

Projet en Calcul Scientifique/Analyse de Données :

Séance 3 : Reconnaissance et Reconstruction de Visages

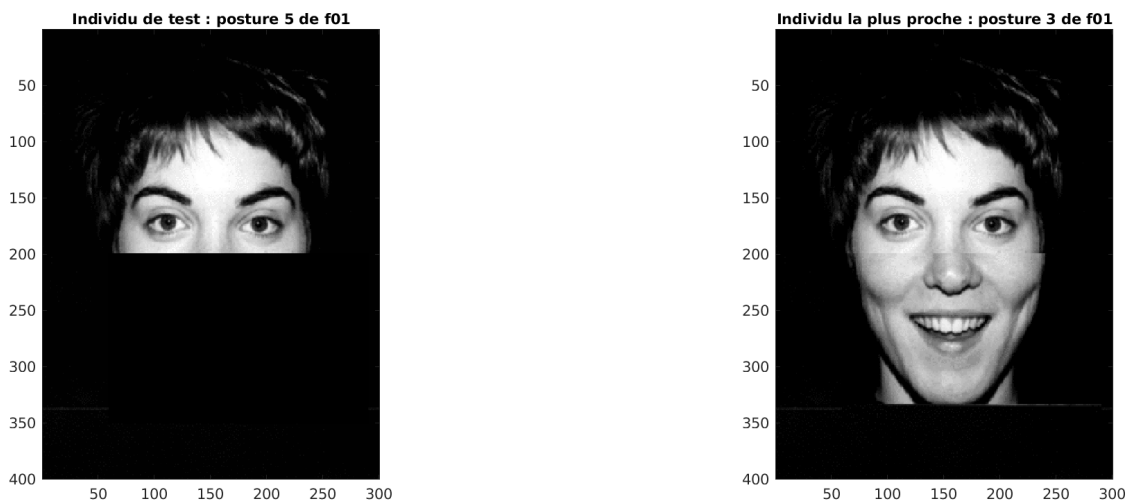


FIGURE 1 – Reconstruction du visage d’une personne dont la posture n’est pas connue

**Projet réalisé par : GUILLOTIN Damien, JOURDAN Pierre-Eliot
et HEURTEBISE Tom**

Encadré par : LOYE Justin

Table des matières

Table des figures	2
1 Introduction	3
2 Retour sur les eigenfaces	4
2.0.1 Question 1	4
3 Reconnaissance et reconstruction	8
3.1 Reconnaissance	8
3.1.1 Question 2	8
3.1.2 Question 3	10
3.2 Reconstruction	13
3.2.1 Question 4 : Reconstruction	13
3.2.2 Question 5 : évaluation	15
4 Bibliographie	17

Table des figures

1	Reconstruction du visage d'une personne dont la posture n'est pas connue	1
2	Morceau de la chaine de traitement pour la reconnaissance faciale (partie reconnaiss- sance seulement)	3
3	Base d'apprentissage non masquée considérée pour les questions 1 à 4	4
4	Eigenfaces non masquées	5
5	Méthode pour créer les masques (lignes 160-161)	5
6	Base d'apprentissage masquée considérée pour les questions 1 à 4	6
7	Eigenfaces masquées	6
8	Utilisation de subspace_iter_v3	7
9	Conditionnement de Sigma 2	7
10	Inertie en fonction du nombre de composante principale	8
11	3 et 2 premières composantes principales de l'ACP	9
12	Code pour la reconnaissance	10
13	Tests du classifieur sur des individus de la base d'apprentissage (à gauche image soumis, à droite image proposée par le classifieur)	11
14	Test avec un individu connu mais dont la posture ne l'est pas	11
15	Classification avec un individu qui n'est pas dans la base	12
16	Tests du classifieur sur des individus masqués de la base d'apprentissage masquée (à gauche image soumis, à droite image proposée par le classifieur)	12
17	Test avec un individu connu mais dont la posture ne l'est pas	13
18	Classification avec un individu qui n'est pas dans la base	13
19	Tests de la reconstruction sur des individus masqués de la base d'apprentissage mas- quée (à gauche image soumis, à droite image proposée par le classifieur)	14
20	Test avec un individu connu mais dont la posture ne l'est pas	14
21	Classification avec un individu qui n'est pas dans la base	15
22	Validation croisée sur 8 itérations et en gardant 2 hommes et 2 femmes par itération comme base d'apprentissage	16
23	Erreur quadratique moyenne obtenue pour l'ensemble des tests	16

1 Introduction

Ce rapport est le troisième et dernier du projet de Calcul Scientifique/Analyse de Données, la lecture des deux premiers est un préalable à la compréhension de ce que vous allez lire.

Au cours de la troisième séance de projet nous avons mené à bien le but initial du projet qui était de pouvoir reconstruire numériquement le visage de personnes masquées. Le travail de cette séance s'est appuyé sur les deux premières dont voici un bref rappel.

Pendant la première séance nous avons mis en forme les "eigenfaces", produit d'une ACP permettant de réduire les dimensions du problème (séance 1). Grâce au travail de la séance 2, cette ACP a été réalisée de façon efficace vis à vis du calcul des vecteurs propres nécessaires. Si on récapitule l'avancement sous la forme d'un schéma nous en étions à l'obtention du vecteur c .

$$\mathbf{x} \in \mathbb{R}^{120000} \rightarrow \boxed{\text{ACP}} \rightarrow \mathbf{c}(\mathbf{x}) \in \mathbb{R}^{q \ll 120000} \rightarrow \boxed{h} \rightarrow h(\mathbf{x})$$

FIGURE 2 – Morceau de la chaîne de traitement pour la reconnaissance faciale (partie reconnaissance seulement)

Durant cette troisième séance nous avons donc utilisé la méthode des k-plus-proche-voisins afin d'identifier l'image de la base d'apprentissage la plus proche d'une image de test. Bien entendu nous avons appliqué cette algorithme sur les composantes principales de l'image de test et sur celles de la base de données afin de drastiquement réduire le temps de calcul. Une fois que la reconnaissance s'est avérée efficace nous avons opérée cette dernière sur un visage masqué afin de trouver l'image non masquée correspondante dans la base d'apprentissage masquée et puis par suite dans celle non masquée. Enfin puisque nous disposons de l'image sans masque la plus proche de notre visage test il suffisait de découper la zone correspondante à celle masquée pour la coller sur notre visage test. Ainsi nous sommes parvenus à reconstruire le visage situé sous le masque d'une image test. Notez par ailleurs, nous y reviendrons, que les performances de cette reconstruction sont optimales lorsque l'on teste une image de la base d'apprentissage mais que celles-ci diminuent en utilisant un autre visage quelconque.

