

1. Introduction

Dans ce chapitre, nous allons consacrer la première section à présenter les outils que nous avons utilisés dans l'implémentation de notre solution. Une deuxième section permettra d'implémenter nos approches décrites au chapitre précédent. La troisième section sera réservée à exposer les interfaces graphiques de notre application développée en langage GUI MATLAB.

2. Environnement du travail

2.1 Matlab

Avant toute chose, il est important de définir ce qu'est MATLAB. Pour certains, c'est un logiciel, un outil, pour d'autres un langage.

En fait, c'est un peu tout cela.

- lorsque l'on parle du logiciel MATLAB, on fait référence à l'outil que l'on utilise, l'interface utilisateur ;
- lorsque l'on parle du langage MATLAB, on désigne la syntaxe spécifique que l'on met en œuvre dans cet outil.

Le nom MATLAB vient de l'anglais *MATrix LABoratory* une traduction littérale nous amène à voir MATLAB comme un laboratoire pour manipuler des matrices. Nous reviendrons sur ce point, qui est un élément fondamental du langage MATLAB : la plupart des fonctions définies dans MATLAB le sont pour des grandeurs matricielles, et par extension, pour des données tabulées.

MATLAB comprend de nombreuses fonctions, de calcul ou de traitements de données, d'affichage, de tracés de courbes, de résolution de systèmes et d'algorithmes de calculs numériques au sens large du terme.

Toutes ces fonctions sont définies par défaut dans MATLAB dans un langage de programmation spécifique que l'on appellera... MATLAB !

Ce langage comprend de nombreuses fonctions prédéfinies pour le calcul matriciel, mais pas seulement. De ce fait, les domaines d'application sont extrêmement variés, et l'on peut citer par exemple :

- le calcul numérique dans le corps des réels ou des complexes ;
- le calcul de probabilités ou les statistiques ;
- le calcul intégral ou la dérivation ;
- le traitement du signal ;
- l'optimisation ;
- le traitement d'image ;
- l'automatisme. [11]

2.2 La base de données ORL

La base de données ORL des visages contient un ensemble d'images faciales prises entre avril 1992 et avril 1994 au laboratoire. La base de données a été utilisée dans le cadre d'un projet de reconnaissance de visage mené en collaboration avec le Groupe de la parole, de la vision et de la robotique du Département d'ingénierie de l'Université de Cambridge.

Il existe dix images différentes de chacun de 40 sujets distincts. Pour certains sujets, les images ont été prises à différents moments, variant l'éclairage, les expressions faciales (yeux ouverts / fermés, souriant / pas souriant) et les détails du visage (lunettes / pas de lunettes). Toutes les images ont été prises contre un fond sombre homogène avec les sujets en position verticale et frontale (avec tolérance pour certains mouvements latéraux).

Les fichiers sont en format BMP. La taille de chaque image est de 92x112 pixels, avec 256 niveaux de gris par pixel. Et sont organisées dans 40 répertoires (un pour chaque sujet entre 1 et 40). Dans chacun de ces répertoires, il existe dix images différentes de ce sujet, entre 1 et 10. [2]



Figure 4.1 Échantillons de la base ORL

2.3 La base de données FERET

La base de données FERET présente 1199 personnes pour un total de plus de 14000 images couleur (ou non) de taille 512 _ 768 (dans la nouvelle version de la base).

La base présente des variations d'expressions faciales, de pose, d'illumination ainsi qu'un certain délai entre différentes captures d'une même personne. Le protocole bien défini ainsi que le très grand nombre d'images disponibles en ont fait une des bases de données les plus populaires en reconnaissance de visages. [26]



Figure 4.2 Échantillons de la base FERET

3. Principe de notre système

On peut définir notre système et son rôle en mode identification, ce qui le processus de comparaison d'un échantillon biométrique inconnu à de multiples références dans une base de données pour connaître l'identité du propriétaire de cet échantillon inconnu. Les étapes de ce système sont situées dans la figure suivant :

