

Table des matières

Introduction générale.....	1
----------------------------	---

Chapitre 1 : La biométrie

1. Introduction	3
2. La biométrie	9
2.1. Définition	9
2.2 Les systèmes biométriques:	10
2.2.1 Systèmes biométriques d'identification.....	10
2.2.2 Systèmes biométriques de vérification	10
3. Présentation des différents systèmes biométriques	12
3.1. Systèmes Morphologiques	12
3.1.1. Empreintes digitales.....	12
3.1.2. Géométrie de la main / du doigt	6
3.1.3. Iris	13
3.1.4. Rétine	7
3.1.5. Reconnaissance vocale	14
3.1.5. La reconnaissance des visages	14
3.2. Systèmes Comportementales	15
3.2.1. Dynamique des signatures.....	15
3.2.2. Dynamique de la frappe au clavier	16
4.État du marché de la biométrie.....	16
5. Conclusion	17

Chapitre 2 : Techniques de détection et de reconnaissance de visage

1. Introduction	18
2. Détection de visages.....	18
3. Approches de la détection de visage	18
3.1. Approches basées sur l'apparence	18
3.2. Approches basées sur les connaissances acquises.....	19

3.3 Approches basées sur le « Template-matching ».....	20
3.4 Approches basées sur des caractéristiques invariantes	22
3.4.1 Méthodes basées sur les caractéristiques du visage	22
3.4.2. Méthodes basées sur l'analyse de la couleur de la peau.....	22
4. Prétraitement.....	24
4.1 Normalisation Photométrique	24
4.2. Normalisation géométrique	26
5. Les algorithmes de reconnaissance de visage	27
5.1. Analyse en Composantes Principales	27
5.1.1. Présentation	27
5.1.2. Le processus de reconnaissance (Karhunen-Loeve).....	28
5.2. Analyse Discriminante Linéaire	28
5.2.1. Présentation	28
5.3. DF-LDA	29
5.5. Analyse en Composantes Indépendantes	31
5.6. Le réseau de neurones	34
5.6.1. Introduction	34
5.6.2. Le réseau de neurones LVQ	30
5.7. Machine à vecteurs de support	30
5.7.1. Le principe de SVM.....	36
5.7.2. La séparation optimale	36
5.7.3. Le concept d'hyperplan.....	36
5.7.4. La classification linéaire :	37
5.7.5. La classification non linéaire	38
5.8. Modèle de Markov caché	38
5.9. Méthodes basées sur l'apparence locale (Local appearance-based methods). 39	
5.9.1. ACP modulaire	39
5.10. Template matching	40
5.10.1. L'algorithme du template matching	40

6. Performances d'un système de reconnaissances de visage	40
6.1. Performance d'un système d'identification	40
6.2. Performance d'un système de vérification	41
7. Conclusion	43

Chapitre 3:Reconnaissance par ACP (Eigenface)

1. Introduction	44
2. Explication de la méthode de Viola & Jones	44
2.1 Principe	44
2.2 Apprentissage du classifieur	45
2.3 Les caractéristiques :	45
2.4 Les images intégrales	46
2.5 Sélection par boosting	48
3. Présentation de la méthode Eigenface	49
3.1. Création et gestion d'une base de données d'images (Prétraitement)	50
3.2. Convertir les images faciales de la base de données à des vecteurs facials	50
3.2. Normaliser les vecteurs facials.....	51
3.2.1. Calcule l'image moyenne Ψ de toutes les images collectées	51
3.2.2. Calculer les vecteurs propres et les valeurs propres de cette matrice de covariance	52
3.2.3. Choix du nombre de vecteurs propres	54
3.2.4. Utilisation des Eigenfaces pour la classification des images visage.....	56
3.3. Résumé de la méthode.....	58
3.4 : Mesures de Distance :.....	58
3.4.1. Distances Euclidiennes:	59
3.4.2. Distances dans l'Espace de Mahalanobis :	59
3.4.2.1. De l'espace des images à l'espace de Mahalanobis :	59
3.4.2.2. Mahalanobis L1 (MahL1) :	60
3.4.2.3. Mahalanobis L2 (MahL2) :	60
3.4.2.4. Cosinus de Mahalanobis (MahCosine) :	60

3.4.3. Choix de la distance de similarité	61
4. La méthode générale de reconnaissance	61
5. Organigramme détaillé de l'approche Eigenface	62
5.1. Organigramme du prétraitement :	63
4.2. Organigramme de la phase d'apprentissage.....	64
5.3. Organigramme de la phase d'identification :	65
6. Conclusion	66

Chapitre 4 :Test et Resultats

1. Introduction	Error! Bookmark not defined.
2. Environnement du travail	Error! Bookmark not defined.
2.1. Environnement matériel.....	Error! Bookmark not defined.
2.2. Outils de développement.....	Error! Bookmark not defined.
2.2.1. Visual studio 2012.....	Error! Bookmark not defined.
2.2.2.EmguCv.....	Error! Bookmark not defined.
3. Présentation de l'application.....	Error! Bookmark not defined.
3.1. Sous système du prétraitement	Error! Bookmark not defined.
3.1.1. La détection et l'extraire des visages .	Error! Bookmark not defined.
3.1.2. Les paramètres de détection	Error! Bookmark not defined.
3.1.3. Gestion des visages.....	Error! Bookmark not defined.
3.2. Sous système de la reconnaissance facial .	Error! Bookmark not defined.
4. Fonctionnement du système	Error! Bookmark not defined.
4.1. Phase d'apprentissage.....	Error! Bookmark not defined.
4.2. Phase d'identification	70
5. Avantages du système.....	71
6. Conclusion.....	71
Conclusion générale.....	72
Bibliographie.....	73

Introduction générale

De nos jours on parle de plus en plus de l'insécurité dans divers secteurs ainsi que des moyens informatiques à mettre en œuvre pour contrer cette tendance. La vérification et l'identification des individus est l'un des moyens permettant d'assurer cette sécurité. L'être humain se sert quotidiennement de son système visuel pour identifier les personnes de façon automatique, bien que le processus mis en jeu soit complexe.

L'homme a mis en place des moyens de vérification d'identité qui sont liés, soit à ce que possède une personne telle qu'une carte d'identité ou un passeport, soit à ce que sait cette personne, c'est le cas du mot de passe ou un code PIN. Néanmoins, ces éléments peuvent être oubliés, volés ou falsifiés. Pour contourner ces limitations, un autre moyen de sécurité a été développé qui permet d'utiliser, non pas l'information qu'un individu possède ou connaît, mais une information (propre) intrinsèque à cette personne. Cette nouvelle façon d'identification des individus est la biométrie.

L'intérêt principal de la biométrie est donc de reconnaître et d'identifier automatiquement les identités des individus en utilisant leurs caractéristiques physiologiques ou comportementales. Les caractéristiques physiologiques peuvent inclure le visage, l'iris, les empreintes, la géométrie de la main. Les caractéristiques comportementales incluent la voix, la signature, la démarche etc.

Dans ce mémoire, nous nous intéressons à la biométrie du visage qui possède beaucoup d'avantages tels que, la facilité d'utilisation, l'acceptation par l'utilisateur (car elle est non intrusive) et le faible coût. Ainsi, la reconnaissance du visage est déjà intégrée dans des systèmes de sécurité biométriques utilisant un certain nombre d'algorithmes classiques.

La difficulté de la reconnaissance de visage par ordinateur varie énormément suivant que les conditions d'acquisition. Dans un environnement contrôlé, des paramètres tels que l'arrière-plan, la direction et l'intensité des sources lumineuses, l'angle de la prise de vue, la distance de la caméra au sujet sont des paramètres maîtrisés par le système. Dans un environnement non contrôlé, une série de prétraitements sont souvent nécessaires avant de faire la reconnaissance proprement dite.

Les travaux de ce mémoire s'inscrivent dans un contexte global de recherche et développement de d'algorithmes pour la reconnaissance de visages. Ainsi, au cours de ce mémoire, nous aurons à concevoir un système de reconnaissance facial avec la méthode Eigenface qui se base sur une analyse en composante principale. L'ACP est une méthode mathématique qui peut être utilisée pour simplifier un ensemble de données, en réduisant sa dimension. Elle est utilisée pour représenter efficacement les images de visages, qui peuvent être approximativement

reconstruites à partir d'un petit ensemble de poids et d'une image de visage standard.

Nous avons choisi d'articuler notre étude autour de quatre chapitres principaux.