Enfin vous pourrez noter que nous nous sommes cantonnés à 4 personnes et 4 postures pour construire la base d'apprentissage afin de pouvoir tester le classifieur et le reconstituteur sur le reste des images fournies.

NB : Tout ce rapport se base sur [1] et répond aux questions posées par le sujet [2]

2 Retour sur les eigenfaces

Vous noterez que jusqu'à la dernière question nous n'avons exploité que la base d'apprentissage fournie par défaut dans le script et ceci afin de simplifier la compréhension et rendre lisible les affichages graphiques. Nous aurions pu très bien prendre les 6 postures pour les 4 personnes ou encore considérer 16 personnes sur les 32 mais conserver les paramètres par défaut a été fait a dessein : de la validation croisée dans la dernière partie.

2.0.1 Question 1

1) Lire le code de *eigenfaces_part3.m* pour comprendre le rôle des variables

A la lecture du code nous avons immédiatement remarqué que des données mises à disposition (image de 32 personnes x 6 postures), une base d'apprentissage est extraite et comporte les images de 4 personnes x 4 postures. Notons aussi que chaque image est vectorisée et rangée en colonne dans une matrice X sur laquelle on applique l'ACP effectuée durant la première séance du projet. Les variables sont par la suite identiques à la première séance.

Une fois le code lancé nous avons obtenu la base d'apprentissage suivante ainsi que les eigenfaces correspondantes :