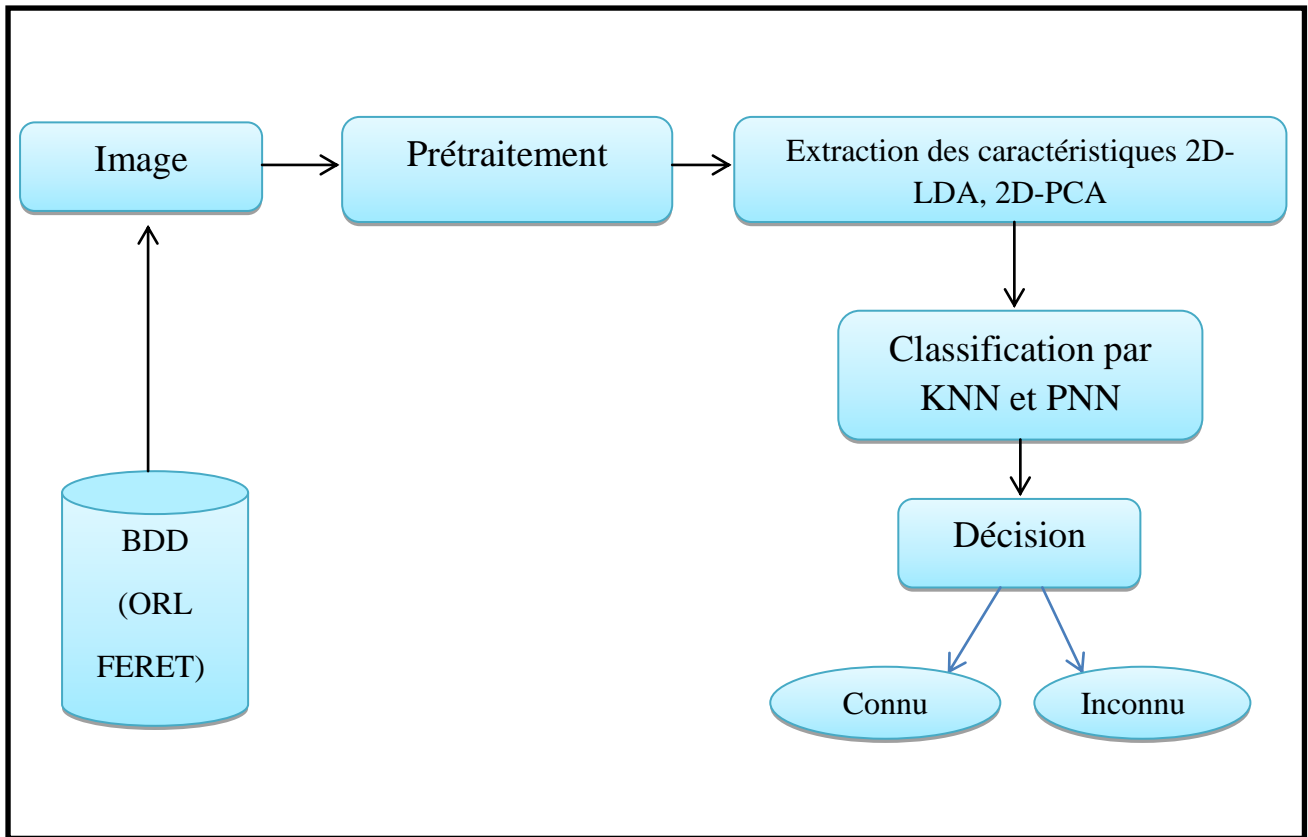


Figure 4.3 le processus de reconnaissance

1. Prétraitement

En premier lieu on a utilisé deux différentes bases de données, la première ORL est composée de 40 individus, chaque individu a 10 captures de l'index et le deuxième FERET est composée de 35 individus, chaque individu à 20 captures de l'index.

L'étape de prétraitement consiste à redimensionner la taille de chaque image de la base et aussi convertir les images en niveau de gris, pour l'amélioration d'images.

2. Extraction des caractéristiques

L'algorithme d'extraction traite le signal biométrique d'entrée d'origine pour en extraire de fortes caractéristiques reproductibles. Le but de cette approche est de réduire l'espace de stockage, alors Dans cette étape nous avons appliqué deux méthodes suivant : PCA, 2D-PCA, 2D-LDA

3. Classification

Classer les objets revient à les situer les un(e)s par rapport aux autres. Alors la classification opérer des regroupements en classes homogènes d'un ensemble d'individus.

Pour la classification, une distance Euclidienne a été utilisée, ils réussissent à obtenir une bonne performance, on a aussi utilisé les réseaux de neurones (PNN)

4. Décision

La décision consiste la mesure de similarité entre les défèrent images. Dans ce cas, nous obtenons les décisions en compte de deux réponse (Connu/ Inconnu).

4. Résultats et expérimentaux

Dans cette expérience nous avons évalué le taux de reconnaissance et d'erreur sur les différentes méthodes suivantes : (PCA, 2D-PCA, 2D-LDA).

Le processus de reconnaissance s'appuie sur deux phases précédentes d'extraction et l'apprentissage. Il implique plusieurs images de visages ayant des classes des variations associées.

Le but de notre projet est d'évaluer le taux de reconnaissance de différents algorithmes que nous avons présenté dans le chapitre précédent, en suivant un protocole de test basé sur la mesure de taux de reconnaissance comme le montre l'équation suivante :

$$\text{Taux de reconnaissance} = \frac{\text{le nombre d'images de teste reconnues}}{\text{le nombre totale des images de teste}} \quad (4.1)$$

L'étape d'extraction crée une liste d'images dites « eigenfaces » dont le nombre égale au nombre d'images de la base. Dans notre cas, nous appelons une image de sortie de cette étape une « Eigentest » vue quelle présente les éléments pertinents qui permettent de distinguer entre les différentes variations (expression, lunette,...etc) de notre base. La figure 4.4 ci-dessous visualise certains Eigenface résultants de cette étape.

