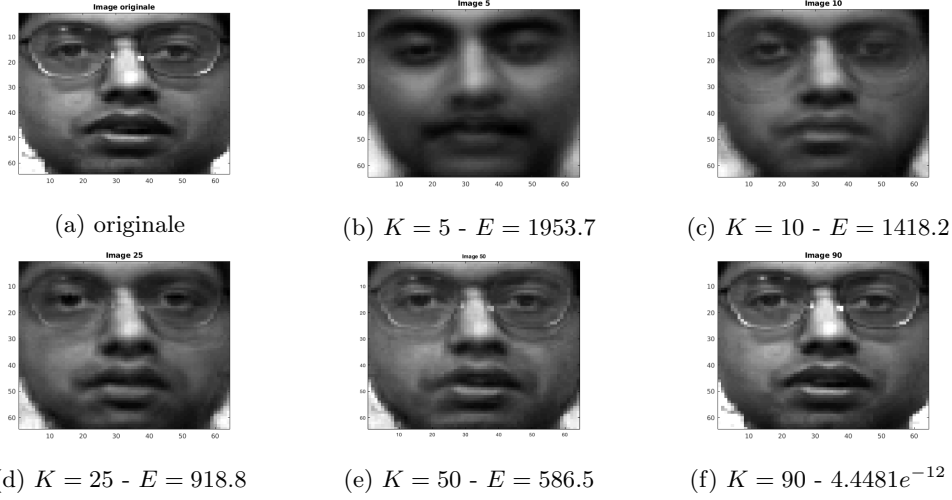


FIGURE 5 – Reconstruction de l'image 55 de la base d'apprentissage pour différentes valeurs de K

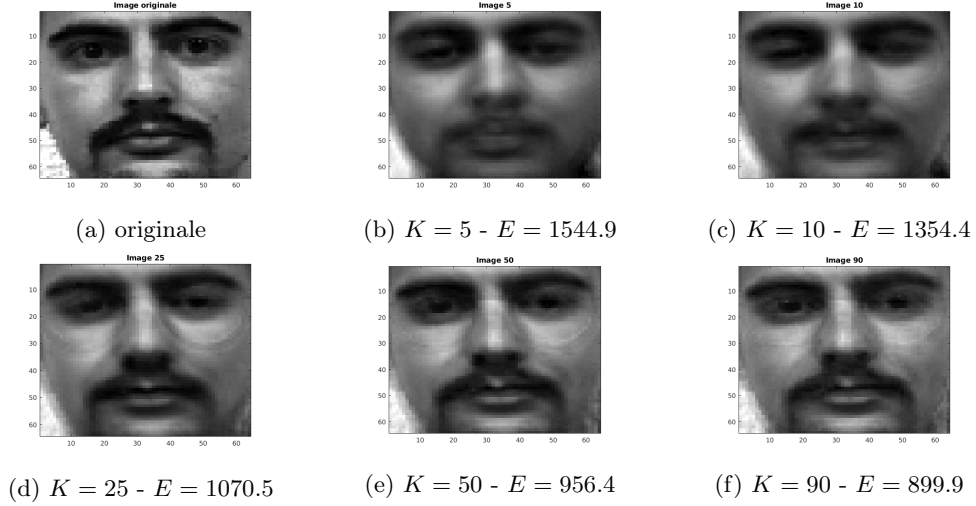


FIGURE 6 – Reconstruction de l'image 17 de la base d'apprentissage pour différentes valeurs de K

Nous constatons sur les figures 4 et 5 que les images 50 et 55 de la base d'apprentissage sont identiques à leur reconstruction. La figure 6 montre cependant que ce n'est pas le cas pour l'image 17 de la base de test : alors que l'erreur de reconstruction est quasiment nulle pour les images de \mathbf{X}^{train} lorsque $K = 90$ ($4.4481e^{-12}$ pour l'image 55), elle est de 899.9 pour l'image de \mathbf{X}^{test} .