Le premier chapitre présente des généralités sur la biométrie et différents systèmes biométriques et ensuit l'état du marché de la biométrie

Dans le second chapitre une analyse détaillée des différentes techniques développées au cours de ces dernières années dans les domaines de la détection de visage et de la reconnaissance est présentée. Notre objectif est de classifier ces méthodes de manière efficace afin de mettre en évidence les particularités ainsi que les avantages et les inconvénients de chacune d'entre elles. Ceci nous permettra, par la suite, de mettre en avant les critères de performances qui vont guider le choix des solutions retenues dans le cadre de notre problématique. Nous évoquerons aussi la normalisation géométrique des images des visages et les différentes normalisations photométriques.

Dans le troisième chapitre nous présenterons notre algorithme qui se base sur une analyse en composante principale (PCA), qui est une méthode mathématique qui peut être utilisée pour simplifier un ensemble de données, en réduisant sa dimension. Elle est utilisée pour représenter efficacement les images de visages, qui peuvent être approximativement reconstruites à partir d'un petit ensemble de poids et d'une image de visage standard. Nous verrons plusieurs approches pour améliorer les performances de PCA, avec la représentation de la base de données utilisée YALE.

Dans le quatrième chapitre nous présenterons notre interface graphique qui a été créé à l'aide du visual studio c# et avec EMGU CV (est une plate-forme .Net wrapper pour la bibliothèque de traitement d'image de OpenCV), nous analyserons les résultats des tests effectués sur la base de données YALE.

1. Introduction

La reconnaissance de visages est une grande partie de la biométrie et ce chapitre est une introduction au domaine de la biométrie. On donne sa définition, l'architecture d'un système biométrique, les types de systèmes biométriques, les biométries utilisées pour l'authentification des personnes et en fin l'état du marché de la biométrie.

2. La biométrie

2.1. Définition

Le terme biométrie désigne l'ensemble des procédés de reconnaissance automatique d'une personne par certaines de ses caractéristiques physiques. C'est la rencontre entre des techniques numériques, des données biologiques du corps humain et un impératif de la société contemporaine : « identifier facilement et sûrement des personnes » [1].

La question qui se pose c'est : Pourquoi utiliser la biométrie ?

Traditionnellement, pour vérifier l'identité, on utilise surtout des codes, mots de passe, badges, cartes à puce et autres qui présentent plusieurs inconvénients :

- Il faut les mémoriser.
- Il y a un risque d'utilisation par des personnes non autorisées.

L'avantage de l'authentification biométrique est que chaque individu a ses propres caractéristiques physiques qui ne peuvent être changées, perdues ou volées.

La méthode d'authentification biométrique peut aussi être utilisée en complément ou en remplacement des techniques classiques.

Généralement, on distingue deux catégories de méthodes d'authentification biométrique [4] :

- Les méthodes basées sur les caractéristiques comportementales :

La dynamique de la signature (la vitesse de déplacement du stylo, les accélérations, la pression exercée, l'inclinaison), la façon d'utiliser un clavier d'ordinateur (la pression exercée, la vitesse de frappe), la manière de marcher (démarche), etc.

- Les méthodes basées sur les caractéristiques morphologiques :

Empreintes digitales, forme de la main, traits du visage, dessin du réseau veineux de l'œil, etc.

Ces éléments morphologiques ont l'avantage d'être stables dans la vie d'un individu et ne subissent pas autant les effets du stress par exemple, que l'on retrouve dans l'authentification comportementale.

2.2 Les systèmes biométriques:

Un système biométrique est un système d'identification ou de vérification d'individus.

2.2.1 Systèmes biométriques d'identification

Le système répond à la question : « qui est-ce ? » par « c'est la personne X dans ma base de données » (classification 1 parmi N où N est le nombre d'individus dans la base).

Le système évalue le degré de ressemblance entre le modèle test et ceux de la base des modèles d'apprentissage et donne une réponse sous forme des K plus proches (réponse par rang) personnes de la base ou simplement le plus proche du modèle test [2].

2.2.2 Systèmes biométriques de vérification

Le système répond à la question : « êtes-vous vraiment la personne X que vous déclarez ? » par « OUI » ou « NON » (classification 1 parmi 2 donc binaire).

La personne teste qui se présente dans le système se proclame comme étant la personne X de la base, ou bien son modèle peut être stocké sur une carte à puce ou sur un support informatisé. Le système associe simplement le modèle réel (celui capturé) à celui enregistré sur le support. Si le niveau de confiance dépasse un certain seuil, alors l'association est réussie et l'identité est vérifiée [2] [3].

Chaque système biométrique comporte deux phases d'exécution :

- La phase d'apprentissage
- La phase de test.

Chacune contient quatre modules de base plus un module propre à la phase de test.

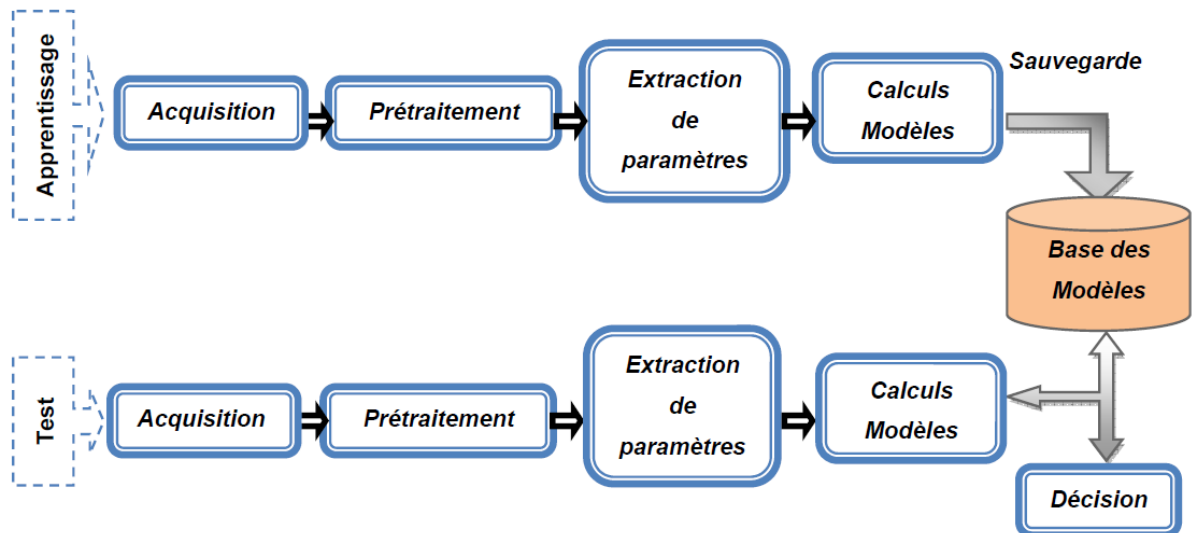


Figure 1.1 : Architecture d'un système biométrique.

Voici les modules de base d'un système biométrique (figure 1.2) [4] [5]:

- **Le module Capture** : pour l'acquisition des données biométriques.
- **Le module Prétraitement** : où les données biométriques sont traitées pour enlever le bruit lié à l'environnement ou au dispositif de capture.
- **Le module Extraction des paramètres** : pour extraire l'information utile contenue dans le signal capturé.
- **Le module Calculs des modèles** : il permet de calculer le modèle biométrique d'un individu suivant l'approche utilisée par le système.
- **Le module Prise de décision** (phase teste) : c'est dans ce module que le système donne sa réponse soit dans une identification par la personne de la base la plus proche, soit par une vérification (oui ou non).

Il existe plusieurs types de caractéristiques physiques, les unes plus fiables que d'autres, mais toutes doivent être :

- discriminantes, pour différencier les personnes sans équivoque.
- invariables, pour assurer leur permanence.
- universelles, pour être appliquées à tout le monde.
- faciles à exploiter.
- acceptables culturellement par les utilisateurs.
- difficilement falsifiables.