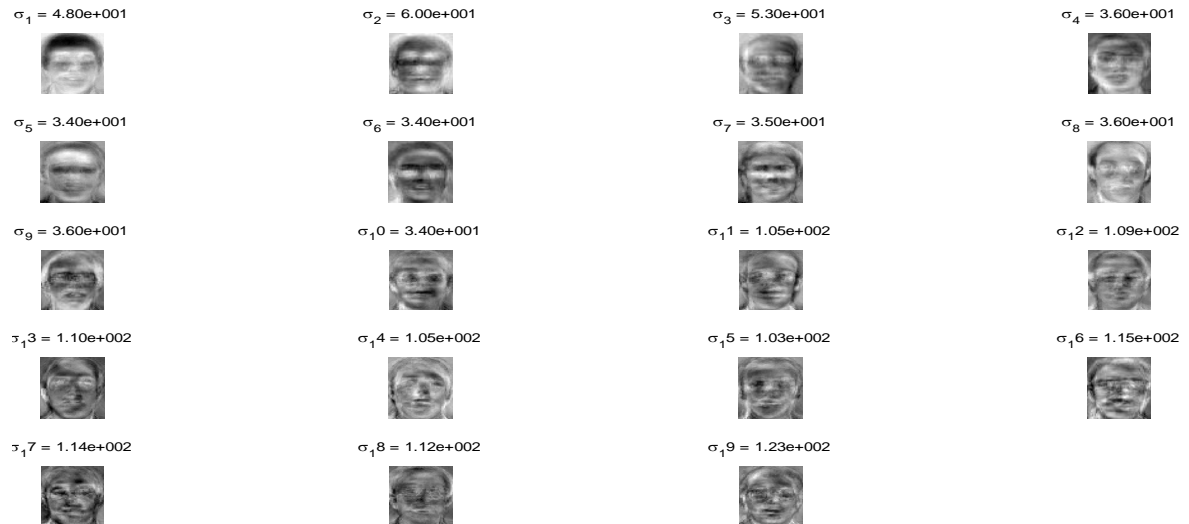


Figure 1 Certains EigenFace calculées à partir de la base ORL.

Une fois les EigenFace sont construites, nous déterminerons la série des poids de chaque image de notre base. Autrement dit, nous calculons les coordonnées de chaque visage dans l'espace engendré par les EigenFace élaborées.

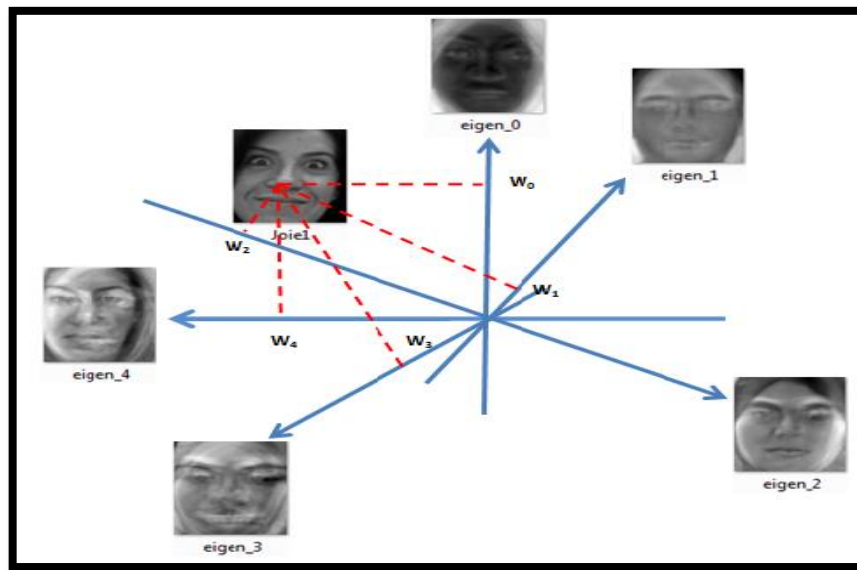


Figure 4.5 Projection d'un visage du teste dans l'espace engendré par les EigenFace.

L'apprentissage sert à maintenir les informations utiles à la classification. Dans le cas de la méthode « EigenFace », cette phase sauvegarde seulement les X premiers vecteurs de poids sous forme d'une matrice dite « matrice de poids ». Il est essentiel de noter que X choisit pour notre implémentation est égale à M (nombre d'images de notre base).

Arrivons maintenant à la reconnaissance proprement dit, Au cours de cette étape, nous réalisons l'extraction du visage de l'utilisateur, ensuite nous calculons le vecteur de poids de l'échantillon présentée par le visage teste et enfin nous comparons le vecteur résultat avec chaque colonne de la matrice de poids. Nous déterminons la classe de visage de test en se basant sur la distance euclidienne minimale (KNN) et le réseau neurone PNN .

a. Architecteur d'un réseau PNN (la Sélection du paramètres PNN)

PNN : l'algorithme d'apprentissage du réseau neuronal PNN utilisé dans notre system reconnaissance de visage du système ne requiert pas de nombreux paramètres que d'autres réseaux de neurones (MLP, BP, LVQ.etc). en revanche un seul paramètre nécessaire pour contrôler la performance du réseau est le paramètre de lissage (smothing) σ . Dans notre application nous avons fait une série d'expériences afin sélection le meilleur paramètre de lissage lequel le réseau va converger.

Le réseau neuronal probabiliste utilisé dans notre système se compose en deux couches:

Couche d'entrée: La première couche est la couche d'entrée et le nombre d'unités cachées est le nombre de variables d'entrée et reçoit les données d'entrée (la taille de vecteur de caractéristique pour chaque approche utilisée dans ce mémoire).

Couche de sortie: la taille de cette couche est le nombre de personnes de la base donnes (Ex: ORL, 40 personnes).