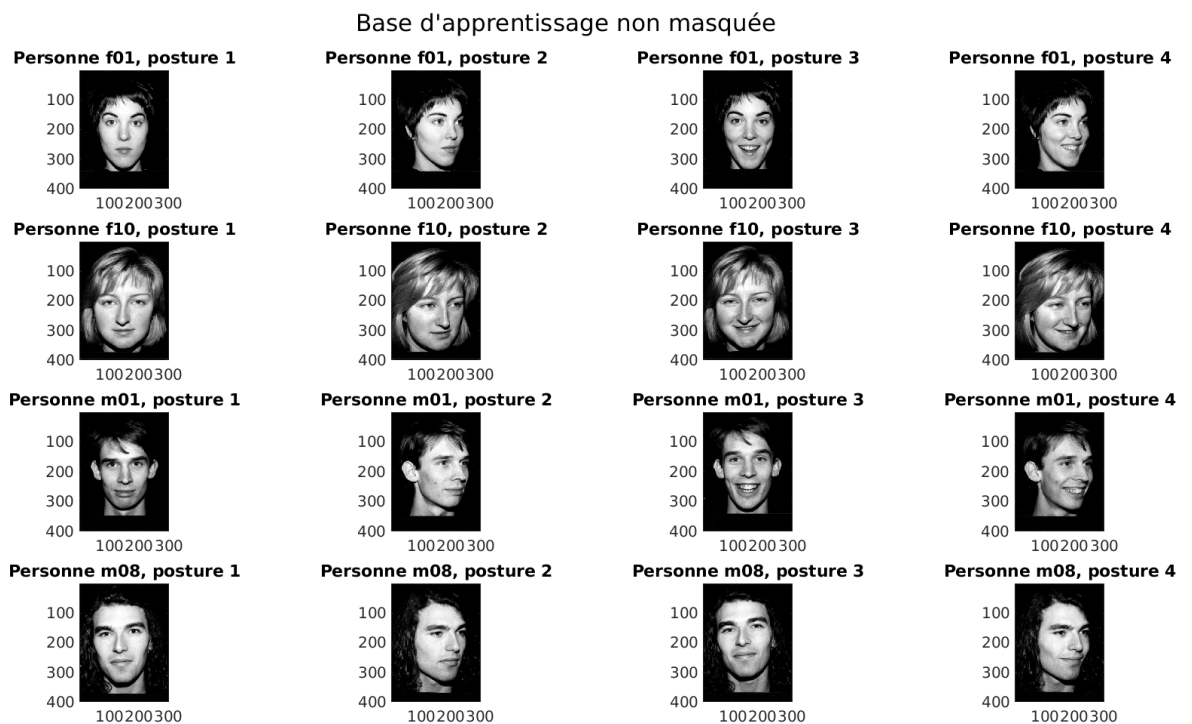


FIGURE 3 – Base d'apprentissage non masquée considérée pour les questions 1 à 4

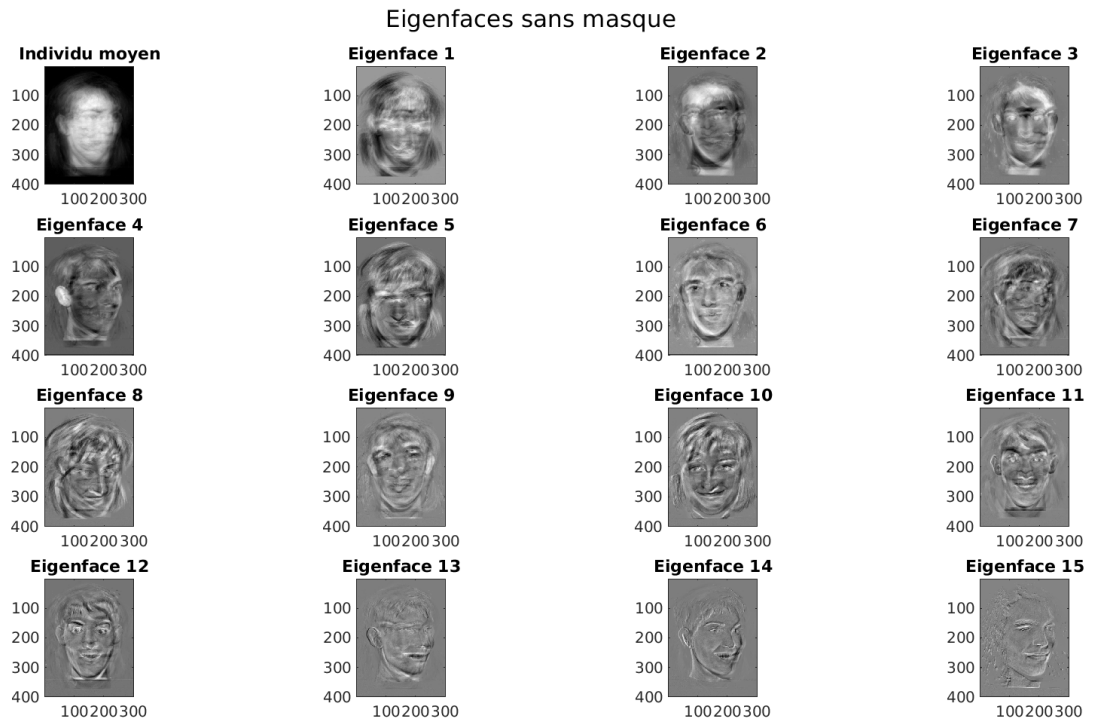


FIGURE 4 – Eigenfaces non masquées

2) Compléter le code pour calculer les "eigenfaces" des visages avec masque

Pour réaliser cette partie nous avons repris le code fourni pour le calcul des eigenfaces sans masque en prenant soin de réaliser une dégradation des images fournies (mise en noir de certains pixels) afin de simuler le port d'un masque. Vous pouvez retrouver cette dégradation ci-dessous (et dans le fichier *eigenfaces_part3.m* :

```
% Dégradation de l'image
img(ligne_min:ligne_max,colonne_min:colonne_max) = 0;
```

FIGURE 5 – Méthode pour créer les masques (lignes 160-161)

Nous avons obtenu la base d'apprentissage suivante ainsi que les eigenfaces correspondantes :

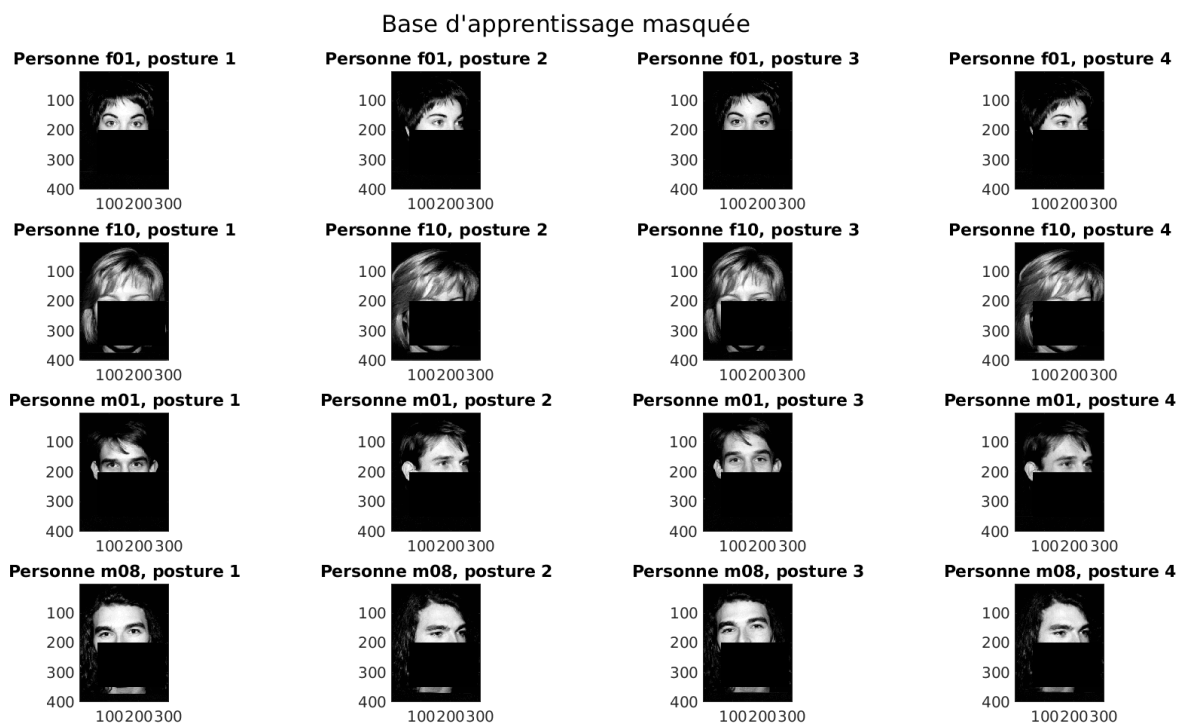


FIGURE 6 – Base d'apprentissage masquée considérée pour les questions 1 à 4

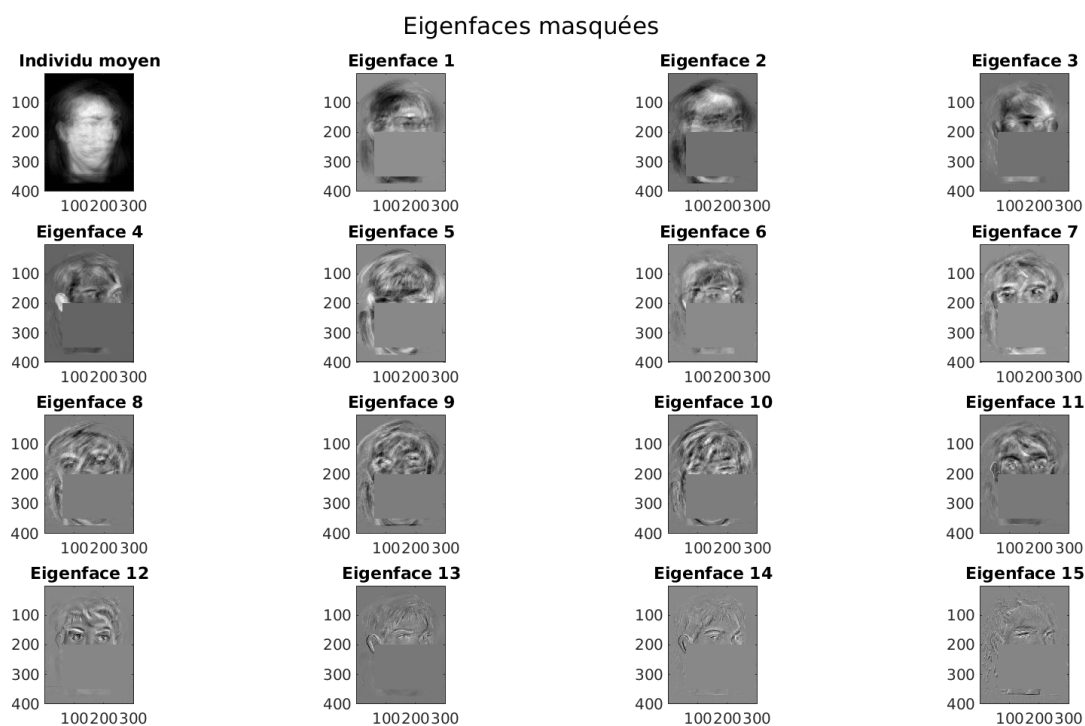


FIGURE 7 – Eigenfaces masquées

3) Modifier le code pour utiliser les fonctions de calcul de couples propres utilisant la méthode "subspace iteration"