Nous en concluons qu'il existe une différence en terme de reconstruction entre les visages issus de la base d'entraînement et ceux issus de la base de test. En effet, le sous-espace des visages dans lequel nous projetons les images est construit à partir de la base d'apprentissage. Les images de la base de test n'entrent, elles, pas en jeu. Les visages de \mathbf{X}^{train} sont donc plus fidèlement reconstruits que ceux de \mathbf{X}^{test} , et ce d'autant plus lorsque K augmente.

Nous traçons également l'évolution de la moyenne de l'erreur de reconstruction des visages de test lorsque K varie de 1 à 90 (le nombre total de eigenfaces). Le résultat est présenté dans la figure 7.

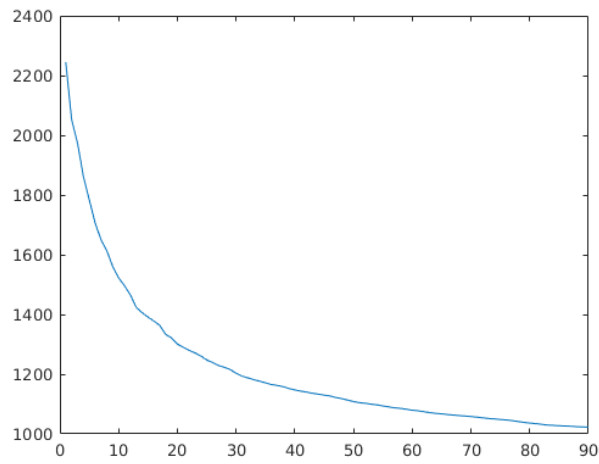


FIGURE 7 – Courbe de la moyenne de l’erreur de reconstruction des visages de X_{test} pour K variant de 1 à 90

Nous pouvons voir que la moyenne de l’erreur de reconstruction décroît exponentiellement lorsque K augmente. Ce résultat est cohérent avec la somme cumulée calculée dans l’exercice 2 (figure 3).

Exercice 4 - Reconnaissance de visage : identification

Le problème de la reconnaissance de visages est défini de la manière suivante : étant donnée une image de visage, on souhaite déterminer l'identité de la personne correspondante. Nous allons comparer le vecteur de caractéristiques du visage à reconnaître avec celui de chacun des visages de la base.

À chaque visage de référence x_k^{train} est associée une identité id_k^{train} . Nous allons identifier un visage x^{test} à partir des visages de \mathbf{X}^{train} .

Soit $x^{test} \in \mathbf{X}^{test}$. La méthode de reconnaissance de visage utilisée est la suivante :

- projection de x^{test} dans la base des eigenfaces $\{a_h\}_{h=1,\dots,K}$

$$z^{test} = \mathbf{W}_K^\top \times (x^{test} - x_{moy})$$

- calcul de la distance entre z^{test} et la projection z_k^{train} de chaque image de référence x_k^{train} dans le sous-espace des visages

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

- détermination du visage de référence x_k^{train} le plus proche du visage x^{test} . Son identifiant $id^{train}(k)$ permet alors la reconnaissance du visage test.

Il est plus intéressant de calculer la distance $E_k(x^{test})$ dans le sous-espace des visages plutôt que dans l'espace de départ car ce dernier est de trop grande dimension (autant de dimensions que de pixels dans l'image). Un classifieur appris sur cet espace est donc trop sensible au bruit. De plus, utiliser la base des eigenfaces diminue la complexité du problème car seules les dimensions les plus pertinentes sont conservées.

Nous mettons au point une fonction `calculMatDist` (annexe D) qui, à partir de l'ensemble \mathbf{X}_c^{train} des images centrées d'apprentissage, de l'ensemble \mathbf{X}_c^{test} des images centrées de test, des eigenfaces \mathbf{W} et de leur nombre conservé K , calcule D , la matrice de distance entre les visages de test et de référence.