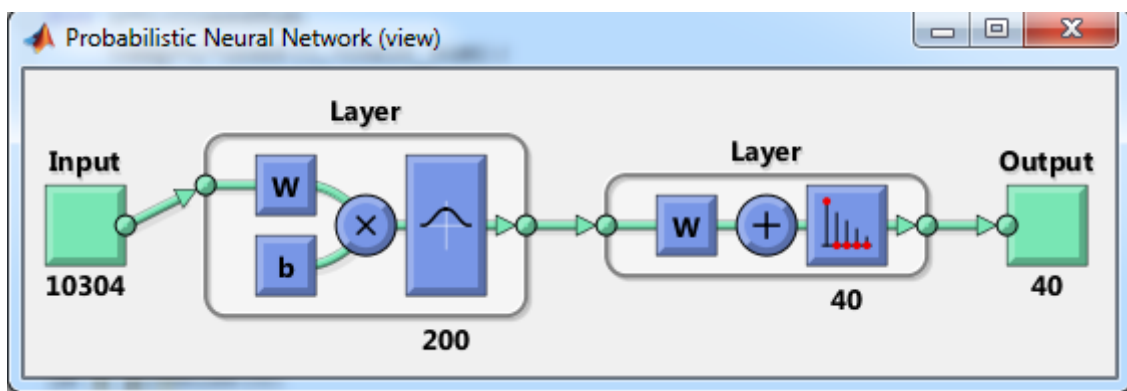


Figure 4.6 Exemple d'architecteur d'un réseau neurone PNN

Toutes les résultats ont été réalisées sur deux base de données ORL et FERET.

- Les méthodes ACP, 2D-ACP, 2D-LDA combiner avec un classifieur PNN sur la Base de données ORL :

	ACP	2D-ACP	2D-LDA
Taux d'erreur	7.5 %	9.5 %	08 %
Performance	92.5 %	90.5 %	92 %
Mse	0.44	0.44	0.34

Tableau 4.1 Taux de reconnaissance des différentes méthodes sur la base ORL.

- Les méthodes ACP, 2D-ACP, 2D-LDA combiner avec un classifieur PNN sur la Base de données FERET :

	ACP	2D-ACP	2D-LDA
Taux d'erreur	17.4 %	17.4 %	16.3 %
Performance	82.6 %	82.6 %	83.7 %
mse	0.32	0.32	0.29

Tableau 4.2 Taux de reconnaissance des différentes méthodes sur la base FERET.

- Les méthodes ACP, 2D-ACP combiner avec un classifieur KNN sur la Base de données ORL :

	ACP	2D-ACP
Taux d'erreur	08 %	07.5 %
Taux de reconnaissance	92 %	92.5%

Tableau 4.3 Taux de reconnaissance des méthodes ACP et 2D-ACP sur la base ORL.

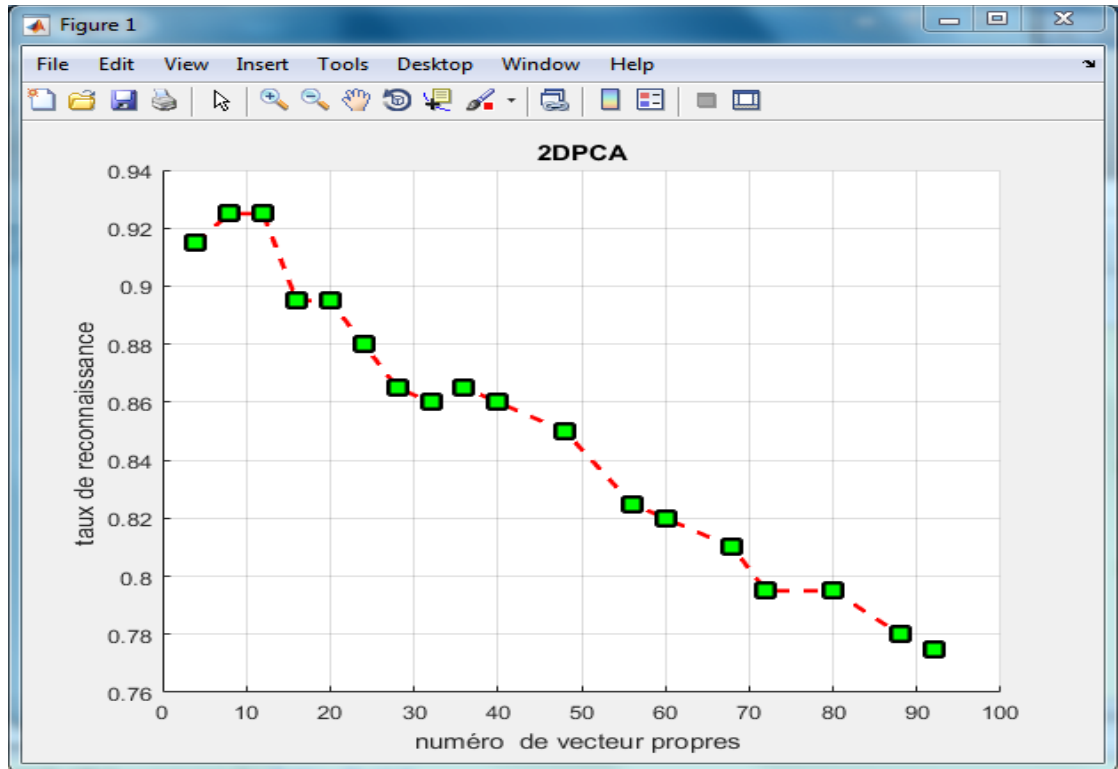


Figure 4.7 Taux de reconnaissance de méthode 2D-ACP sur la base ORL.

- Les méthodes ACP, 2D-ACP combiner avec un classifieur KNN sur la Base de données FERET :

	ACP	2D-ACP
Taux d'erreur	18.3 %	15.5 %
Taux de reconnaissance	81.7 %	84.5 %

Tableau 4.4 Taux de reconnaissance des méthodes ACP et 2D-ACP sur la base FERET.