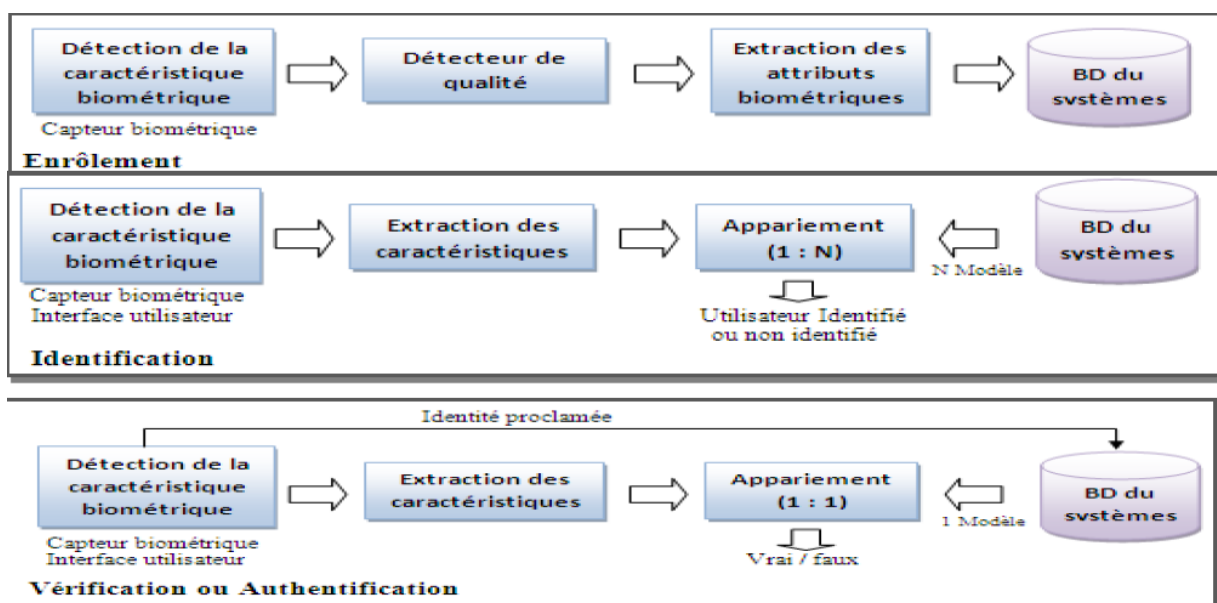


Figure 1.2 : Les différents modules d'un système biométrique [4] [5].

3. Présentation des différents systèmes biométriques

La multitude des caractères biométriques de l'être humain a donné naissance à plusieurs systèmes d'authentification, chacun repose sur un caractère morphologique ou comportemental, parmi ces systèmes il y a ceux qui ont prouvé leur fiabilité et leurs performances et d'autres qui sont toujours en cours d'évolution.

3.1. Systèmes Morphologiques

3.1.1. Empreintes digitales

Il s'agit d'une des premières biométries utilisées dans des machines d'authentification, c'est la technologie la plus mature pour la reconnaissance des criminels.

La formation des empreintes dépend des conditions initiales du développement embryogénique, ce qui les rend uniques à chaque personne et même à chaque doigt.

L'image d'empreinte est prise soit traditionnellement par un scanning du doigt couvert d'encre soit en utilisant un dispositif d'acquisition d'image spécifique [3].

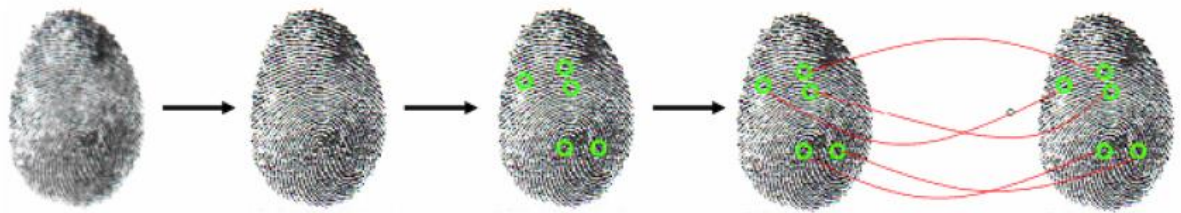


Figure 1.3: Image empreinte.

Les images d'empreintes digitales sont relativement faciles à traiter, car la plupart de l'information est contenue dans la forme des lignes.

Le traitement est généralement effectué sur des images binaires. Autrefois utilisée par les forces de sécurité, cette technique se développe à présent dans le secteur grand-public grâce à l'apparition de capteurs bon marché.

3.1.2. Géométrie de la main / du doigt

Ce type de mesure biométrique est l'un des plus répandus aux États Unis. Il consiste à mesurer plusieurs caractéristiques de la main (jusqu'à 90) telle que la forme de la main, longueur et largeur des doigts, formes des articulations, longueurs inter articulations, ...etc.

La technologie associée à cela est principalement de l'imagerie infrarouge; d'une façon générale, le système présente des FAR (False Acceptance Rate) assez élevés, surtout entre des personnes de la même famille ou bien encore des jumeaux.

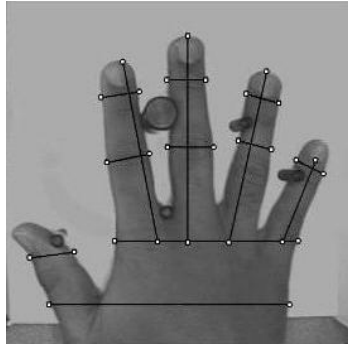


Figure 1.4 : la main.

3.1.3. Iris

Si la couleur, la forme et l'apparence générale de l'iris est déterminée génétiquement, sa texture détaillée est propre à chaque individu, voire même à chaque œil. De plus, cette texture est stable et ne peut être modifiée sans perte importante des capacités visuelles.

Une technique de description de cette texture basée sur des filtres de Gabor a été développée et a conduit à des applications commerciales (la compagnie qui détient les brevets sur la reconnaissance par l'iris s'appelle IriScan).

Cette technique est sûre et moins contraignante pour l'utilisateur que celle de la rétine, mais elle nécessitait jusqu'à présent que l'utilisateur place son œil en bonne position par rapport à la caméra, ce qui a fortement limité son succès en Europe.

Cependant, l'incorporation de techniques de localisation de l'œil permet de relaxer la contrainte à laquelle l'utilisateur est soumis et son utilisation est envisagée dans les distributeurs de billets de banque et pour l'accès sécurisé à Internet.



Figure1.5 : Image iris.

3.1.4. Rétine

Il a été montré que chaque œil possède en sa rétine un arrangement unique des vaisseaux sanguins. La technique basée sur la rétine utilise la texture de ces vaisseaux. Elle est relativement ancienne et a été utilisée essentiellement dans des environnements de haute sécurité, comme l'accès aux sites nucléaires militaires. Cette méthode requiert une collaboration étroite de la part du sujet, car il doit placer son œil extrêmement près de la caméra (quelques centimètres).

Cette caractéristique a limité le développement d'applications grand public.

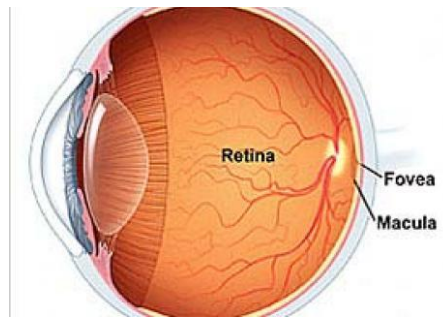


Figure 1.6 : Rétine.

3.1.5. Reconnaissance vocale

La reconnaissance d'un locuteur offre l'avantage d'être bien acceptée par l'utilisateur, quelle que soit sa culture.

De plus, s'il s'agit de sécuriser une transaction téléphonique, la voix est la seule information disponible.

On distingue les systèmes à texte prédéterminé (texte dépendant), où l'utilisateur doit répéter un texte qu'il ne choisit pas et les systèmes où la personne peut parler librement (texte indépendant).

La performance de cette méthode est sujette à la qualité du signal, qui dépend de la variabilité de la voix du locuteur dans le temps comme dans le cas de maladie (rhume), des états émotionnels (angoisse, joie) et de l'âge, des conditions d'acquisition de la voix telles que le bruit et la réverbération, de la qualité des équipements tels que le microphone, sans oublier le fait que différentes personnes peuvent avoir des voix similaires [4] [5].

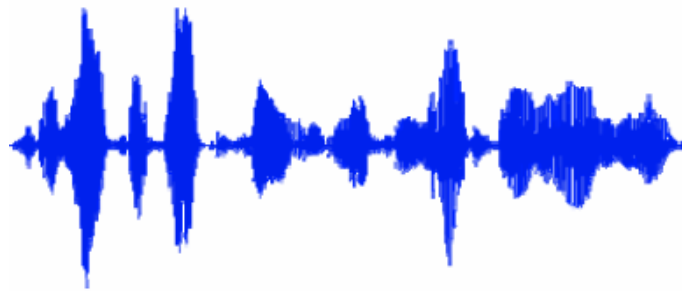


Figure 1.7 : Spectre d'un signal voix.

3.1.5. La reconnaissance des visages

Rien n'est plus naturel qu'utiliser le visage pour identifier une personne. Les images faciales sont probablement la caractéristique biométrique la plus communément employée par l'homme pour effectuer une identification personnelle. L'utilisation d'une caméra permet de capter la forme du visage d'un individu et d'en dégager certaines particularités. Selon le système utilisé, l'individu doit être positionné devant l'appareil ou peut être en mouvement à une certaine distance. Les données biométriques qui sont obtenues sont par la suite comparées

au fichier référence. La méthode la plus utilisée est la méthode Eigenface. Elle utilise une représentation des éléments caractéristiques d'une image de visage à partir d'images modèles en niveau de gris. Des variantes d'Eigenface sont fréquemment utilisées comme base pour d'autres méthodes de reconnaissance comme c'est le cas dans la méthode illustrée ci-dessous.

