



**Rapport Projet Calcul Scientifique et Analyse de
Données
TP3 - Application de l'ACP : les « Eigenfaces »**

Faical TOUBALI HADAoui
Younes EL BOUZEKRAoui

Département Sciences du Numérique - Première année
2019-2020

Table des matières

1	Introduction	3
2	Positionnement du problème :	3
3	Bilan d'exercice 1 : analyse en composantes principales :	3
4	Bilan d'exercice 2 : Projection des images sur les eigenfaces :	4
4.1	Reconstruction des images d'apprentissages	4
4.2	Remarques :	7
4.3	Calcul du RMSE	7
4.4	Remarques :	7
5	Bilan d'exercice 3 : Application à la reconnaissance faciale :	8
6	Question 5 :	10
7	Question 6 :	11
8	Question 7 :	11

Table des figures

1	n = 16 Images d'apprentissages	3
2	Les « Eigenfaces »	4
3	Le calcul des X reconstruit à partir des q premières eigenfaces et q premières composantes principales	4
4	X reconstruit à partir du premier eigenface et de la première composante principale	5
5	Le calcul des X reconstruit à partir des 4 premières eigenfaces et 4 premières composantes principales	5
6	Le calcul des X reconstruit à partir des 10 premières eigenfaces et 10 premières composantes principales	6
7	Le calcul des X reconstruit à partir des 15 premières eigenfaces et 15 premières composantes principales	6
8	Graphe de l'évolution du RMSE entre les images originales et les images reconstruites	7
9	l'individu 40 en posture 6 (37 dans la base) appartient à la base d'apprentissage .	8
10	l'individu 40 en posture 5 (37 dans la base) appartient à la base d'apprentissage .	9
11	l'individu 39 n'appartient pas à la base d'apprentissage	10
12	l'individu 34 n'appartient pas à la base d'apprentissage	10
13	Taux d'erreur pour un classifieur 1ppv	11
14	Taux d'erreur pour un classifieur 3ppv	11
15	Taux d'erreur pour un classifieur 3ppv avec 8 compsanes principales	11
16	Taux d'erreur pour un classifieur 3ppv avec 12 compsanes principales	11

1 Introduction

L'objectif de ce projet est d'utiliser les eigenfaces pour la reconnaissance faciale utilisée de nos jours dans plusieurs domaines. En effet, en s'appuyant sur la méthode de l'ACP, nous pouvons à travers du calcul des composantes principales savoir si une image d'une personne donnée en une posture donnée appartient à une base d'images d'apprentissages.

Pour ce faire, le sujet propose de passer par deux étapes essentielles pour arriver finalement à l'application de la reconnaissance faciale.

2 Positionnement du problème :

Nous disposons d'une base de données d'images de 37 personnes sous 6 postures différentes, ces images sont appelées images d'apprentissage. Une image est représentable par une matrice dont le nombre des colonnes et le nombre des lignes correspondent respectivement à la largeur et la hauteur de l'image. Les images que nous avons sont de dimension 480 x 640 (Hauteur x Largeur). On se propose de stocker n images d'apprentissage après les avoir vectoriser sous forme des vecteurs colonnes dans une matrice X qui sera d'une dimension de n x p avec n est le nombre d'images comme indiqué et p correspond à $480 \times 640 = 307200$.

3 Bilan d'exercice 1 : analyse en composantes principales :

Le sujet propose d'utiliser $n = 16$ images d'apprentissages correspondant à 4 personnes dans 4 postures différentes. En lançant le script données.mat, une matrice X contenant les transposés des vecteurs colonnes modélisant les images est créée. Ces n images représentent alors une base d'apprentissages.