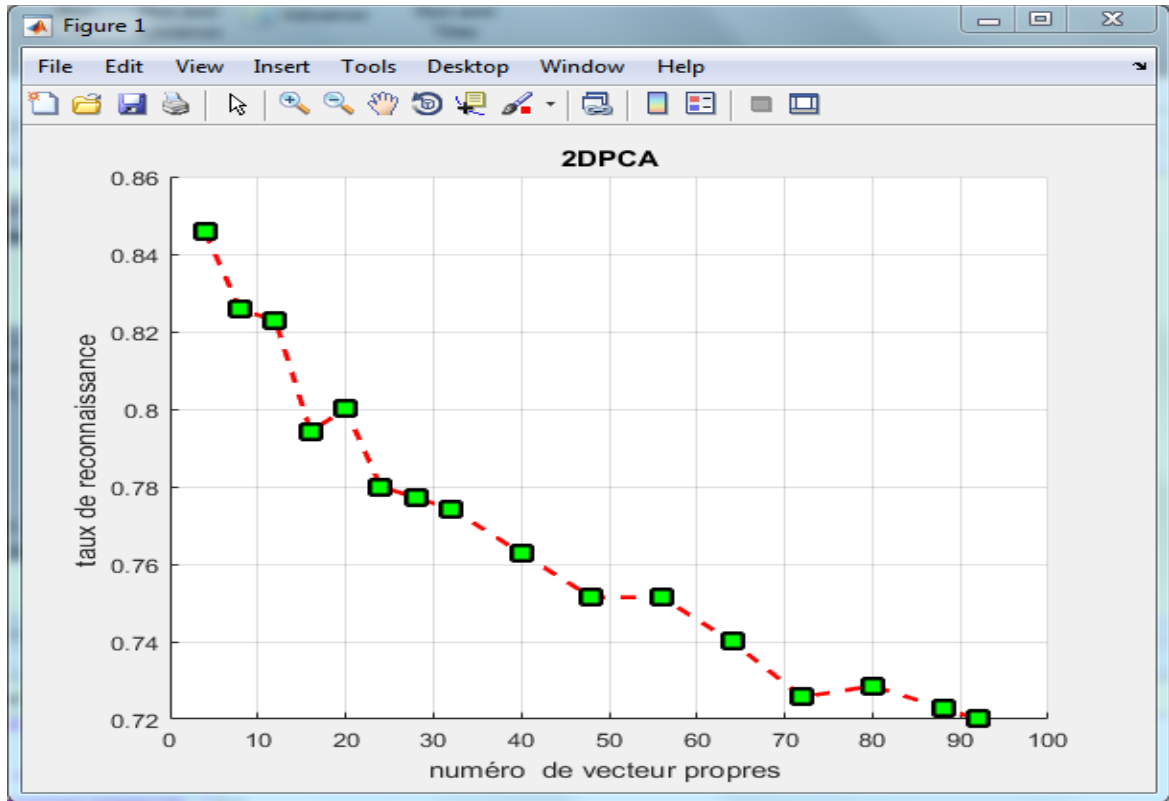


Figure 4.8 Taux de reconnaissance de méthode 2D-ACP sur la base FERET.

- La méthode 2D-LDA combinée avec un classifieur KNN sur la Base de données ORL et FERET :

	2D-LDA (ORL)	2D-LDA (FERET)
Taux d'erreur	05 %	13.5 %
Taux de reconnaissance	95 %	86.5 %

Tableau 4.5 Taux de reconnaissance de méthode 2D-LDA sur la base ORL et FERET.