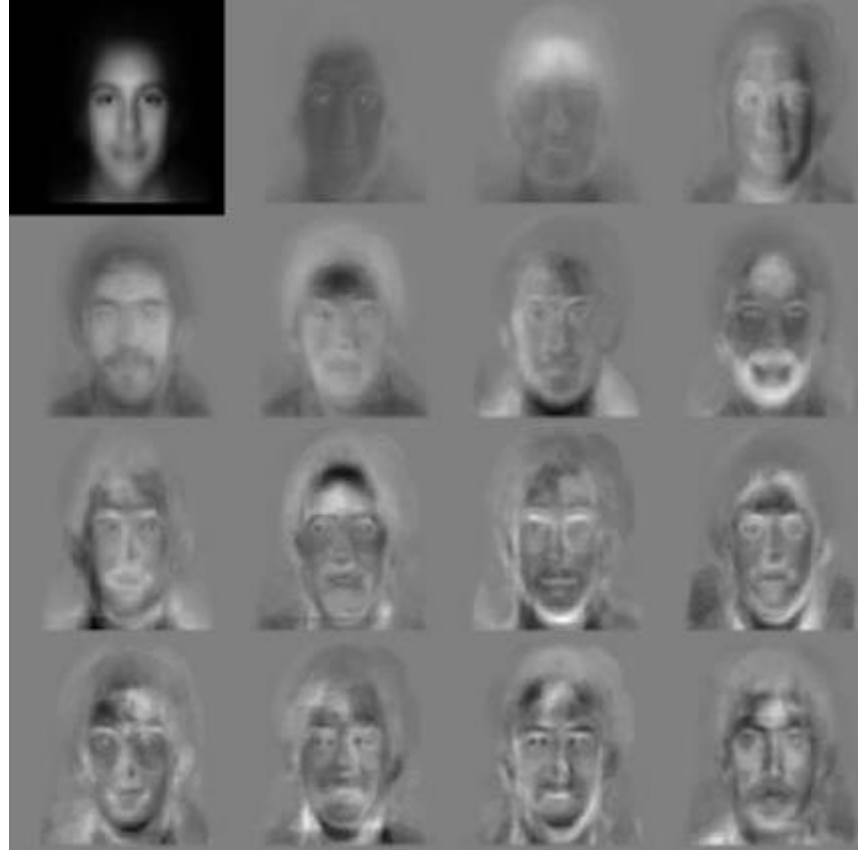


Figure 1.8. La méthode eigenface.

3.2. Systèmes Comportementales

3.2.1. Dynamique des signatures

Ce type de biométrie est à l'heure actuelle peu utilisée, mais ses défenseurs espèrent l'imposer assez rapidement pour des applications spécifiques (documents électroniques, rapports, contrats...). Le procédé est habituellement combiné à une palette graphique (ou équivalent) munie d'un stylo. Ce dispositif va mesurer plusieurs caractéristiques lors de la signature, tel que la vitesse, l'ordre des frappes, la pression et les accélérations, le temps total et tout ce qui peut permettre d'identifier une personne de la façon la plus sûre possible, vu que les données de la signature sont multi changeantes [4] [5].

A handwritten signature in cursive script, likely "Thierry Gauthier", written on a white background.

Figure 1.9 : Signature.

3.2.2. Dynamique de la frappe au clavier

La dynamique de la frappe au clavier est une caractéristique propre de l'individu, c'est en quelque sorte la transposition de la graphologie aux moyens électroniques.

C'est une méthode qui utilise un moyen non intrusif qui exploite un geste naturel, mais elle dépend de l'état physique de la personne (âge, maladies,...).

Les paramètres pris en compte sont généralement la vitesse de frappe, mesure des temps de frappe et pause entre chaque mot [4] [5].



Figure 1.10 : Ecriture au clavier.

4.État du marché de la biométrie

Selon le cabinet américain Frost & Sullivan, se basant sur les sources de l'International Biometric Group, les revenus globaux de la biométrie ont une croissance estimée de 3.4 milliards de dollars en 2009 à 9.4 milliards en 2014.

Ce marché jadis dominé par les besoins gouvernementaux est désormais supplanté de peu par les besoins des secteurs civils. Toujours selon cette même source.

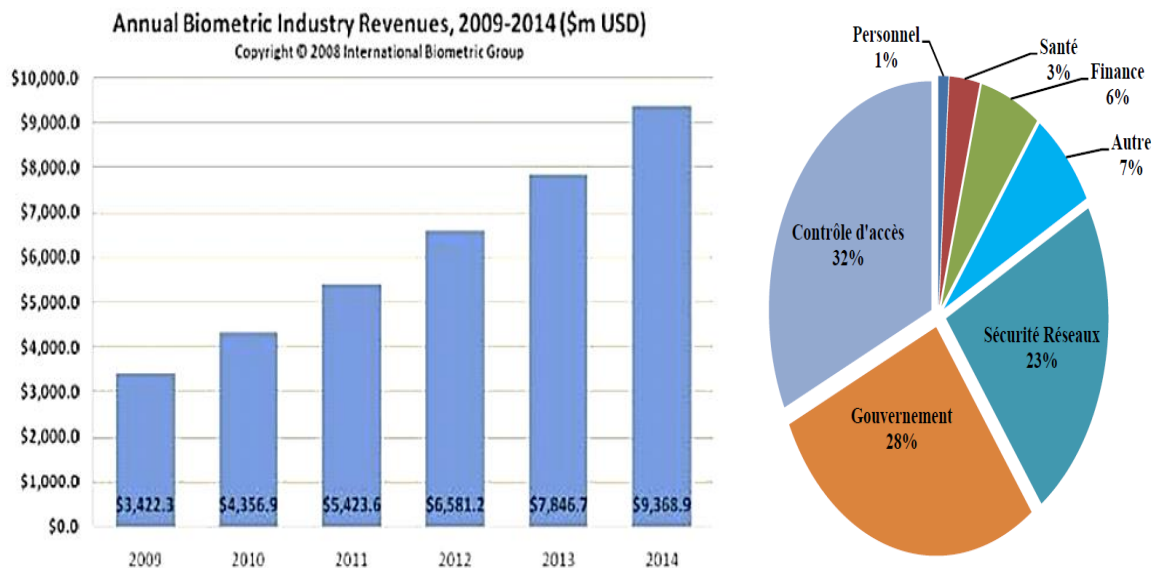


Figure 1.11 : État du marché de la biométrie [3].

5. Conclusion

Aujourd'hui, on utilise la biométrie surtout lorsqu'il s'agit de sécurité mais dans un futur proche, il est presque certain que la biométrie sera présente partout et au quotidien. Elle remplace de plus en plus les autres moyens d'identification comme d'authentification, et ce en raison de sa grande sûreté.

Chapitre II

Techniques de détection et de reconnaissance de visage

1. Préambule

Dans ce chapitre, nous présenterons un état de l'art sur les techniques de détection de visage et les différentes méthodes les plus connues de reconnaissance faciale. Enfin, nous terminerons le chapitre par une synthèse des méthodes et techniques étudiées.

2. Détection de visages

La détection de visages dans l'image est un traitement indispensable et crucial avant la phase de reconnaissance. En effet, le processus de reconnaissance de visages ne pourra jamais devenir intégralement automatique s'il n'a pas été précédé par une étape de détection efficace. Le traitement consiste à rechercher dans une image la position des visages et de les extraire sous la forme d'un ensemble d'images dans le but de faciliter leur traitement ultérieur.

Un visage est considéré correctement détecté si la taille d'image extraite ne dépasse pas 20% de la taille réelle de la région faciale, et qu'elle contient essentiellement les yeux, le nez et la bouche. Elle sera ensuite affinée par un prétraitement [30].

3. Approches de la détection de visage

Il existe plusieurs méthodes pour la détection de visages. Elles peuvent être subdivisées en quatre différents types [30].

3.1. Approches basées sur l'apparence

Ces approches appliquent généralement des techniques d'apprentissage automatique. Ainsi, les modèles sont appris à partir d'un ensemble d'images représentatives de la variabilité de l'aspect facial. Ces modèles sont alors employés pour la détection. L'idée principale de ces méthodes est de considérer que le problème de la détection de visage est un problème de classification (visage, non-visage). Une des approches les plus connues de détection de visage est l'Eigenface [12]. Elle consiste à projeter l'image dans un espace et à calculer la distance euclidienne entre l'image et sa projection. En effet, en codant l'image dans un espace, on dégrade l'information contenue dans l'image, puis on calcule la perte d'information entre l'image et sa projection. Si cette perte d'information est grande (évaluée à partir de la distance, que l'on compare à un seuil fixé a priori), l'image n'est pas correctement représentée dans l'espace : elle ne contient pas de visage.