Que ce soit pour les visages masqués ou non masqués nous avons substitué la fonction eig par les lignes de code suivantes :

```
m = 16;
percentage = 1.000000001;
p = 1;
eps = 1e-5;
maxit = 100000;

[W_res, V_2] = subspace_iter_v3(Sigma2, m, percentage, p, eps, maxit);
D = diag(W_res);
```

FIGURE 8 – Utilisation de subspace_iter_v3

En effet nous avons choisi cette méthode là implantée lors de la séance 2 en suivant les conclusions que nous avons précédemment faite. Pour rappel, il s'agissait de dire que pour les matrice avec un fort conditionnement il vaut mieux utiliser subspace_iter_v3 au détriment de toutes les autres méthodes testées qui sont plus lentes et moins précises. Or nous avons trouvé le conditionnement suivante pour la matrice *Sigma2* :

```
conditionnement_sigma2 =
    1.2563e+17
```

FIGURE 9 – Conditionnement de Sigma 2

Il était donc évident qu'il fallait utiliser *subspace_iter_v3*. Vous noterez que pour les paramètres de cette dernière nous avons pris un pourcentage de la trace très légèrement supérieur à 100%. Ceci est dû au fait qu'en prenant 100% la convergence se stoppe juste avant le calcul de la dernière valeur propre puisque les valeurs propres ne sont pas assez précises à 10^{-10} près et leur somme est légèrement trop élevée en conséquence.

3 Reconnaissance et reconstruction

3.1 Reconnaissance

3.1.1 Question 2

1) Rajoutez dans le script *eigenfaces_part3.m*, le calcul des contrastes si on considère 1, 2, 3,... "eigenfaces" (sans ou avec masque)

Comme nous n'avons pas à disposition *Sigma* mais *Sigma2* nous n'avons pas pu calculer les contrastes mais plutôt les inerties correspondantes aux k premières composantes principales. Rappelons que l'inertie peut s'exprimer de la façon suivante :

$$I_k = \frac{\sum_{i=0}^k \lambda_i}{\sum_{i=0}^n \lambda_i}$$

avec λ_i la *ime* valeur propre de la matrice de variance-covariance Σ . Rappelons que nous avons précédemment montré que les valeurs propres de *Sigma* et *Sigma2* sont identiques. C'est pourquoi nous nous sommes basés sur les valeurs propres de *Sigma2* pour obtenir la courbe suivante :

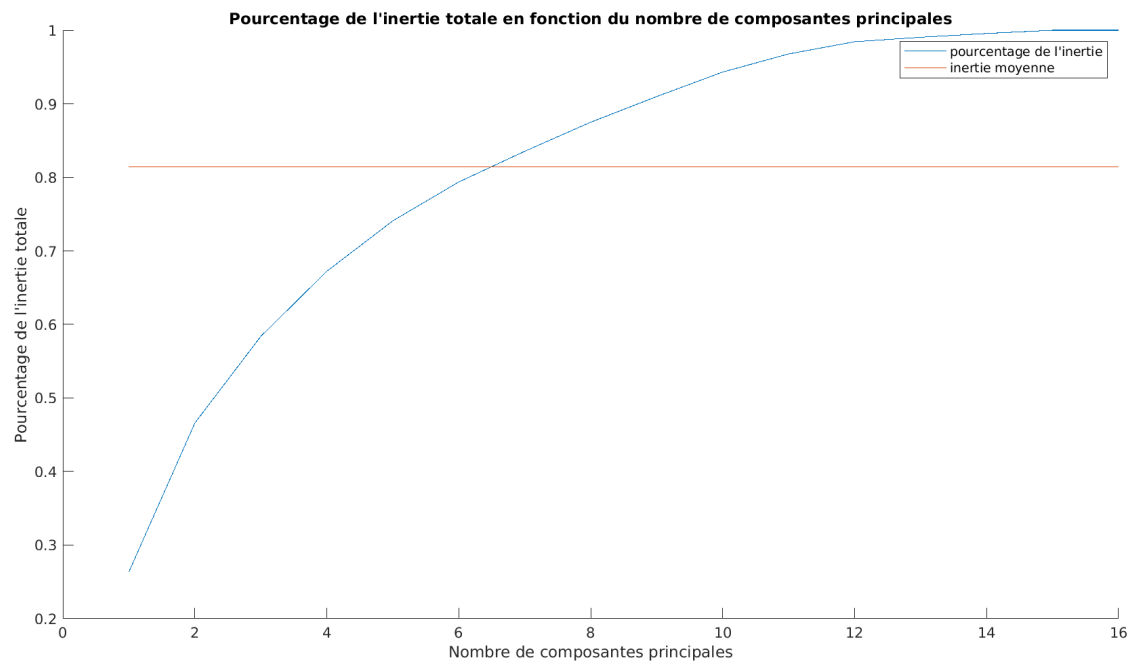
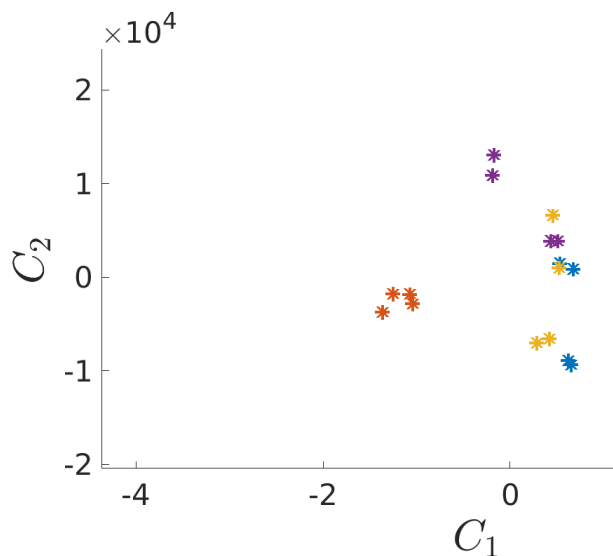