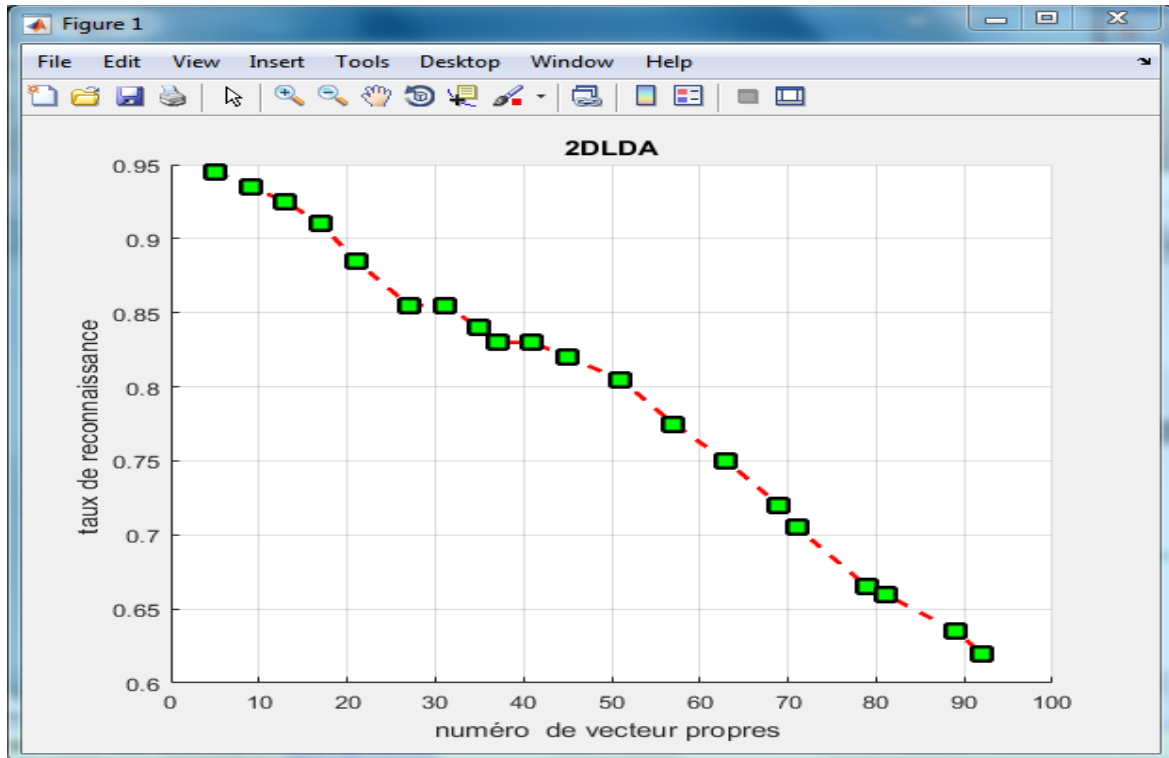


Figure 4.9 Taux de reconnaissance de méthode 2D-LDA sur la base ORL.

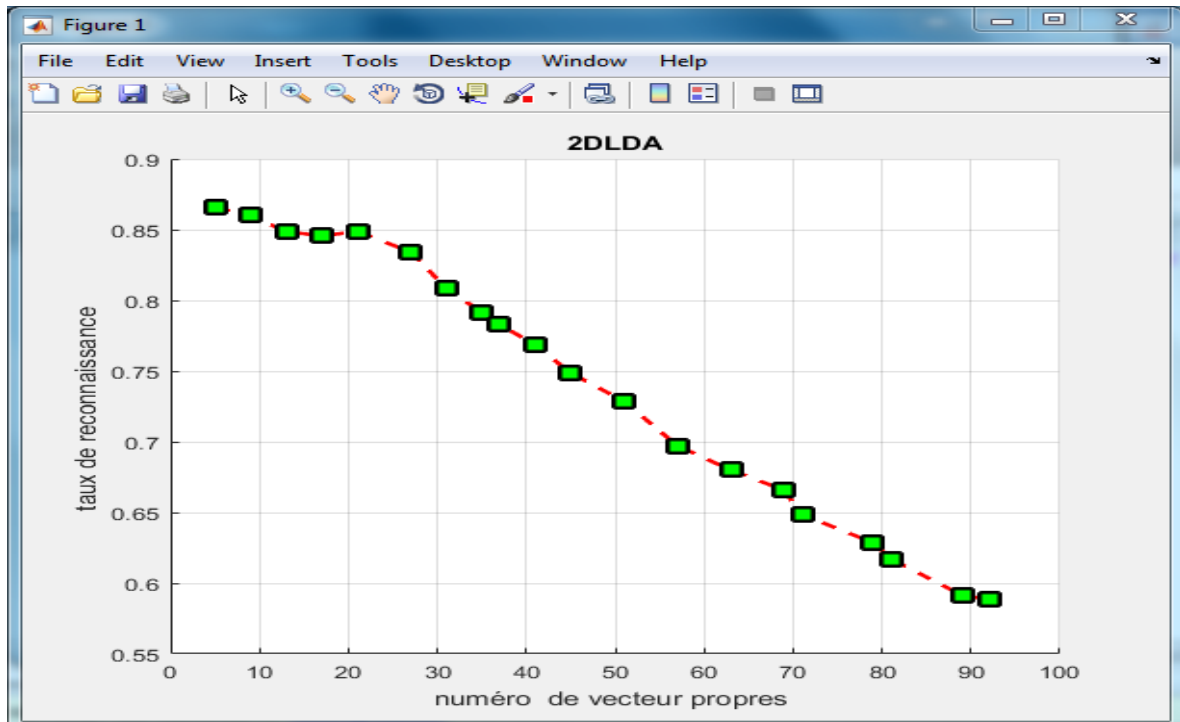


Figure 4.10 Taux de reconnaissance de méthode 2D-LDA sur la base FERET.

➤ **Discussion :**

Les figures (4.7 et 4.8) montrent l'évaluation du taux de reconnaissance en fonction du visage propre choisie. Après une série d'expérience nous remarquons un meilleur taux de reconnaissance on fonction de visage propre ($K=10$) qui a atteint 93% sur la base ORL.

A partir du 11^{eme} vecteur propre nous remarquons une diminution de taux de reconnaissance jusqu'à 88,7%.

Les figures (4.9 et 4.10) montrent l'évaluation du taux de reconnaissance en fonction du visage propre. Après une série d'expériences nous remarquons un meilleur taux de reconnaissance en fonction du septième visage propre ($K=7$) qui a atteint 94,6% sur la base ORL.

A partir du 11^{eme} vecteur propre nous remarquons une diminution du taux de reconnaissance jusqu'à 88,4%.

La diminution de taux de reconnaissance pour les deux approches liées au phénomène sur-apprentissage 'overfitting'.

nous remarquons aussi que le choix de base de données influe sur la performance de notre system(le taux avec la base ORL et meilleur que de la base FERET).

Une série d'expériences ont été réalisées afin montre une comparions entre l'approche PCA et 2DPCA en terme de taux de reconnaissance et temps d'exécution, nous constatons que le taux de reconnaissance obtenu par 2DPCA et 2DLDAest meilleur par rapport à celui obtenu avec PCA et LDA avec le classifieur KNN.

On remarque que le temps d'exécution de 2DPCA est moins long que celui de la PCA.