À partir de cette matrice, nous écrivons une fonction `identification` (annexe D) qui calcule le vecteur \hat{id}^{test} donnant l'indice du visage de référence le plus proche de chaque visage de test.

Nous calculons maintenant, pour $K = 30$, le taux d'identification en comparant \hat{id}^{test} aux labels id^{test} . Il est de 90%.

Nous faisons ensuite varier K de 1 à 90 (le nombre total de valeurs propres de \mathbf{W}) puis traçons la courbe du ratio de visages reconnus en fonction de K (figure 8).

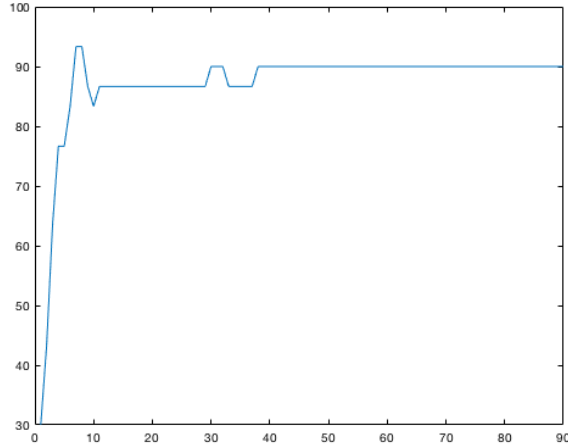


FIGURE 8 – Taux de visages de X_{test} reconnus en fonction de K

La figure 8 présente l'évolution du taux d'identification des visages de \mathbf{X}^{test} en fonction du nombre de dimensions conservées K . Nous observons que, pour $K = 1$ à $K = 40$, le nombre de visages reconnus augmente, puis stagne lorsque K dépasse 30 (l'allure non strictement croissante de la courbe est justifiée par le peu de visages que contient \mathbf{X}^{test}). Cela permet de montrer expérimentalement que, jusqu'à une certaine valeur, plus le nombre de dimensions K utilisées pour modéliser l'espace de visages est grand, plus notre système est performant. Cependant, à partir d'un certain K , le taux d'identification n'augmente plus et reste constant. Cela s'explique par le fait que, les visages tests n'ayant pas servi à l'apprentissage du modèle, ce dernier aura toujours une marge d'erreur. Pour avoir une bonne reconnaissance et un temps de calcul faible, on peut prendre $K = 40$.

Pour $K = 30$, nous calculons pour chaque visage de \mathbf{X}^{train} sa distance dans le sous-espace par rapport à l'ensemble des visages de \mathbf{X}^{train} . Nous visualisons dans la figure 9a le résultat sous la forme d'une image de matrice.

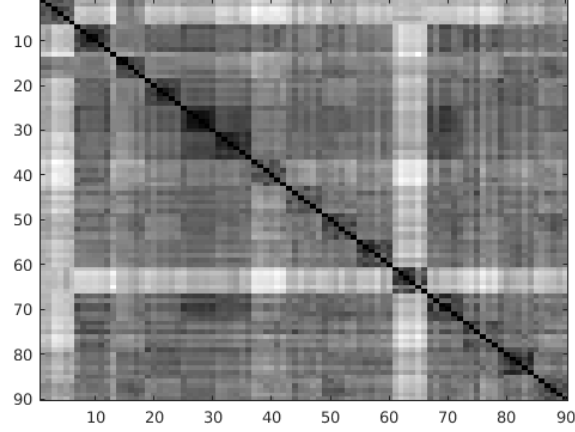
Nous remarquons que la diagonale de cette matrice ne contient que des zéros. En effet, chaque élément de cette diagonale correspond à la distance $\|z - z'\|_2$ entre deux visages x et x' avec $x = x'$. Or, $z = \mathbf{W}_K^\top \times (x - x_{moy}) = \mathbf{W}_K^\top \times (x' - x_{moy}) = z'$. D'où $\|z - z'\|_2 = 0$.