Cette méthode donne des résultats assez encourageants, mais le temps de calcul est très important. Dans Rowley et al. [31], les auteurs proposent un système de détection de visage basé sur la classification par des réseaux de neurones. Leur technique est divisée en deux étapes: la localisation des visages en utilisant un réseau de neurones et la vérification des résultats obtenus. Les auteurs ont construit un réseau de neurones qui, à partir d'une image prétraitée de 20x20 pixels, indique s'il s'agit d'un visage ou non. Le prétraitement consiste à égaliser l'histogramme. L'image est balayée avec des fenêtres de 20x20. Pour détecter les

visages de différentes tailles, une analyse multi-résolutions est effectuée. L'extension a aussi été proposée afin de déterminer un réseau de neurones indiquant le degré de rotation d'un visage. Ainsi, le système est capable de détecter des visages ayant subi des rotations dans le plan et de différentes échelles. L'un des avantages des réseaux de neurones est leur robustesse au bruit. Malheureusement, les réseaux de neurones, sont souvent difficiles à construire. Leur structure (nombre de couches cachées pour les perceptrons par exemple) influe beaucoup sur les résultats et il n'existe pas de méthode pour déterminer automatiquement cette structure.

La phase d'apprentissage est difficile à mener puisque les exemples doivent être correctement choisis (en nombre et en configuration).

3.2. Approches basées sur les connaissances acquises

Ces méthodes sont basées sur la définition de règles strictes à partir des rapports entre les caractéristiques faciales. Elles s'intéressent aux parties caractéristiques du visage comme le nez, la bouche et les yeux. Ces méthodes sont conçues principalement pour la localisation de visage. Dans [32], Kotropoulous et Pitas utilisent une méthode à base de règles. Les caractéristiques du visage sont localisées à l'aide de la méthode de projection proposée par Kanade [33] pour détecter les contours d'un visage. Soit $I(x,y)$ l'intensité de la luminance du pixel (x,y) de l'image $m*n$, les projections horizontale et verticale de cette image sont définies par l'équation suivante :

$$HI(x) = \sum_{y=1}^n I(x,y) \text{ Et } VI(x) = \sum_{x=1}^m I(x,y) \quad (2.1)$$

Le profil horizontal de l'image originale est calculé en premier. Les deux minimas locaux sont déterminés, ils correspondent aux bords gauche et droit du visage. Ensuite, le profil vertical est à son tour calculé. Les minima locaux de ce profil vertical correspondent aux positions de la bouche, du nez et des yeux. L'inconvénient de cette méthode est qu'elle n'arrive pas à détecter le visage lorsque ce dernier se trouve sur un arrière-plan complexe. Yang and Huang. [34], quant à eux, ont étudié les évolutions des caractéristiques du visage en fonction de la résolution. Quand la résolution de l'image d'un visage est réduite progressivement, par sous-échantillonnage ou par moyenne, les traits macroscopiques du visage disparaissent. Ainsi, pour une résolution faible, la région du visage devient uniforme. Yang et Huang se sont basés sur cette observation pour proposer une méthode hiérarchique de détection de visages.

En commençant par les images à faible résolution, un ensemble de candidats de visage est déterminé à l'aide d'un ensemble de règles permettant de rechercher les régions uniformes dans une image. Les candidats de visage sont ensuite vérifiés en cherchant l'existence de traits faciaux proéminents grâce au calcul des minimas locaux à des résolutions supérieures.

Une caractéristique intéressante de cette technique « descendante » de recherche de zone d'intérêt (informations globales vers des informations plus détaillées) est de réduire le temps de calcul nécessaire par l'utilisation d'images sous-échantillonnées. Malheureusement, cette technique occasionne de nombreuses fausses détections et un taux faible de détection.

3.3 Approches basées sur le « Template-matching »

Les templates peuvent être définis soit "manuellement", soit paramétrés à l'aide de fonctions. L'idée est de calculer la corrélation entre l'image candidate et le template. Ces méthodes rencontrent encore quelques problèmes de robustesse liés aux variations de lumière, d'échelle, etc. Sinha [35] utilise un ensemble d'invariants décrivant le modèle du visage. Afin de déterminer les invariants aux changements de luminosité permettant de caractériser les différentes parties du visage (telles que les yeux, les joues, et le front); cet algorithme calcule ainsi les rapports de luminance entre les régions du visage et retient les directions de ces rapports (par exemple, la région 1 est-elle plus claire ou plus sombre que la région 2). La figure 2.1 montre un modèle prédéfini correspondant à 23 relations. Ces relations prédéfinies sont classifiées en 11 relations essentielles (flèches) et 12 relations confirmations (gris). Chaque flèche représente une relation entre deux régions. Une relation est vérifiée si le rapport entre les deux régions qui lui correspond dépasse un seuil. Le visage est localisé si le nombre de relations essentielles et de confirmation dépasse lui aussi un seuil.

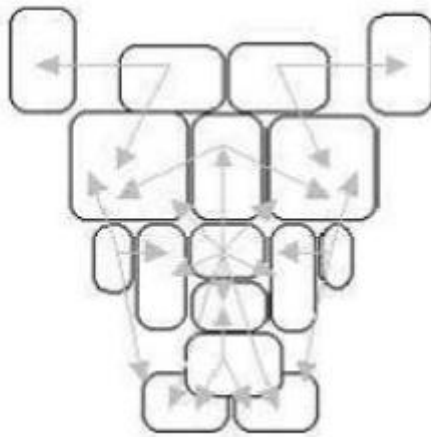


Figure2.1 : Modèle de visage composé de 16 régions (les rectangles) associées à 23 relations (flèches).

Yuille et al. [36] ont utilisé un template déformable pour modéliser les caractéristiques faciales. Ce template adapte un modèle élastique, connu a priori, aux caractéristiques faciales (ex. yeux). Dans cette approche les caractéristiques faciales sont décrites par des templates paramétriques. Une fonction d'énergie est définie pour lier les contours, les sommets et les vallées dans l'image d'entrée aux paramètres correspondants dans le template. Le meilleur ajustement du modèle élastique est trouvé en minimisant une fonction d'énergie des paramètres. Bien que leurs résultats expérimentaux démontrent une bonne performance pour le suivi de caractéristiques non rigides, un inconvénient de cette approche est que le template déformable doit être initialisé dans la proximité de l'objet d'intérêt. Pour détecter les caractéristiques faciales pour la reconnaissance de visage, Brunelli et Poggio. [37] ont utilisé, pour chaque région extraite, un détecteur approprié. Ils se sont aussi inspirés de la méthode de Kanade.



Figure 2.2: Différentes régions utilisées pour la phase de template matching.

Pour les régions yeux, nez et bouche (figure 2.2), ils utilisent la direction du gradient vertical et horizontal. La bouche et le nez sont localisés en utilisant des stratégies similaires. La position verticale est déterminée grâce aux standards anthropométriques. D'abord, une estimation fine de leur position réelle est obtenue en cherchant les pics de la projection horizontale du gradient vertical pour le nez et les vallées de la projection horizontale de l'intensité pour la bouche.

La position des sourcils et leur épaisseur peuvent être trouvées par une analyse similaire. La recherche est une fois encore limitée à la fenêtre d'intérêt, juste au-dessus des yeux, et les sourcils sont trouvés en utilisant la carte du gradient vertical. Le détecteur du sourcil cherche les paires de pics du gradient ayant des directions opposées.

3.4 Approches basées sur des caractéristiques invariantes

Ces approches sont utilisées principalement pour la localisation de visage. Les algorithmes développés visent à trouver les caractéristiques structurales existantes même si la pose, le point de vue ou la condition d'éclairage changent. Puis ils emploient ces caractéristiques invariables pour localiser les visages. Nous pouvons citer deux familles de méthodes appartenant à cette approche : Les méthodes basées sur la couleur de la peau et les méthodes basées sur les caractéristiques de visage qu'on détaillera ci-dessous.

3.4.1 Méthodes basées sur les caractéristiques du visage

En général ces algorithmes utilisent tout d'abord une hypothèse sur la position du haut du visage ensuite l'algorithme de recherche parcourt le visage de haut en bas afin de trouver l'axe des yeux caractérisé par une augmentation soudaine de la densité de contours (mesurée par le rapport noir/blanc le long des plans horizontaux). La longueur entre le haut du visage et le plan de l'œil est alors utilisée comme une longueur de référence pour construire un « template » facial flexible. Ce « template » couvrant des caractéristiques telles que les yeux et la bouche est initialisé à partir de l'image d'entrée. La forme initiale du « template » est obtenue en utilisant la longueur anthropométrique en respectant la longueur de référence.