Après ces figures au-dessous nous avons remarqué que la reconstruction de l'image par l'approches 2D-PCA est meilleur que celui l'PCA dont l'erreur de projection de 2D-PCA (5,05%) et plus fable par rapport à l'erreur de projection de la PCA (16 ,34%).



Figure. 4.11 la reconstruction d'une image de test par 2DPCA



Figure.4.12 la reconstruction d'une image test avec PCA

5. Présentation de l'application

On présente dans cette section les différents aspects du notre système de reconnaissance.

a. Accueil

La première interface contient deux boutons :

- ☺ Suivant : pour démarrer le programme.
- ☺ Sortir : pour quitter le programme.

Pour lancer le programme on clique sur « Suivant »



Figure 4.13 Fenêtre d'accueil

b. Interface principale

L'interface principale contient les différentes fonctionnalités de notre système comme le prétraitement de la base de données, les méthodes d'extraction et de classification.

On peut choisir la base de donnée qui on utilise et on clique sur le buttons « Sélectionner » pour choisir l'image de test.

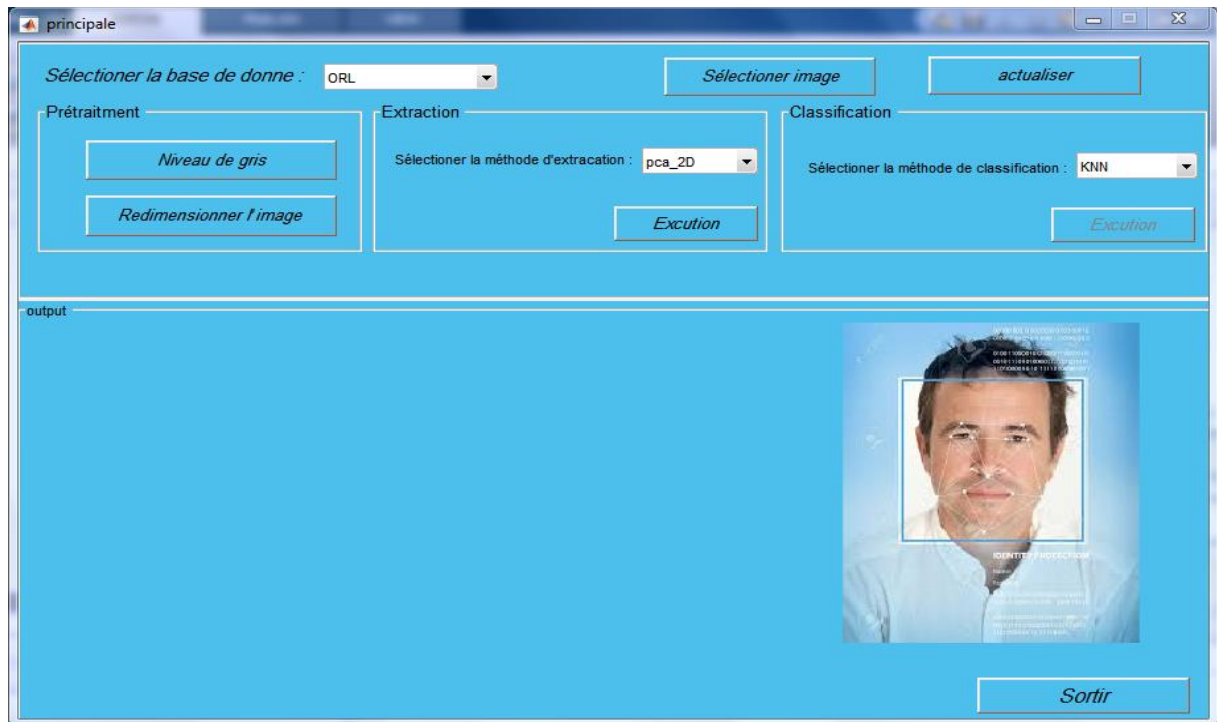


Figure 4.14 Interface principale

- **Prétraitement** : A travers les buttons « Redimensionner l'image » et « Niveau de gris » on peut afficher l'image prétraités



Figure 4.15 Prétraitement d'image

- **Extraction** : pour obtenir les caractéristiques d'une image test on choisit premièrement les méthodes 2D-PCA ou 2D-LDA et on clique sur le buttons « Exécution ».