Nous traçons également l'évolution du taux de visages de référence reconnus en fonction de K (figure 9b), ainsi que l'évolution de ce même taux en fonction de K mais sans prendre en compte la diagonale de D , i.e., les distances entre images identiques (figure 9c).

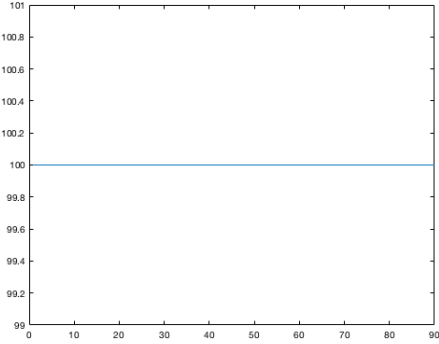
Nous constatons que le premier taux d'identification est de 100% pour toute valeur de K , mais que le second varie de 30% pour $K = 1$ à plus de 90% pour $K = 90$. En effet, en conservant cette diagonale, chaque visage de \mathbf{X}^{train} a une distance minimale avec elle-même et est donc affectée à son propre identifiant, d'où un taux d'identification de 100%. En supprimant cette diagonale, on ne considère plus les distances entre images identiques et le taux d'identification est donc plus faible (mais toujours légèrement meilleur que pour les images de test).

| | distance minimale | distance maximale |
|---------------------|-------------------|-------------------|
| même classe | 270.8728 | 4426.7246 |
| classes différentes | 1264.6149 | 6730.6295 |

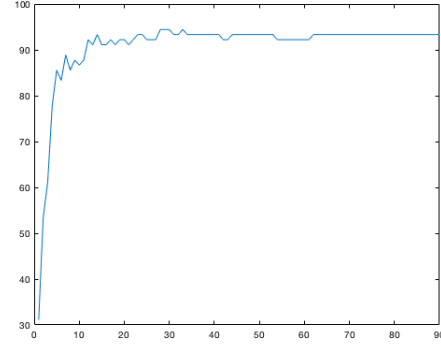
TABLE 1 – Distances min et max entre deux visages de la même classe et de classes différentes



(a) Matrice des distances entre chaque image de \mathbf{X}^{train} dans le sous-espace des visages



(b) Taux de visages de X_{train} reconnus en fonction de K



(c) Taux de visages de X_{train} reconnus en fonction de K , sans tenir compte de la diagonale de D

Nous calculons également les distances minimales et maximales entre deux visages appartenant à la même classe (même personne), et deux visages de classes différentes. Les résultats sont présentés dans le tableau 1. Nous constatons que les distances minimale et maximale entre deux images correspondant au même visage sont plus petites que celles entre deux visages de personnes différentes : notre modèle tient donc de la similitude entre images de même classe.

Nous pouvons mettre en place un seuil θ pour détecter la présence d'un visage inconnu : si la distance minimale calculée entre un visage x et un visage de référence est supérieure à la distance maximale entre deux visages de classe différentes, alors x est un visage inconnu.

Exercice 5 - Classification visage/non-visage

Dans cette partie, nous utiliserons l'erreur de reconstruction pour vérifier qu'une image est bien une image de visage. Un "non-visage" sera le terme employé pour désigner une image contenant autre chose qu'un visage. Les non-visages sont contenus dans la matrice \mathbf{X}^{noface} .

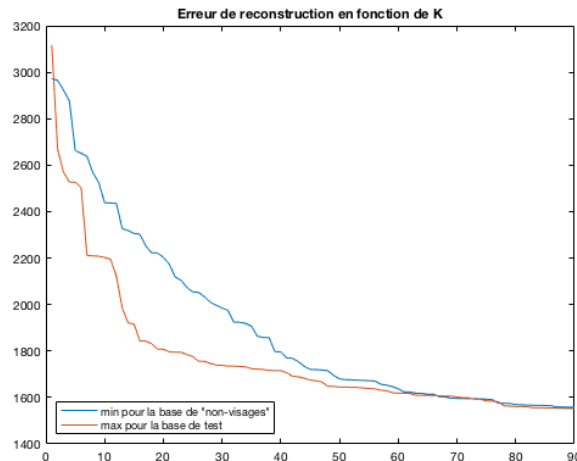
Nous calculons l'erreur de reconstruction des images de \mathbf{X}^{noface} et \mathbf{X}^{test} en fonction de K . Dans les deux cas, nous calculons la moyenne de ces erreurs, ainsi que l'erreur minimum pour les non-visages, et l'erreur maximum pour les visages. Les courbes associées sont présentées dans la figure 10.