Le template flexible est alors ajusté par rapport aux positions finales des caractéristiques en utilisant un algorithme de réglage fin qui emploie une fonction de coût basée contour.

Bien que ces algorithmes réussissent à détecter les caractéristiques d'ethnies différentes puisqu'ils ne se basent pas sur les informations de niveaux de gris et de couleur, ils n'arrivent pas cependant à détecter correctement ces caractéristiques si l'image du visage contient des lunettes ou bien si les cheveux couvrent le front.

3.4.2. Méthodes basées sur l'analyse de la couleur de la peau

Les méthodes de détection basées sur l'analyse de la couleur de la peau sont des méthodes efficaces et rapides. Elles réduisent l'espace de recherche de la région visage dans l'image. De plus, la couleur de la peau est une information robuste face aux rotations, aux changements d'échelle et aux occultations partielles. Plusieurs espaces couleur peuvent être utilisés pour détecter, dans l'image, les pixels qui ont la couleur de la peau. L'efficacité de la détection dépend essentiellement de l'espace couleur choisi. Les espaces couleur les plus utilisés sont :

- L'espace RVB, mis au point en 1931 par la Commission Internationale de l'Eclairage (CIE). Il consiste à représenter l'espace des couleurs à partir de trois rayonnements monochromatiques de couleurs : Rouge-Vert-Bleu. Cet espace

correspond à la façon dont les couleurs sont généralement codées informatiquement, ou plus exactement à la manière dont les écrans à tubes cathodiques (ordinateurs, TV) représentent ces couleurs.

- L'espace HSL (Hue, Saturation, Luminance), appelé aussi TSL (Teinte, Saturation, Luminance) en Français, s'appuie sur les travaux du peintre Albert H. Munsell. C'est un modèle de représentation dit "naturel", car il est proche de la perception physiologique de la couleur par l'œil humain. En effet, le modèle RGB aussi adapté soit-il pour la représentation informatique de la couleur ou bien l'affichage sur les périphériques de sortie, ne permet pas de sélectionner facilement une couleur.

Le modèle HSL consiste à décomposer la couleur selon des critères physiologiques :

- la teinte (en Anglais Hue), correspondant à la perception de la couleur.
- la saturation, décrivant la pureté de la couleur, c'est-à-dire son caractère vif ou terne.
- la luminance, indiquant la quantité de lumière de la couleur, c'est-à-dire son aspect clair ou sombre.

Il existe d'autres modèles naturels de représentation proches du modèle HSL :

- HSB : Hue, Saturation, Brightness soit en Français Teinte, Saturation, Brilliance. La brillance décrit la perception de la lumière émise par une surface.
- HSV : Hue, Saturation, Value soit en Français Teinte, Saturation, Valeur.
- HSI : Hue, Saturation, Intensity soit en Français Teinte, Saturation, Intensité.
- HCI : Hue, Chrominance, Intensity soit Teinte, Chrominance, Intensité.
- Le modèle YCrCb est un signal non-linéaire codé à partir du signal RVB. Le paramètre Y représente la luminance (c'est-à-dire l'information en noir et blanc), tandis que Cr et Cb permettent de représenter la chrominance, c'est-à-dire l'information couleur.
- Le modèle colorimétrique $L^*a^*b^*$ (aussi connu sous le nom de CIELab), dans lequel une couleur est repérée par trois valeurs :
- L^* , la luminance, exprimée en pourcentage (0 pour le noir à 100 pour le blanc).
- a^* et b^* deux gammes de couleur allant, respectivement, du vert au rouge et du bleu au jaune avec des valeurs allant de -120 à +120.

Après avoir vu quelques algorithmes de détection de visage, nous passerons à la phase suivante qui est la reconnaissance. Avant d'arriver à cette dernière, nous devons passer par une étape transitoire de prétraitement qui est la normalisation de l'image de visage détecté dans le but de minimiser les effets du bruit engendré durant la chaîne d'acquisition de l'image.

4. Prétraitement

La phase de prétraitement vient après la phase de détection. Elle permet de préparer l'image du visage de telle sorte qu'elle soit exploitable dans la phase d'enrôlement. On l'appelle aussi phase de normalisation puisqu'elle ramène à un format prédéfini toutes les images extraites de l'image brute. Elle consiste généralement en un centrage du visage dans l'image et une élimination des zones non informatives.

Pour garantir la bonne performance du système de reconnaissance de visages, il est important que toutes les images soient de taille identique, à la même échelle et au même format concernant les couleurs (par exemple, les images couleur sont parfois converties en niveaux de gris). Ceci améliore incontestablement le fonctionnement de l'étape d'extraction de signatures et par conséquent la qualité de cette dernière.

La normalisation est constituée de deux processus : géométrique et photométrique. La normalisation géométrique est nécessaire parce que la taille du visage à l'intérieur de l'image acquise peut varier en fonction de la distance entre le module d'acquisition et la personne.

Le visage doit donc être extrait à partir de l'image et une transformation géométrique, pour obtenir une taille fixe, est appliquée. L'approche standard consiste à définir l'emplacement des centres des yeux dans une position constante au sein de l'image de sortie.

L'étape de normalisation photométrique tente d'éliminer ou de réduire les effets de l'illumination de l'image.

4.1 Normalisation Photométrique

Dans le domaine de la reconnaissance par le visage, un certain nombre de méthodes de normalisation de l'illumination ont été présentées. Ces méthodes peuvent être classées en deux grandes catégories : les méthodes basées sur la modification de la dynamique et des méthodes basées sur l'estimation de la réflectance de l'image. Les quatre normalisations présentées ci-dessous sont les plus utilisées dans la reconnaissance du visage et font partie de ces deux catégories.

Il s'agit de :

- L'égalisation d'histogramme : Le but de cette normalisation est de répartir

uniformément les niveaux de gris de l'image afin d'améliorer son contraste.

- La correction Gamma : cette normalisation permet de réduire l'écart entre les zones sombres et claires du visage par une transformation non linéaire des niveaux de gris.
- Le MultiScale-Retinex [94] : L'objectif de cette méthode est d'estimer la partie luminance de l'image pour en déduire la réflectance.
- Le lissage anisotropique [26]: cette approche estime aussi la luminance de l'image mais en se basant sur l'anisotropie de la réflectance.

Nous avons opté pour la méthode de l'égalisation d'histogramme qui a amélioré le taux d'identification de notre système de reconnaissance.

- **Égalisation d'histogramme**

Cette normalisation appartient à la catégorie des méthodes basées sur l'amélioration de la dynamique de l'image. Le but de l'égalisation est d'harmoniser la répartition des niveaux de gris de l'image. Elle permet de tendre vers un même nombre de pixels pour chacun des niveaux de l'histogramme. Cette opération vise à augmenter les nuances dans l'image et donc son contraste (Figure2.4).

Plus concrètement, si n_i est le nombre de pixels à un niveau i de gris, la probabilité qu'un pixel x de l'image ait un niveau i est :

$$P(x_i) = \frac{n_i}{n}, \quad i \in 0, \dots, L \quad (2.2)$$

Avec n le nombre total des pixels de l'image et L le nombre des niveaux de gris. La probabilité P représente dans ce cas l'histogramme de l'image normalisée à $[0; 1]$. Soit c la distribution cumulative de l'histogramme normalisé P , elle est donnée par :

$$C(i) = \sum_{j=0}^i P(x_j) \quad (2.3)$$

L'idée est de trouver une transformation $y = T(x)$ qui, pour chaque niveau x de l'image, produira un niveau y de telle façon que la distribution cumulative des différents niveaux de l'image transformée soit linéaire. Cette fonction est définie par :

$$y_i = T(x_i) = L \frac{C_i}{n} \quad (2.4)$$

Avec L le niveau maximal de gris.