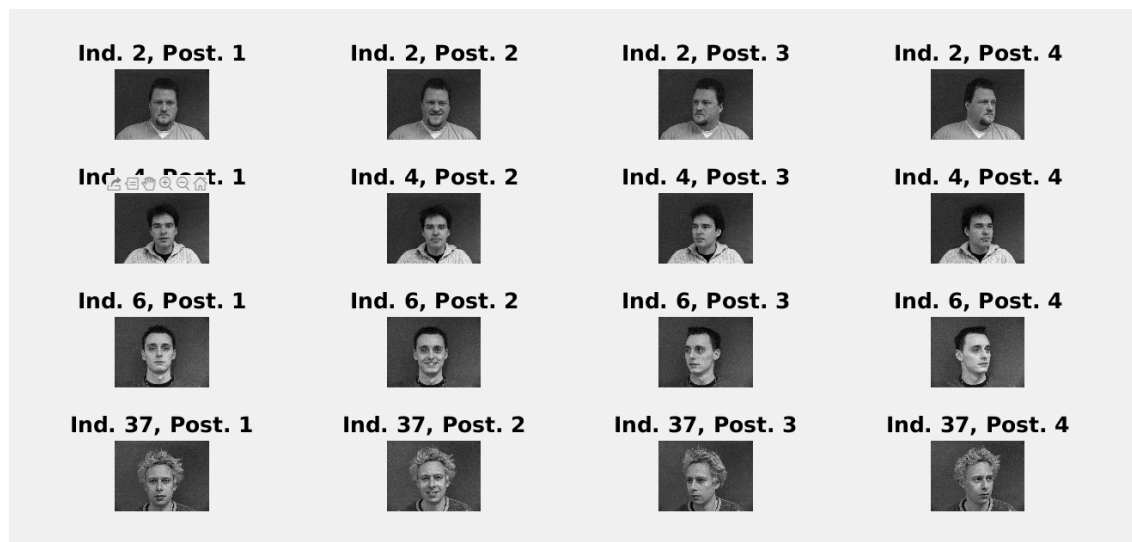


FIGURE 1 – $n = 16$ Images d'apprentissages

Le rôle de l'exercice 1 est d'obtenir les axes principaux des images d'apprentissages en calculant les $n-1$ vecteurs propres de la matrice de variance/covariance Σ_1 . Je note ici qu'il existe une différence dans le calcul de la matrice de variance/covariance par rapport aux TP vus précédemment, cette différence provient de la complexité spatiale et temporelle du calcul du Σ_1 (voir Annexe1) que la fonction eig ne peut supporter, d'où l'utilisation de Σ_2 dont la dimension

est plus petite, cette utilisation découle de la possibilité de tirer les éléments propres de Σ_1 à partir de ceux de Σ_2 . Cette différence est clairement expliquée dans l'Annexe 1 du sujet.

Les vecteurs propres de la matrice de variance/covariance Σ_1 représentent les «Eigenfaces».