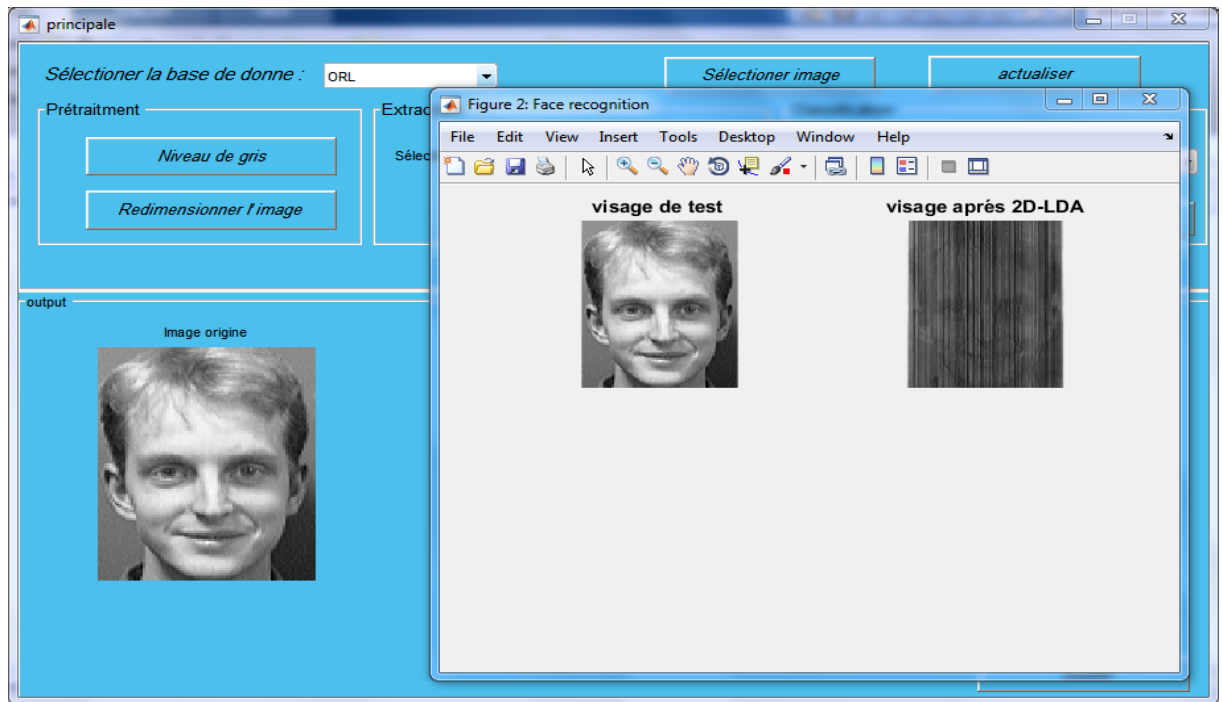


Figure 4.16 Extraction des caractéristiques

- **Classification** : pour classer une image on peut choisi la méthode KNN ou PNN et cliquer sur le buttons « Exécution »

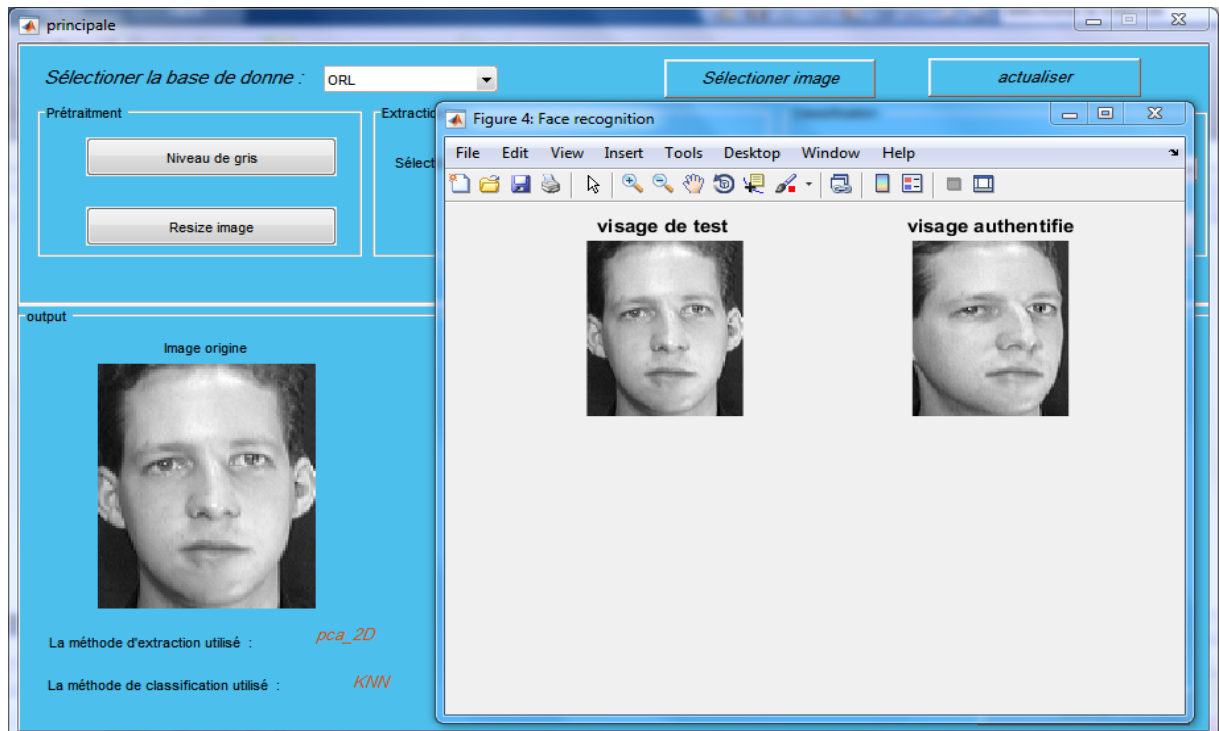


Figure 4.17 Classification d'image

Conclusion

Nous avons présenté, tout au long de ce chapitre, les détails concernant l'implémentation de notre solution. Dans la première section, nous avons présenté les outils matériels et logiciels avec lesquelles nous avons travaillé.

La seconde section a été réservée à l'exhibition des différents blocks et résultats de notre application. notre system a été évalué sur deux base de données ORL et FERET.

Enfin, nous avons exposé et commenté, dans la dernière section, les différentes interfaces composant l'application en GUI MATLAB produit notre de recherche.

.