FIGURE 10 – Inertie en fonction du nombre de composante principale

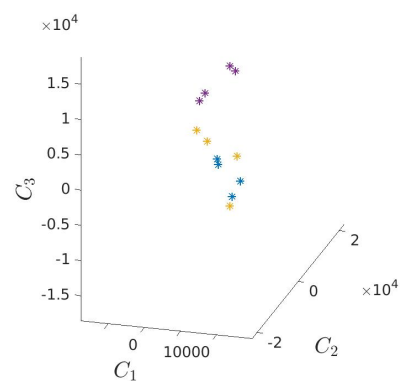
D'après ce graphique et le critère de Kaiser il suffirait de prendre les 7 premières composantes principales pour avoir une perte d'information acceptable dans notre ACP. Notez que par la suite nous souhaitons conserver 95% de l'information originale dans notre ACP et que d'après ce graphique nous avons donc pris 11 composantes principales.

2) Dans le script *clusters.m*, rajoutez l’affichage avec les 3 premières composantes (affichage dans \mathbb{R}^3).

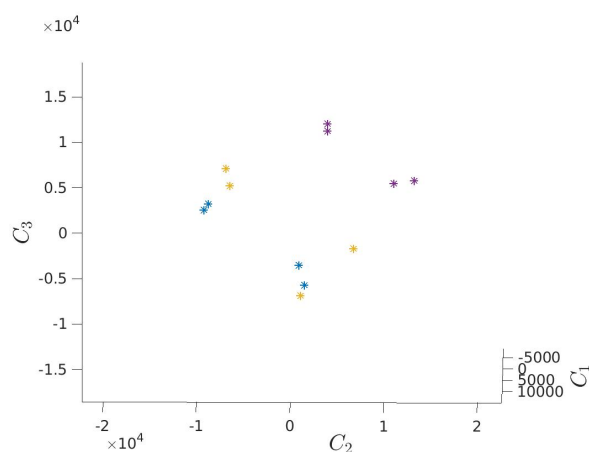
En réalisant la consigne nous avons obtenu les deux graphiques suivants :



(a) 2 premières composantes principales



(b) 3 premières composantes principales, vue 1



(c) 3 premières composantes principales, vue 2

FIGURE 11 – 3 et 2 premières composantes principales de l’ACP

Il semble bien que considérer plus de composantes poussent les composantes à mieux se répartir dans l’espace. Ce que nous avons établi à la question précédente est vérifié.

3.1.2 Question 3

Pour les questions qui suivent nous avons apportés quelques modifications puisque les images ne sont pas tirées aléatoirement dans la base totale des images mais dans la base d'apprentissage. Nous effectuons aussi un test sur avec des images pour lesquelles la personne est dans la base d'apprentissage mais pas sous cette posture et enfin un test pour lequel la personne n'est pas dans la base d'apprentissage.

1) Complétez le script pour trouver l'individu (personne+posture) dans la base d'apprentissage qui est le plus proche de l'image de test

Nous avons simplement ajouté les lignes de codes suivantes pour répondre à cette question :

```
N = 11;      % calculé d'après l'inertie
K = 1;
ListeClass = 1:nb_postures_base*nb_personnes_base;

C = X_centre*W;
c = (image_test-X_moyen)*W;

partition = kppv(C, ListeClass, c, 1, K, ListeClass);

% pour l'affichage (A CHANGER)
personne_proche = floor((partition-1)/nb_postures_base)+1
posture_proche = mod(partition-1,nb_postures_base)+1
```

FIGURE 12 – Code pour la reconnaissance

Ainsi dans ces lignes on crée un ensemble de label intitulé *ListClass*, on prend on projette l'image de test sur les vecteurs principaux de la base d'apprentissage et on effectue un appel à la méthode du k plus proches voisins. Après divers tests ils nous a semblé que considérer un seul voisin est suffisant. Notez que le gros du travail ici est sur la façon dont nous retrouvons le numéro de la personne et celui de sa posture.

Pour ce script nous avons effectué divers tests avec des personnes faisant partie de la base d'apprentissage tout d'abord :