Nous constatons sur la figure 10a que les deux courbes d'erreur décroissent de la même manière en fonction de K . Cependant, l'erreur moyenne de reconstruction E_{noface} des images de non-visages est significativement supérieure à l'erreur moyenne E_{test} de reconstruction des visages ($E_{noface} - E_{test} \approx 3000$).

Sur la figure 10b, nous observons que l'erreur de reconstruction maximale pour une image de test est, jusqu'à $K = 60$, toujours supérieure à l'erreur de reconstruction minimale pour une image de non-visage.



(a) Erreur de reconstruction moyenne



(b) Erreur de reconstruction minimale pour les non-visages et maximale pour les visages de test

FIGURE 10 – Erreur de reconstruction des images de la base de non-visages et des images de visages test en fonction de K

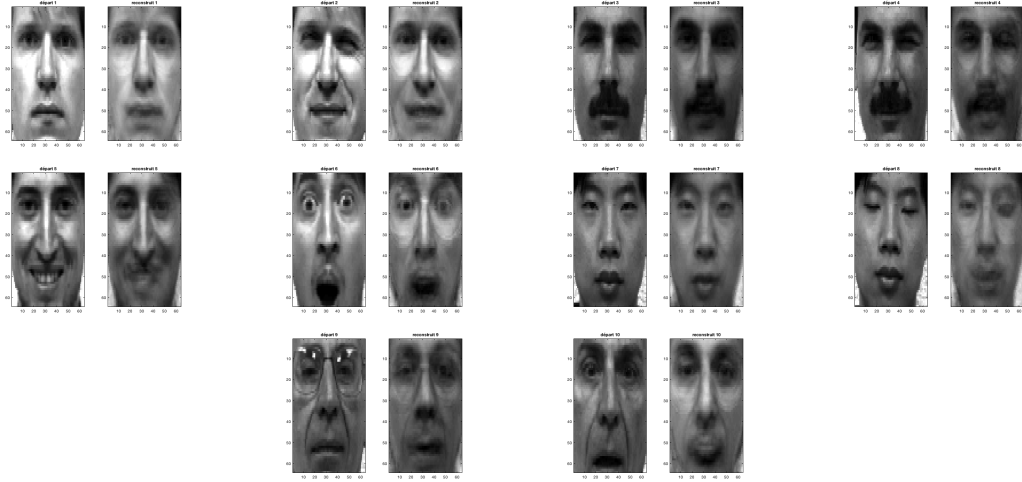


FIGURE 11 – Reconstruction de 10 images de la base \mathbf{X}^{test}

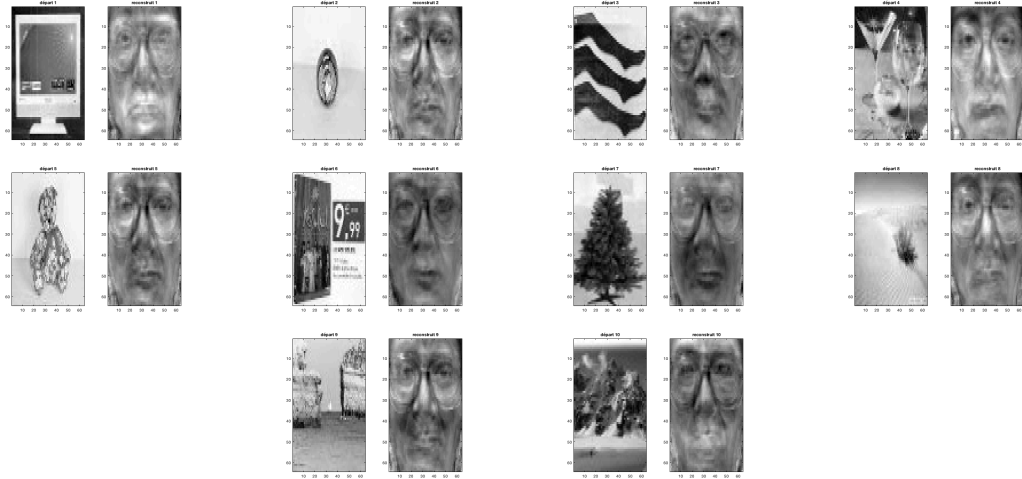


FIGURE 12 – Reconstruction de 10 images de la base \mathbf{X}^{noface}

Nous constatons sur la figure 12 que les images reconstruites n'ont aucune cohérence avec les images de départ. Notre modèle essaye de reconstruire des visages alors que les images passées en entrées n'en sont pas. Ce problème est dû au fait que la technique que nous utilisons se base sur la projection d'images dans un sous-espace de visages, et qu'elle ne peut donc être utilisée que pour la reconnaissance de visages.

A Exercice 1 - Fonctions

- calculMoy

```
1 function [Imoy]=calculMoy(I)
2     Imoy=mean(I,2);
3 end
```

- calculCen

```
1 function [B_c]=CalculCen(B_I,moy)
2     B_c=B_I-moy;
3 end
```

B Exercice 2 - Fonctions