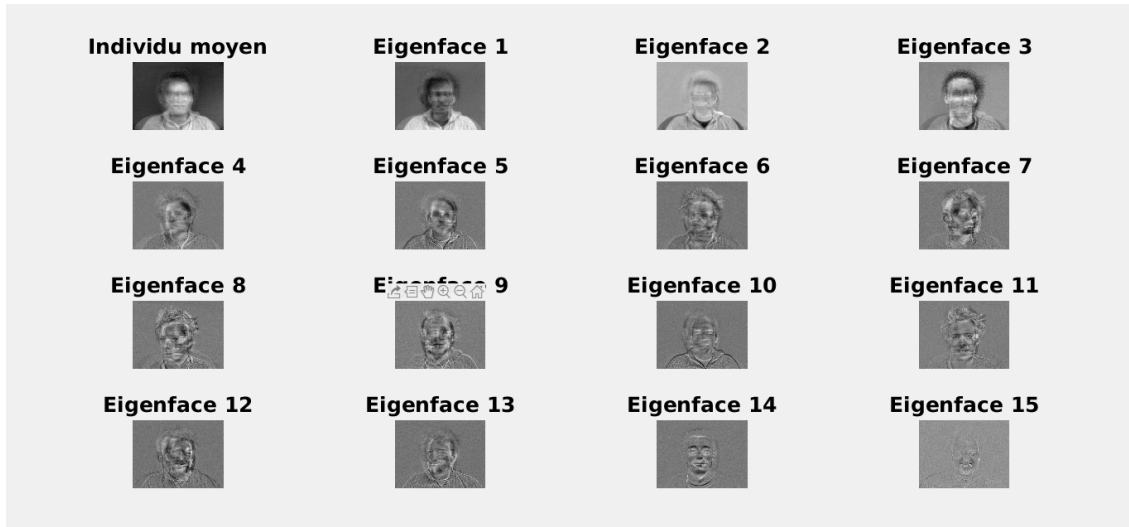


FIGURE 2 – Les « Eigenfaces »

4 Bilan d'exercice 2 : Projection des images sur les eigenfaces :

4.1 Reconstruction des images d'apprentissages

Après avoir calculer les composantes principales en exercice 2 par produit matriciel entre le X centre et les matrices des vecteurs propres (matrice d'échantillon), on s'intéresse à la reconstruction des images d'apprentissages à partir de q premières eigenfaces et de q premières composantes principales avec q allant de 0 jusqu'à $n - 1$.

Explications :

Un simple raisonnement mathématique nous mène à calculer le X reconstruit en utilisant la meme relation utilisée pour calculer les composantes principales mais cette fois-ci dans le sens inverse et en utilisant juste les q premières eigenfaces et composantes principales, à savoir, en calculant le produit matriciel entre la matrice d'échantillon C et la transposée de la matrice contenant les vecteurs propres sans oublier d'ajouter l'individu moyen. Le calcul se fait comme suit :

```
% Composantes principales des donnees d'apprentissage
C = X_centre * W;

for q = 1:n-1
    C_q = C(:,1:q);    % q premieres composantes principales
    W_q = W(:,1:q);    % q premieres eigenfaces
    X_reconstruit = C_q * W_q' + individu_moyen ;
```

FIGURE 3 – Le calcul des X reconstruit à partir des q premières eigenfaces et q premières composantes principales

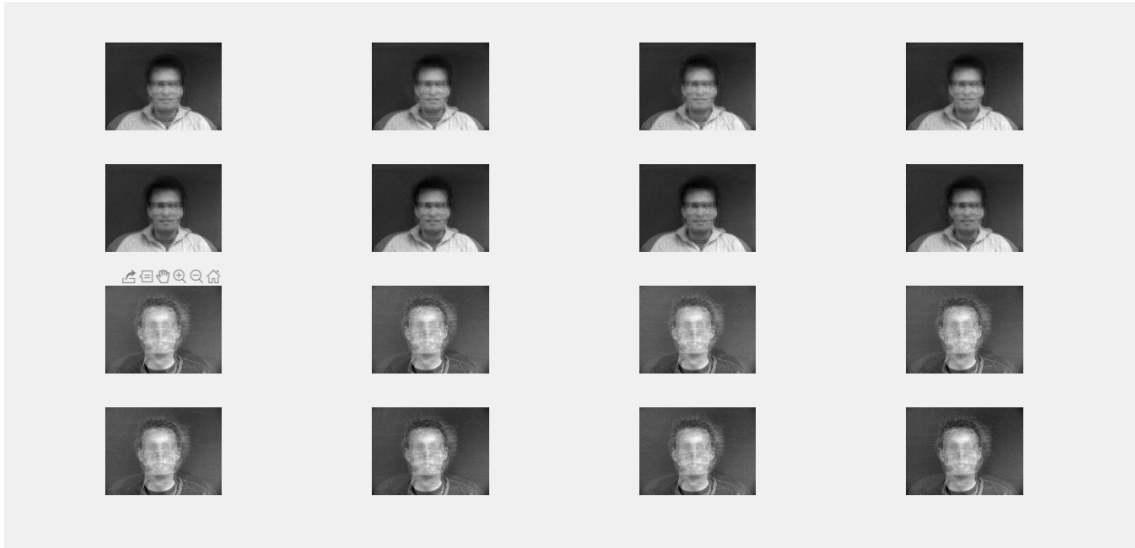


FIGURE 4 – X reconstruit à partir du premier eigenface et de la première composante principale