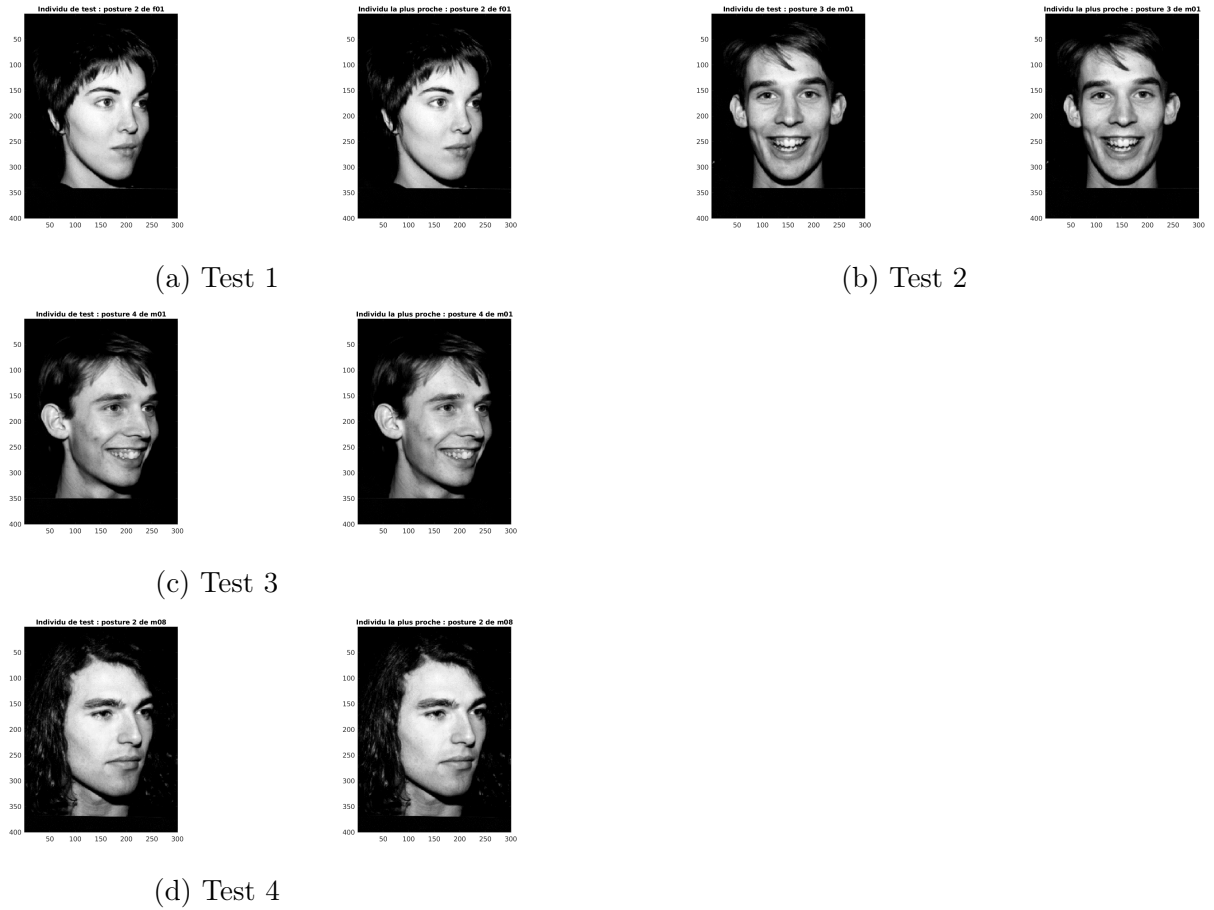


FIGURE 13 – Tests du classifieur sur des individus de la base d'apprentissage (à gauche image soumise, à droite image proposée par le classifieur)

Ensuite nous avons proposé au classifieur un individu dans la base d'apprentissage mais dans une posture qui lui est inconnue :

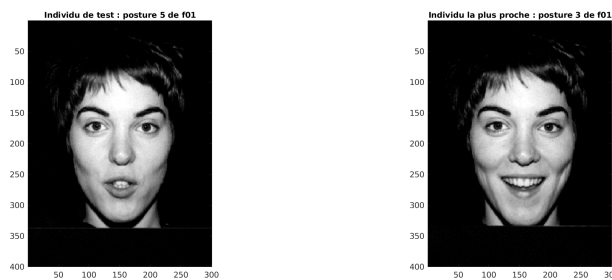


FIGURE 14 – Test avec un individu connu mais dont la posture ne l'est pas

On constate que le résultat est très proche et le classifieur semble de bonne qualité. Enfin nous avons testé le classifieur en lui soumettant l'image d'un individu qu'il ne connaît pas :

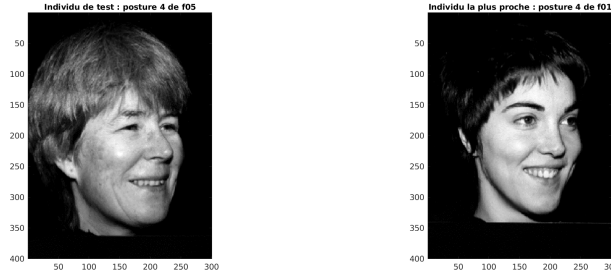


FIGURE 15 – Classification avec un individu qui n'est pas dans la base

On note alors que la posture est bonne et que la personne que le classifieur propose ressemble vaguement à celle qu'on lui soumet. Ceci semble tout à fait correct.

2) En recopiant le script dans un fichier *reconnaissance_avec_masque.m*, faites la reconnaissance avec masque

Par la suite nous avons répété la classification en prenant soin de masquer les individus au préalable. En suivant la même logique de tests que ce qui précède en obtient les résultats suivants :