Figure 2.3 : (a) Image RGB (b) Image niveau de gris (c) égalisation d'histogramme

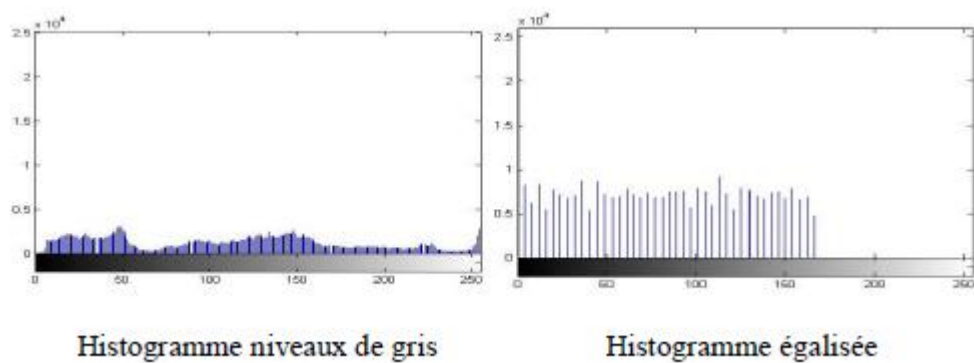


Figure 2.4: Histogramme de l'image avant et après égalisation

4.2. Normalisation géométrique

En utilisant un algorithme de reconnaissance se basant sur la réduction de l'espace, nous ne pouvons pas négliger un point très important qui est la normalisation géométrique des images de visage. Cette normalisation géométrique consiste à extraire la zone du visage de l'image originale, ensuite une rotation du visage est effectuée afin d'aligner l'axe des yeux avec l'axe horizontal. Enfin, une réduction proportionnelle à la distance entre les centres des deux yeux est appliquée. On obtient alors une image de visage dont la distance entre les centres des yeux est fixe.

Les dimensions de l'image du visage sont calculées à partir de la distance à obtenir entre les centres des deux yeux.

5. Les algorithmes de reconnaissance de visage

Dans cette partie de ce chapitre, on décrit quelques algorithmes classiques de reconnaissance de visages, notamment : PCA, LDA, LVQ, SVM, HMM, Template matching, EGM...

5.1. Analyse en Composantes Principales

5.1.1. Présentation

L'analyse en composantes principales (ACP) est une méthode mathématique qui peut être utilisée pour simplifier un ensemble de données, en réduisant sa dimension.

Elle est utilisée pour représenter efficacement les images de visages, qui peuvent être approximativement reconstruites à partir d'un petit ensemble de poids et d'une image de visage standard.

Ces poids sont obtenus en projetant l'image dans un espace de visage engendré par les visages propres (Eigenfaces). Ainsi au lieu de stocker un vecteur d'image de taille Résolution R , on stocke un vecteur de taille N nombre d'image d'apprentissage $M \ll R$. Dans l'approche ACP, la normalisation d'éclairage est toujours indispensable [38]. Les visages propres sont des images de la même taille que les images d'apprentissage et qui montrent des visages ayant un aspect fantomatique. Mathématiquement, elles sont les composantes principales de la distribution des visages, ou les vecteurs propres (Eigenvectors) de la matrice de covariance de l'ensemble des images de visage. Chaque image de visage de l'ensemble d'apprentissage peut être exactement représentée en termes de combinaison linéaire des Eigenfaces et du visage moyen [39].

Le nombre possible d'Eigenfaces est égal au nombre d'images de visage dans l'ensemble d'apprentissage. Néanmoins les visages peuvent être approximés, en utilisant seulement les meilleurs Eigenfaces (ayant les plus larges valeurs propres qui représentent en fait, la plupart de *variance* dans l'ensemble d'images de visage) qui définissent le "Low Dimensional Space", permettant ainsi de diminuer les calculs. Dans le cas où on ne prend que les N ($N < M$) grandes valeurs propres (Eigenvalues) et les vecteurs propres qui leur sont associés, on réduit encore plus la mémoire de stockage et le temps de calcul, mais on réduit aussi les performances dû à la négligence d'une partie de l'information. C'est pour cela qu'on a pris dans notre application $N = M$, bénéficiant à la fois d'une réduction de l'espace ($R \rightarrow M$) acceptable, et aussi en ayant la totalité de l'information, garantissant ainsi les performances optimales. Une variante de l'ACP, plus utilisée dans la communauté du traitement du signal est connue sous le nom de transformée de Karhunen-Loève. L'ACP est une technique rapide, simple et populaire dans l'identification de modèle, c'est l'une des meilleures techniques. Les

projections de l'ACP sont optimales pour la reconstruction d'une base de dimension réduite. Cependant, l'ACP n'est pas optimisé pour la séparabilité (discrimination) de classe. Une alternative qui est l'analyse discriminante linéaire LDA tient compte de ceci.

5.1.2. Le processus de reconnaissance (Karhunen-Loeve)

Au début de la phase d'apprentissage, on fait l'acquisition, la lecture et la normalisation des images d'apprentissage (de taille R). Puis on calcule le visage moyen de ces images. On soustrait ensuite le visage moyen de chaque image d'apprentissage normalisée (ces nouvelles images représentent les colonnes d'une matrice S). On fait le calcul de la matrice de covariance qui est définie par la formule $L=S'S$. On calcule ensuite les vecteurs propres V et les valeurs propres D de la matrice L . Puis on passe à l'étape de calcul des visages propres selon la formule $U = S * V * (abs(D))^{-0.5}$. Et finalement on calcule les poids des visages de la base (de taille M) en les projetant dans le sous-espace engendré par les visages propres "Face Space".

Pendant la phase de vérification, on fait l'acquisition, la lecture et la normalisation de l'image de vérification (de taille R). Puis on soustrait le visage moyen (calculé auparavant lors de la phase d'apprentissage) de l'image de vérification normalisée. Ensuite on calcule le poids de l'image (de taille M) en utilisant les visages propres comme une base de projection. Et finalement on utilise la distance euclidienne comme une mesure de similarité [41].

5.2. Analyse Discriminante Linéaire

5.2.1. Présentation

L'analyse discriminante linéaire (LDA) et les discriminants linéaires de Fisher (*FLD*) relatifs, sont utilisés pour trouver la combinaison linéaire des caractéristiques qui séparent le mieux les classes d'objet ou d'événement.

Les combinaisons résultantes peuvent être employées comme classificateur linéaire, ou généralement dans la réduction de caractéristiques avant la classification postérieure.

LDA est étroitement lié à l'ACP (*PCA en anglais*), du fait que tous les deux recherchent des combinaisons linéaires des variables qui représentent au mieux les données.

LDA essaye explicitement de modéliser la différence entre les classes des données. *PCA* quant à elle, ne tient pas compte des différences entre les classes.

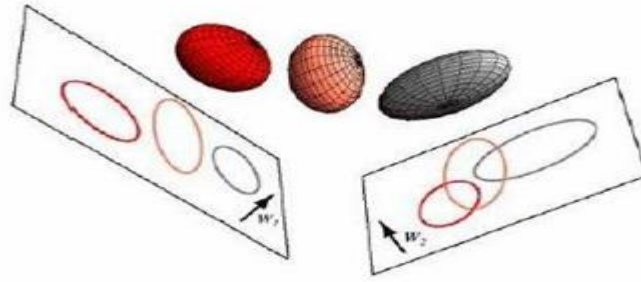


Figure 2.5 : Illustration du principe de séparation optimale des classes par le LDA.

Trois distributions 3D sont projetées sur deux sous-espaces 2D décrits par les vecteurs W_1 et W_2 . Puisque le LDA essaye de trouver la plus grande séparation parmi les classes, on voit bien que W_1 est ici le vecteur optimal.

Chaque visage, qui se compose d'un grand nombre de Pixel, est réduit à un plus petit ensemble de combinaisons linéaires avant la classification

Chacune des nouvelles dimensions est une combinaison linéaire des valeurs de pixel, qui forment un *template*. Les combinaisons linéaires obtenues en utilisant *FLD* s'appellent les *Fisherfaces*, en analogie avec les *Eigenfaces* [42]. *LDA* est une technique qui cherche les directions qui sont efficaces pour la discrimination entre les données.

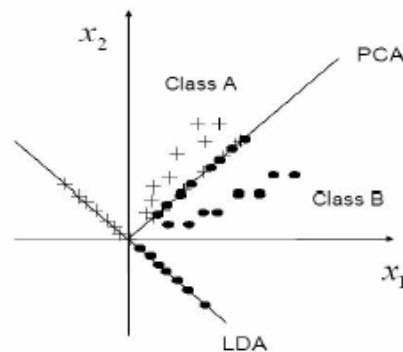


Figure 2.6 : Les projections PCA et LDA d'un ensemble de données.

LDA est plutôt connue pour sa maximisation de l'éparpillement d'inter-classe "*the between-class scatter*" et sa réduction au minimum de l'éparpillement d'intra-classe "*the within-class scatter*", qui se manifeste par le groupement des vecteurs de poids de la même classe (faible distance entre ces vecteurs), et par la séparation des vecteurs de poids de classes différentes (grande distance entre ces vecteurs) [43].

5.3. DF-LDA

La plupart des méthodes traditionnelles basées sur *LDA*, souffrent de l'inconvénient que leurs critères d'optimalité ne sont pas directement liés aux capacités de classification de la

représentation obtenue des caractéristiques.