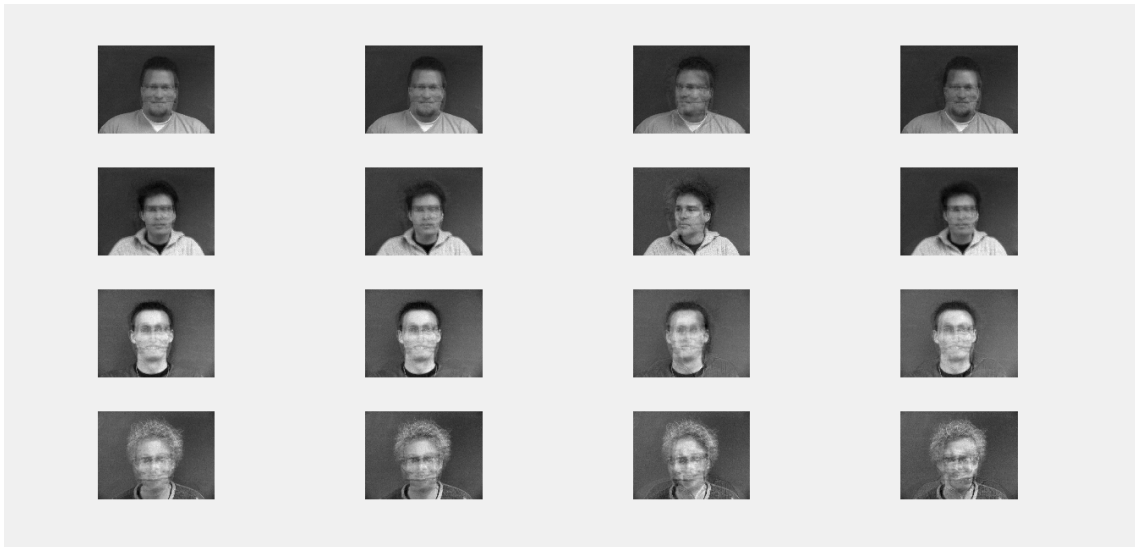


FIGURE 5 – Le calcul des X reconstruit à partir des 4 premières eigenfaces et 4 premières composantes principales

