



Compte-rendu TP Matlab

La reconnaissance de visages

Bluem Juliette, 3A 2i



**UNIVERSITÉ
DE LORRAINE**

LORRAINE INP
vos talents se lèvent à l'Est



Sommaire

I - Introduction.....	3
II - Chargement de la base et centrage des visages	3
III - Calcul des vecteurs caractéristiques.....	4
IV - Projection dans le sous-espace des visages	5
V - Conclusions	6



I - Introduction

Dans ce TP, nous disposons d'une base de donnée (une matrice) contenant des photos de visages. On a 15 individus représentés avec 8 images chacun. Un visage est une matrice de 64×64 . Nous allons étudier le principe de reconnaissance de visages au travers plusieurs étapes clés. Nous commencerons par le centrage de visages. Nous verrons ensuite le calcul de vecteurs caractéristiques. Puis, la projection dans le sous-espace des visages. Pour finir avec leur identification.

II - Chargement de la base et centrage des visages

Dans cette partie, nous commençons par importer le fichier de base de données à l'aide de la fonction *load*. Puis, nous calculons le visage moyen de la matrice grâce à la fonction *mean*. Nous centrons les visages avec une simple soustraction. Enfin, nous créons une fonction permettant la recombinaison de l'image visage.

Notre fonction s'appelle *convertir*, et prend deux paramètres en entrée : une matrice *X* et un nombre *c*. Nous commençons par créer un compteur *j*, et une matrice carrée vide, *m*, de taille 64. Puis, à l'aide d'une boucle *for*, nous allons associer une valeur de *X* par rapport à *c*, à un « emplacement » de *m* (*c* étant l'image que l'on veut recomposer). Enfin, nous affichons notre image, avec la fonction *imshow* étudiée lors du précédent TP.

Voici ce que nous obtenons avec en entrée, la matrice *X_train* (le nombre *c* est implémenté à l'aide de boucle, dans notre cas, nous affichons les 9 premières images de la matrice) :



Figure 1 : 9 premières images de la matrice *X_train*

Et là, le cas où la matrice d'entrée est la matrice *X_train* centrée :



Figure 2 : 9 premières images de la matrice X_{train} centrée

Nous voyons que les images centrées sont plus uniformes que les images initiales. (elles sont plus « plates »).

III - Calcul des vecteurs caractéristiques

Nous divisons cette partie en plusieurs étapes. La première est le calcul d'une matrice U contenant les vecteur caractéristiques recherchés et sa valeur propre associé. Pour cela, nous appelons la fonction *svd*. Ensuite nous calculons le pourcentage associé à chaque valeur propre par rapport à la somme des valeurs propres. Nous mettons donc le carré des valeurs les valeurs de la diagonale de S^* dans la matrice P . On calcule leur pourcentage associé. Pour finir, nous traçons la courbe de la somme cumulée des valeurs propres en pourcentage à l'aide des fonctions *cumsum* et *plot*.