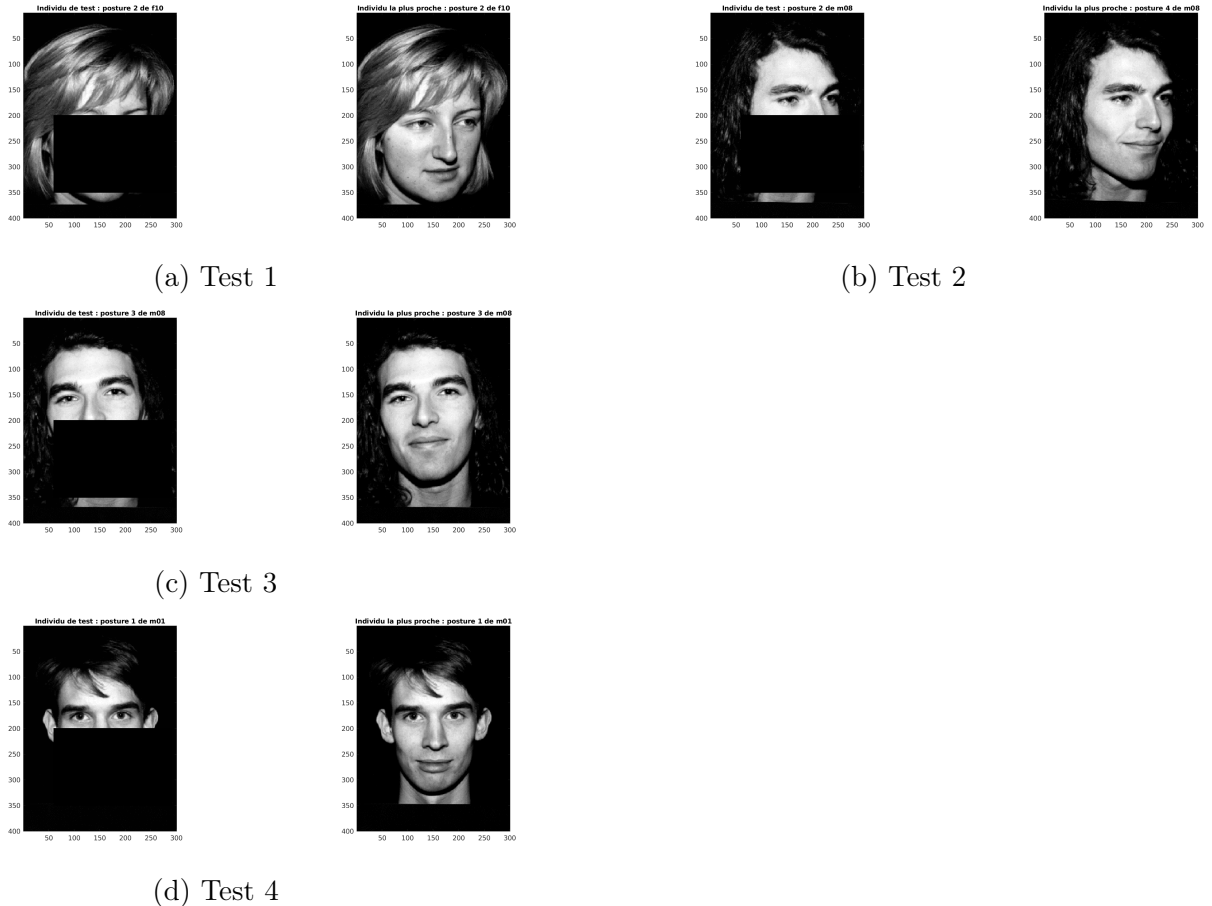


FIGURE 16 – Tests du classifieur sur des individus masqués de la base d'apprentissage masquée (à gauche image soumis, à droite image proposée par le classifieur)

Ensuite nous avons proposé au classifieur un individu dans la base d'apprentissage mais dans une posture qui lui est inconnue :

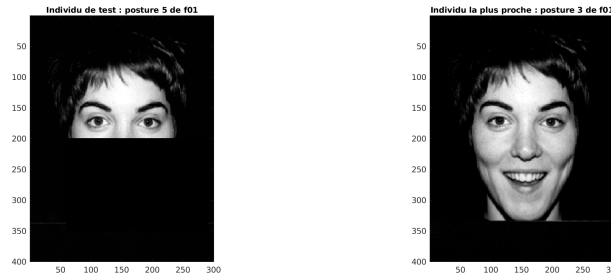


FIGURE 17 – Test avec un individu connu mais dont la posture ne l'est pas

On constate que le résultat est très proche et pourrait très bien correspondre à la posture de la personne.

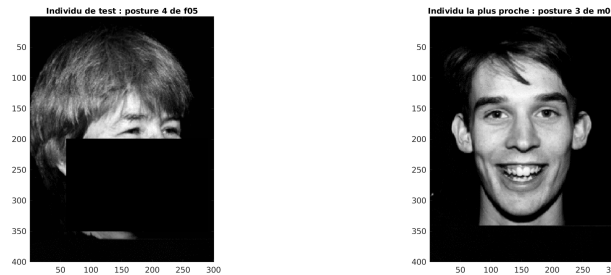


FIGURE 18 – Classification avec un individu qui n'est pas dans la base

Pour ce résultat cependant il semble que l'on soit assez éloigné de la réalité et que le classifieur ait ses limites.

3) Identifiez les paramètres de votre classifieur et essayez de trouver des valeurs optimales à ces paramètres (sans ou avec masque)

Comme évoqué précédemment nous avons constaté qu'en utilisant $K = 1$ et la norme2 on obtient des résultats relativement corrects.

3.2 Reconstruction

3.2.1 Question 4 : Reconstruction

En partant du script *reconnaissance_avec_masque.m* (que vous recopierez dans un fichier *reconnaissance_reconstruction.m*), développez le code permettant de reconstruire le visage de test.

Par la suite nous avons utilisé l'image retrouvée par classification pour venir découper la zone masquée et la coller sur l'image de test. Ci-dessous vous avez les résultats que nous obtenons pour des images issues de la base d'apprentissage.

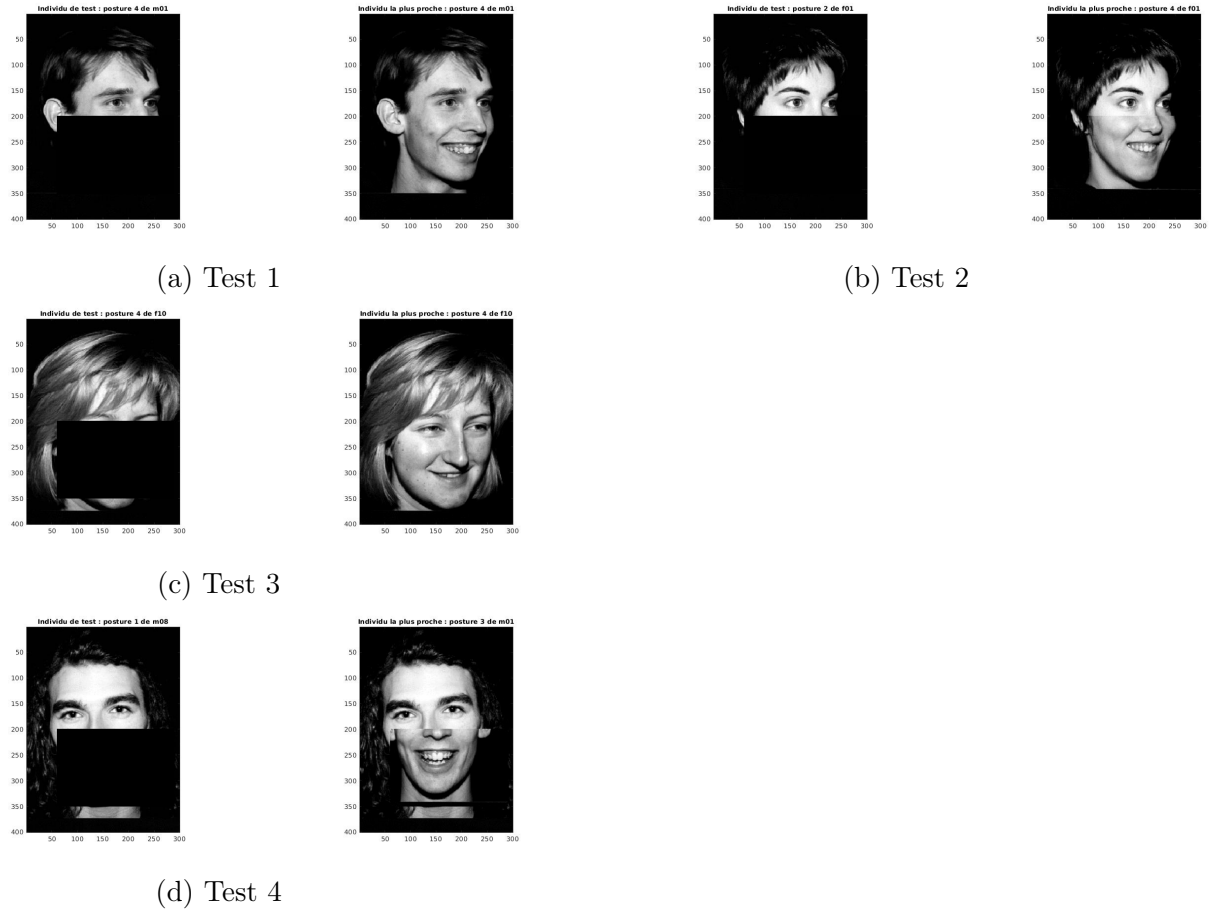


FIGURE 19 – Tests de la reconstruction sur des individus masqués de la base d'apprentissage masquée (à gauche image soumis, à droite image proposée par le classifieur)