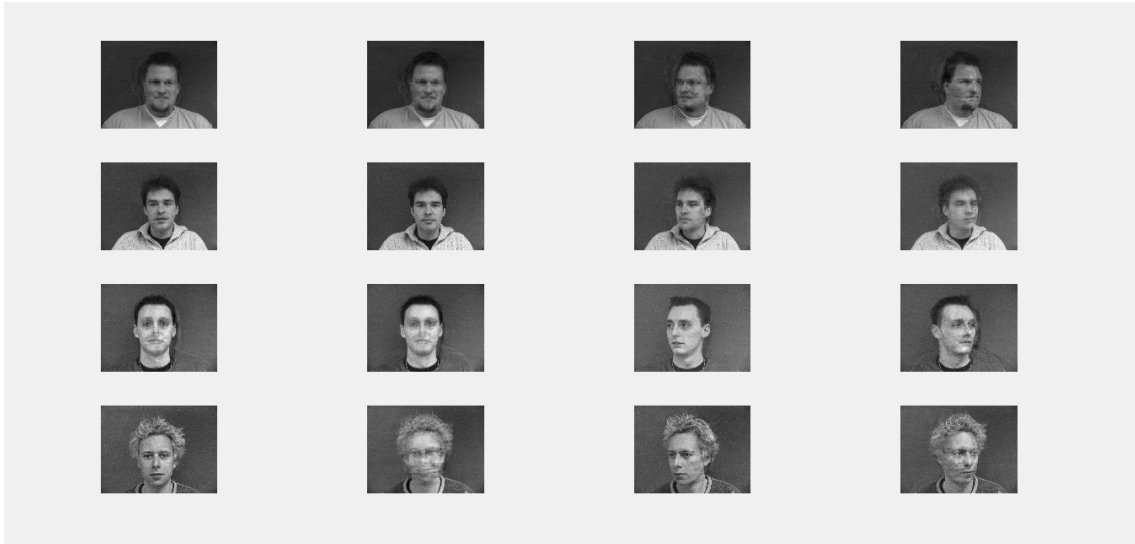


FIGURE 6 – Le calcul des X reconstruit à partir des 10 premières eigenfaces et 10 premières composantes principales

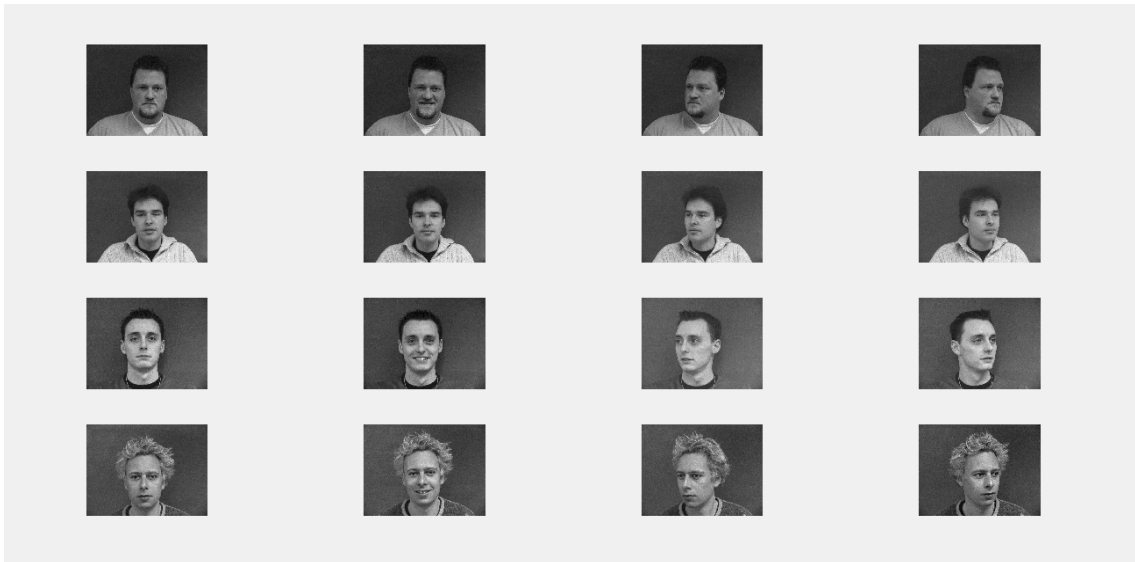


FIGURE 7 – Le calcul des X reconstruit à partir des 15 premières eigenfaces et 15 premières composantes principales

4.2 Remarques :

On constate une évolution progressive dans la reconstitution des images d'apprentissages (les images deviennent de plus en plus claires) ce qui est normale et attendue, car on sait que les axes principaux reflètent par ordre décroissant (tri décroissant des vecteurs propres de la matrice de variance/covariance) le taux d'information que contiennent les images. En d'autres mots, plus on utilise d'eigenfaces et de composantes principales plus on dispose d'informations sur l'image et donc plus l'image s'affiche plus clairement. En arrivant à $q = n - 1$ nous aurons évidemment des images d'apprentissages totalement reconstruits.