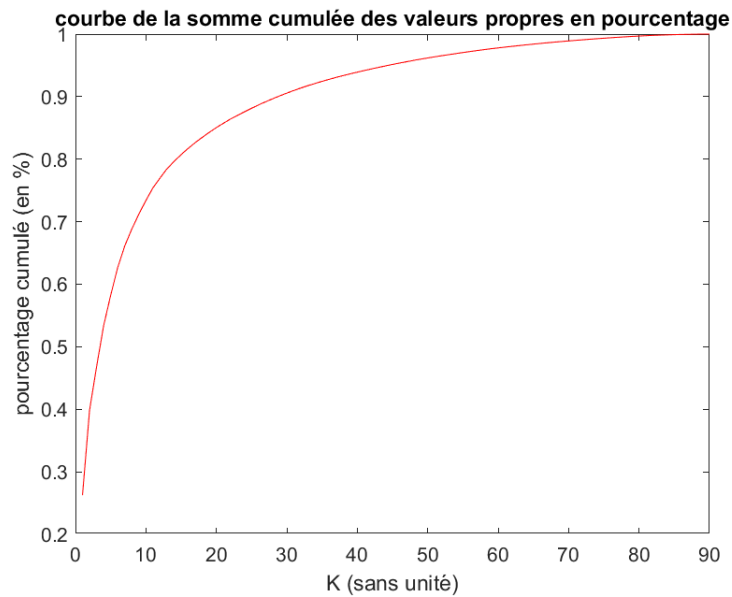


Figure 3 : Courbe de la somme cumulée des valeurs propres en pourcentage

D'après la courbe obtenue, nous voyons que si $K=15$, 80% des valeurs propres sont cumulées. Et, pour $K \geq 11$, nous obtenons une bonne reconstitution.

* : S matrice de valeurs singulières

IV - Projection dans le sous-espace des visages

Dans cette partie, nous allons projeter et comparer avec les vecteurs caractéristiques nos images ; en commençant par calculer les coordonnées de l'image projetée z . Puis nous reconstruisons l'image de départ. Pour cela, nous créons une fonction *reconstruction* qui prend une image de la matrice, calcul son projeté, et applique la formule de reconstruction. Puis, nous voulons calculer l'erreur de reconstruction entre l'image obtenue et l'image initiale, mais je n'ai malheureusement pas réussi. Enfin, nous affichons les deux images pour différentes valeurs de K , et différentes images de base. J'ai choisi d'utiliser une boucle *for* et mes différentes fonctions.

Voici le résultat pour $K=2$:

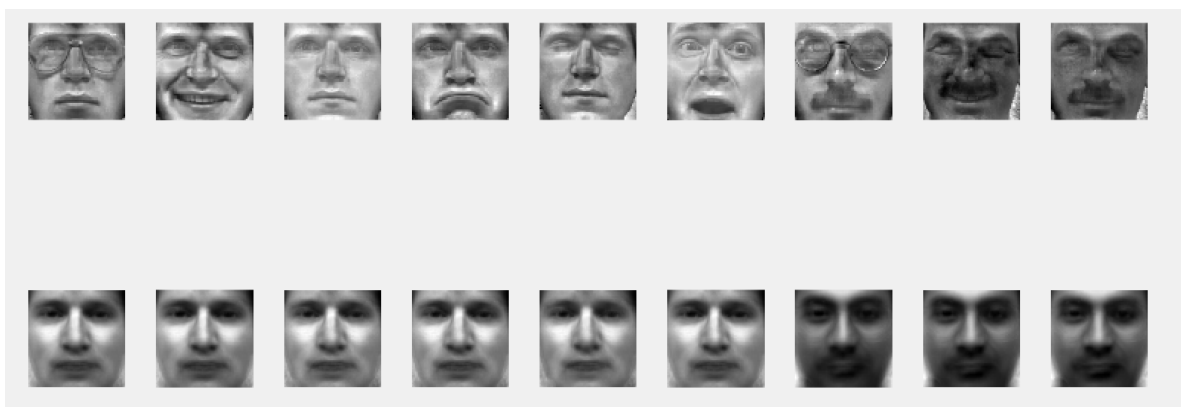


Figure 4 : Reconstruction d'images avec $K=2$



Avec $K=2$, le logiciel fait beaucoup d'erreur de reconstruction, dans le sens où nous ne voyons que deux visages différents pour résultats, alors qu'il y en a bien plus.