- eigenfaces

```
1 function [U, lambdas]=eigenfaces(Xc)
2 [U,S,V] = svd(Xc,0);
3 lambdas = diag(S).^2;
4 end
```

C Exercice 3 - Fonctions

- calculeProj

```
1 function [z] = calculeProj(x,x_moy,K,W)
2     z = W(:,1:K)' * (x - x_moy);
3 end
```

- reconstruction

```
1 function [x_r] = reconstruction(z,x_moy,K,W)
2     x_r = x_moy + W(:,1:K) * z;
3 end
```

- erreur_Reconstruction

```
1 function [Er] = erreur_Reconstruction(x_r,x)
2     Er = sqrt(sum((x_r - x).^2));
3 end
```

- affiche_Reconstruction

```
1 function [Er] = affiche_Reconstruction(x, x_moy, U)
2     figure(1);
3     colormap(gray);
4     imagesc(reshape(x,[64,64]));
5
6     title("Image originale");
7     K_ = [5, 10, 25, 50, 90];
8     Er = zeros(1,5);
9     for i=1:5
10         k = K_(i);
11         z = calculeProj(x, x_moy, k, U);
12         x_r = reconstruction(z, x_moy, k, U);
13         Er(1,i) = erreur_Reconstruction(x_r, x);
```

```

14         figure(k);
15         colormap(gray);
16         imagesc(reshape(x_r,[64,64]));
17         title(strcat("Image ", int2str(k)));
18     end
19 end

```

D Exercice 4 - Fonctions

- calculMatDist

```

1 function [D] = calculMatDist(Xc_train, Xc_test, W, K)
2     [p, n] = size(Xc_train);
3     [q, m] = size(Xc_test);
4     z_train = W(:,1:K)' * Xc_train;
5     z_test  = W(:,1:K)' * Xc_test;
6     D = zeros(n, m);
7     for i=1:n
8         for j=1:m
9             t = (z_train(:, i) - z_test(:, j)) .^2;
10            D(i, j) = sqrt(sum(t));
11        end
12    end
13    D = D';
14 end

```

- identification

```

1 function [ id_test_hat ] = identification( D )
2     [p, id_test_hat] = min(D');
3 end

```