Dans chaque itération pour calculer le X reconstruit, nous calculons aussi le RMSE (Root Mean Square Error), cette quantité représente la racine carré de l'erreur quadratique moyenne résidant entre les images originales et les images reconstruites.

4.3 Calcul du RMSE

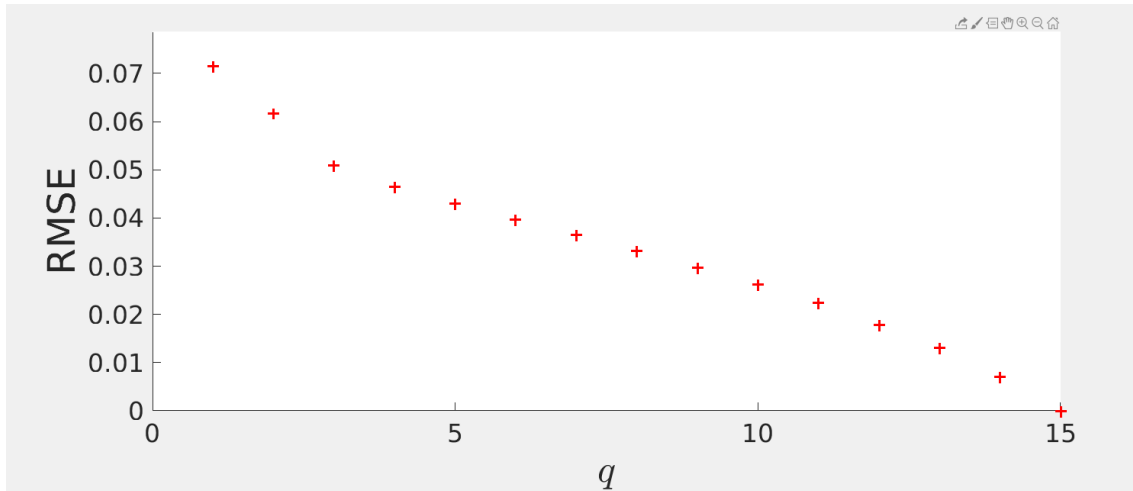


FIGURE 8 – Graphe de l'évolution du RMSE entre les images originales et les images reconstruites

4.4 Remarques :

On constate une évolution décroissante de la racine carré de l'erreur quadratique moyenne résidant entre les images originales et les images reconstruites ce qui est normale et attendue puisque plus on utilise d'eigenfaces et de composantes principales plus on dispose d'informations sur l'image et donc plus le RMSE devient plus petit. En utilisant toutes les eigenfaces et composantes principales, on arrive à un RMSE nul qui signifie une reconstruction totale des images originales.

5 Bilan d'exercice 3 : Application à la reconnaissance faciale :

On cherche maintenant à savoir si un individu donnée appartient à notre base d'apprentissage quelque soit sa posture. Pour ceci, on tire aléatoirement une image de test correspondant à un individu donnée (entre 1 et 37) dans une posture quelconque (entre 1 et 6).

On calcule ensuite la distance entre les $N = 8$ premières composantes principales de l'image de test et des images d'apprentissages.. à partir d'un seuil que l'on doit déterminer, on pourra savoir certainement si l'individu appartient à notre base. Mathématiquement, l'image de test et les images d'apprentissages sont représentés par des point dans un espace de dimension N , les points qui sont plus proche en considérant un seuil représentent à priori le meme individu. Le cas de $N = 2$ est montré par le script clusters.m

Après avoir implanté correctement l'algorithme, il nous ne reste qu'à déterminer le meilleur seuil à partir duquel le voisin le plus proche serait surement le meme indivu de test et ceci dans n'importe quelle posture. Pour ce faire, on prend des individus qui sont dans notre base d'apprentissages en n'importe quelle posture, si le programme affirme qu'il ne connait pas l'individu, on augmente de manière adéquate le seuil. On reprenant cette tache, nous parviendrons à un meilleur seuil. en l'occurence, nous avons observé que le meilleur seuil que nous pouvons avoir c'est $s = 21.5$. à partir de ce seuil notre algorithme nous apparait très précis et n'a commis aucune erreur à la reconnaissance facile comme le montre les figures suivantes :