Pour $K=11$:

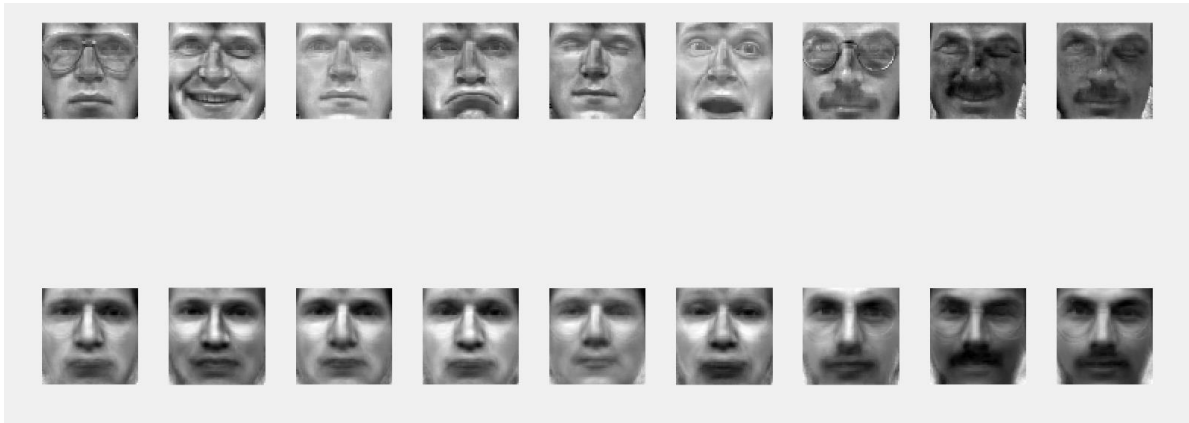


Figure 5 : Reconstruction d'images avec $K=11$

Nous avons les images de départ sur la ligne du haut, et leur reconstruction associées sur la ligne du bas. Nous voyons que avec $K=11$, il n'y a pas « d'erreurs », mais ce n'est pas parfait

Pour $K=90$:

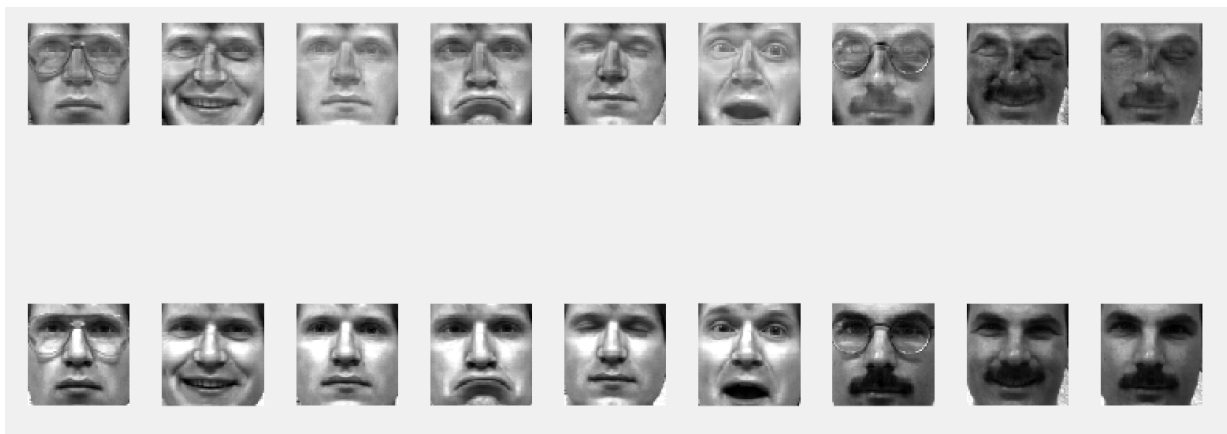


Figure 6 : Reconstruction d'images avec $K=90$

Dans ce cas, le résultat est vraiment bluffant et correspond à la réalité ! Malheureusement, cette solution est très coûteuse en espace mémoire.

V - Conclusions

Au cours de ce TP, nous devons décortiquer une méthode d'apprentissage pour la reconnaissance de visages. Même si nous n'avons pas fait la dernière étape, il a été très formateur, et nous avons pu centrer, recomposer et reconstruire des images issues d'une base de donnée.