Ensuite nous avons proposé au restructeur un individu dans la base d'apprentissage mais dans une posture qui lui est inconnue :

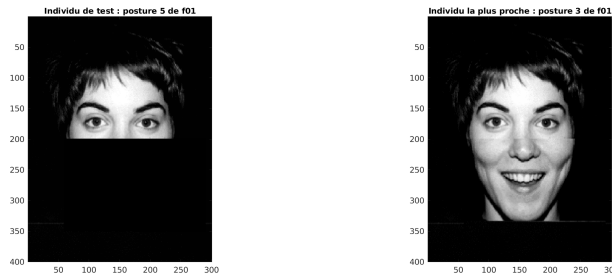


FIGURE 20 – Test avec un individu connu mais dont la posture ne l'est pas

On constate que le résultat est très proche et pourrait presque correspondre avec l'image d'origine.

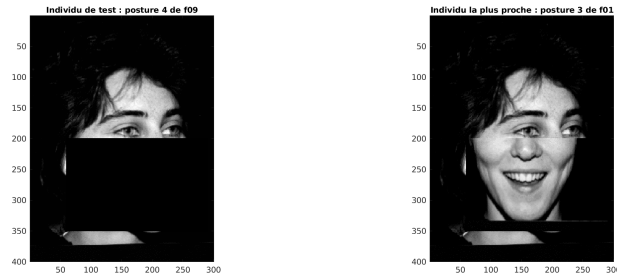


FIGURE 21 – Classification avec un individu qui n'est pas dans la base

Pour ce résultat cependant il semble que l'on soit assez éloigné de la réalité et que le reconstituteur ait ses limites.

3.2.2 Question 5 : évaluation

Pour évaluer la reconnaissance du classifieur, vous utiliserez les images reconstruites correspondant à plusieurs requêtes. En réutilisant les outils qui vous ont été fournis ou que vous avez développés pour la partie 1 et cette partie 3, imaginez une évaluation de la reconstruction.

Afin d'évaluer la reconstruction nous avons opéré une validation croisée en divisant les 32 personnes en classe 4 personnes chacun (toujours 2 hommes et 2 femmes). Voici ce que nous avons fait schématiquement :

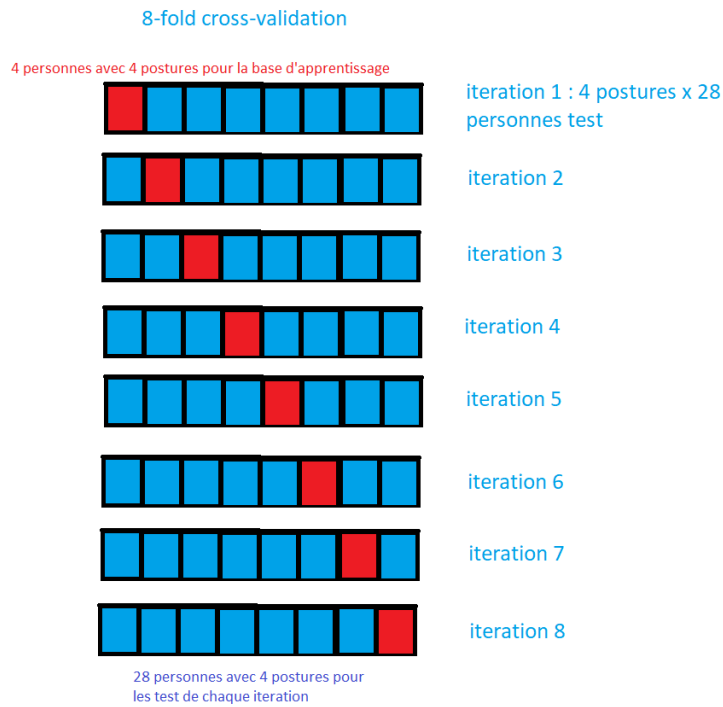


FIGURE 22 – Validation croisée sur 8 itérations et en gardant 2 hommes et 2 femmes par itération comme base d'apprentissage

Au cours de la première itération nous utilisons une base d'apprentissage constituée de 2 hommes et 2 femmes et nous testons la reconstruction sur les $32 - 4$ personnes restantes, puis après la base d'apprentissage change au cours d'une deuxième itération,... etc Pour chaque zone masquée reconstruite nous avons calculé l'erreur en moyenne quadratique et au final nous avons calculer la moyenne de ces erreurs quadratiques pour obtenir le résultat suivant qui ne paraît pas aberrant :

$$\text{ecart_quadratique_moyen} = 6.4856$$

FIGURE 23 – Erreur quadratique moyenne obtenue pour l'ensemble des tests

Conclusion

Au cours ce projet nous avons pu mettre en pratique les outils étudiés en Analyse de Données ainsi qu'en Calcul Scientifique. Ce projet a été très intéressant et nous a montré ce qu'il est possible de faire avec les connaissances que nous avons acquises au cours de ces derniers mois.

Notons cependant que notre "reconstructeur" est très loin d'être parfait et mériterait d'être entraîné sur une base d'apprentissage beaucoup plus étendue. Il s'agit ainsi plus d'un projet à vocation pédagogique que d'un projet qui pourrait être réutilisé en l'état.

4 Bibliographie

- [1] Matthew Turk and Alex Pentland. Eigenfaces for recognition. J. Cognitive Neuroscience, 3(1) :71–1991, 1991.
- [2] Sujet séance-Projet 3 : Reconnaissance et Reconstruction de Visages, 2020-2021. Sujet de la séance 3 du projet de Calcul Scientifique/Analyse de Données.