FIGURE 9 – l'individu 40 en posture 6 (37 dans la base) appartient à la base d'apprentissage



FIGURE 10 – l'individu 40 en posture 5 (37 dans la base) appartient à la base d'apprentissage

Remarquons ici que ces deux images de cet individu correspondent aux deux postures 5 et 6 que la base d'apprentissage ne dispose pas et malgré ceci le programme l'a détecté et c'est exactement le but de la reconnaissance faciale. En effet, il suffit que l'on ait quelques postures de l'individu dans notre base pour détecter cet même individu dans n'importe quel posture.

Dans le cas où l'individu n'appartient pas à notre base d'apprentissage, le programme ne le reconnaît effectivement pas.

Posture numero 3 de l'individu numero 39 Je ne reconnais pas cet individu !



FIGURE 11 – l'individu 39 n'appartient pas à la base d'apprentissage

Posture numero 6 de l'individu numero 34 Je ne reconnais pas cet individu !



FIGURE 12 – l'individu 34 n'appartient pas à la base d'apprentissage

6 Question 5 :

Après l'implémentation d'un classifieur de type kppv, On a créé une nouvelle base d'apprentissage qui contient 3 postures pour

chaque individu (37 individu).

Ensuite, On a tester le classifieur pour un ensemble de Test contenant 37 individu dont la posture est tiré aléatoirement, ainsi les images testés n'appartiennent pas forcément a la base d'apprentissage, ceci est fait exprès pour tester la performance.

A partir des résultats du classifieur on Construit la matrice de Confusion puis on calcule le taux d'erreur qui va nous donner une idée sur la performance du classifieur et de la reconnaissance facial.

On teste le classifieur pour differentes valeurs de k :

Taux_erreur = 0.2973

FIGURE 13 – Taux d'erreur pour un classifieur 1ppv

Taux_erreur = 0.27027

FIGURE 14 – Taux d'erreur pour un classifieur 3ppv

on remarque que plus k est grand plus le classifieur est de bonne qualité ceci revient au fait que chaque individu dans la base d'apprentissage qu'on a construit possède 3 postures différentes alors il est preferable de comparer l'image Test avec l'ensemble des postures (3ppv) au lieu de la comparer avec une seul posture qui est la plus proche (1ppv).

? On peut aussi augmenter la performance du classifieur et de la reconnaissance facial en au augmentant le nombre N de composante principales et donc avoir un bon pourcentage d'information. On compare le taux d'erreur du classifieur 3ppv pour deux valeurs de N (nombre de composantes principales) :

Taux_erreur = 0.27027

FIGURE 15 – Taux d'erreur pour un classifieur 3ppv avec 8 compsanetes principales

Taux_erreur = 0.21622

FIGURE 16 – Taux d'erreur pour un classifieur 3ppv avec 12 compsanetes principales

On remarque que pour 12 composante principale le classifieur est plus performant.

7 Question 6 :

Compte tenu de la quantité de données fournie on remarque que la taille des images est beaucoup plus grande que le nombre des images $307200 \gg 37 \times 6$.

Alors il est préférable de passer par la méthodes de sigma2 de taille beaucoup mois que celle de la matrice sigma1 307200×307200 pour le calcul des couples propres.

8 Question 7 :

Compte tenu de la quantité de données fournies puisque la matrice σ_2 a laquelle on veut calculer les couples propre a une taille de l'ordre de 100.

Et d'après la conclusion de TP2 qui traite les différentes méthodes de calcul des couples propres pour une matrice de taille de l'ordre de grandeur de 100, on peut déduire que la méthode Subspace Iter v3 sera la plus performante dans ce cas.