D'ailleurs, leur exactitude de classification est affectée par le problème de la petite dimension de l'échantillon (« SSS », small sample size) qui est souvent rencontré en reconnaissance de visages. L'algorithme de DF-LDA "DIRECT FRACTIONAL-STEP LDA" traite ces deux imperfections d'une façon efficace et rentable, offrant des performances supérieures à celles des techniques de reconnaissance de visages classiques.

L'algorithme de *DF-LDA* est représenté comme suit [44]:

- L'acquisition et la lecture des images d'apprentissage

$$\begin{array}{cccc}
 \begin{array}{c} \text{Image 1} \\ = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{N^2} \end{pmatrix} \end{array} &
 \begin{array}{c} \text{Image 2} \\ = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{N^2} \end{pmatrix} \end{array} &
 \begin{array}{c} \text{Image 3} \\ = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N^2} \end{pmatrix} \end{array} &
 \begin{array}{c} \text{Image 4} \\ = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N^2} \end{pmatrix} \end{array} \\
 \\
 \begin{array}{c} \text{Image 5} \\ = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{N^2} \end{pmatrix} \end{array} &
 \begin{array}{c} \text{Image 6} \\ = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N^2} \end{pmatrix} \end{array} &
 \begin{array}{c} \text{Image 7} \\ = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_{N^2} \end{pmatrix} \end{array} &
 \begin{array}{c} \text{Image 8} \\ = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_{N^2} \end{pmatrix} \end{array}
 \end{array}$$

- Normalisation des images
- Calcul du visage moyen m
- Calcul du visage moyen de chaque classe

$$\begin{aligned}
 \bar{x} &= \frac{1}{2} \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_{N^2} + b_{N^2} \end{pmatrix}, & \bar{y} &= \frac{1}{2} \begin{pmatrix} c_1 + d_1 \\ c_2 + d_2 \\ \vdots \\ c_{N^2} + d_{N^2} \end{pmatrix}, \\
 \bar{z} &= \frac{1}{2} \begin{pmatrix} e_1 + f_1 \\ e_2 + f_2 \\ \vdots \\ e_{N^2} + f_{N^2} \end{pmatrix}, & \bar{w} &= \frac{1}{2} \begin{pmatrix} g_1 + h_1 \\ g_2 + h_2 \\ \vdots \\ g_{N^2} + h_{N^2} \end{pmatrix}
 \end{aligned} \tag{2.5}$$

- Soustraction des visages d'apprentissage

$$\begin{aligned}
\vec{a}_m &= \begin{pmatrix} a_1 & - & x_1 \\ a_2 & - & x_2 \\ \vdots & & \vdots \\ a_{N^2} & - & x_{N^2} \end{pmatrix}, \vec{b}_m = \begin{pmatrix} b_1 & - & x_1 \\ b_2 & - & x_2 \\ \vdots & & \vdots \\ b_{N^2} & - & x_{N^2} \end{pmatrix}, \vec{c}_m = \begin{pmatrix} c_1 & - & y_1 \\ c_2 & - & y_2 \\ \vdots & & \vdots \\ c_{N^2} & - & y_{N^2} \end{pmatrix} \\
\vec{d}_m &= \begin{pmatrix} d_1 & - & y_1 \\ d_2 & - & y_2 \\ \vdots & & \vdots \\ d_{N^2} & - & y_{N^2} \end{pmatrix}, \vec{e}_m = \begin{pmatrix} e_1 & - & z_1 \\ e_2 & - & z_2 \\ \vdots & & \vdots \\ e_{N^2} & - & z_{N^2} \end{pmatrix}, \vec{f}_m = \begin{pmatrix} f_1 & - & z_1 \\ f_2 & - & z_2 \\ \vdots & & \vdots \\ f_{N^2} & - & z_{N^2} \end{pmatrix} \\
\vec{g}_m &= \begin{pmatrix} g_1 & - & w_1 \\ g_2 & - & w_2 \\ \vdots & & \vdots \\ g_{N^2} & - & w_{N^2} \end{pmatrix}, \vec{h}_m = \begin{pmatrix} h_1 & - & w_1 \\ h_2 & - & w_2 \\ \vdots & & \vdots \\ h_{N^2} & - & w_{N^2} \end{pmatrix}
\end{aligned} \tag{2.6}$$

- La construction des scatter matrices (une matrice par classe) :

$$S_1 = (\vec{a}_m \vec{a}_m^T + \vec{b}_m \vec{b}_m^T), S_2 = (\vec{c}_m \vec{c}_m^T + \vec{d}_m \vec{d}_m^T) \tag{2.7}$$

$$S_3 = (\vec{e}_m \vec{e}_m^T + \vec{f}_m \vec{f}_m^T), S_4 = (\vec{g}_m \vec{g}_m^T + \vec{h}_m \vec{h}_m^T)$$

- Le calcul de la *within-class scatter matrix* SW:

$$S_w = S_1 + S_2 + S_3 + S_4 \tag{2.8}$$

- Le calcul de la *weighted between-class scatter matrix* :

$$\hat{S}_{BTW} = \sum_{i=1}^C \phi_i \phi_i^T \tag{2.9}$$

$$\text{Avec : } \phi_i = (L_i/L)^{1/2} \sum_{j=1}^i (W(d_{ij}))^{1/2} (Z_i - Z_j) \tag{2.10}$$

Est la moyenne de la classe Z_i ,

$L_i = \dim(Z_i)$, L est le nombre d'image d'apprentissage, C est le nombre de classe, (d_{ij}) est la distance euclidienne entre les moyennes des classes i et j , La fonction pesante $G(d_{ij})$ est une fonction monotoniquement décroissante de la distance d_{ij}

(Il est recommandé d'utiliser : $G(d_{ij}) = (d_{ij})^{-2p}$ avec $p=2,3,\dots$)

$$\text{Le calcul de la matrice : } S_{TOT} = S_{WTH} + \hat{S}_{BTW} \tag{2.11}$$

La métrique utilisée dans le *DF-LDA* est :

$$\Psi = \arg \max_{\Psi} \frac{|\langle \Psi^T S_{BTW} \Psi \rangle|}{|\langle \Psi^T S_{TOT} \Psi \rangle|} \tag{2.12}$$

5.5. Analyse en Composantes Indépendantes

PCA est une technique optimale de recherche de représentation réduite qui minimise l'erreur de reconstruction, cependant les vecteurs de base tenant compte de l'erreur de reconstruction peuvent ne pas être optimaux pour coder l'information appropriée de l'image à la classification. L'*analyse en composantes indépendantes (ICA)* est une

généralisation de *PCA* qui utilise en plus des statistiques d'ordre 2, les statistiques d'ordre plus supérieur, ce qui peut produire une représentation de données plus puissante.

Le but d'*ICA* est de trouver des vecteurs de base (images de base) localisés dans l'espace et qui sont statistiquement indépendants, en réduisant au minimum la dépendance statistique.

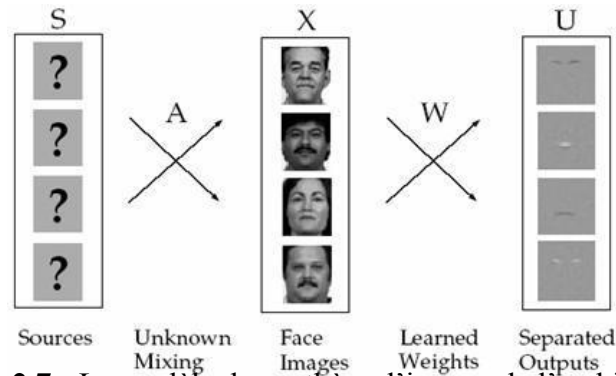


Figure 2.7 : Le modèle de synthèse d'image de l'architecture 1 d'ICA.

Avant l'apprentissage, on soustrait l'image moyenne des images d'apprentissage X et on les filtre par un filtre de blanchiment (whitening filter)

$$W_Z = 2(XX^T)^{-1/2} \quad (2.13)$$

Pour supprimer les statistiques d'ordre 1 et 2 (la moyenne et les covariances sont mises à zéro et les variances sont égalisées). Pour trouver un ensemble d'images composantes indépendantes, les images d'apprentissage X sont considérées comme étant une combinaison linéaire d'images de base statistiquement indépendantes S , où A est une matrice de mixage inconnue.

Les images de base sont retrouvées par une matrice de filtre d'apprentissage $W_i = W \times W_z$, qui produit des sorties U statistiquement indépendantes. Ces images de base sont considérées comme un ensemble de caractéristiques faciales statistiquement indépendantes, où les valeurs des pixels de chaque image caractéristique sont statistiquement indépendantes des valeurs des pixels des autres images caractéristiques [45].

$$\text{Face Image} = b_1 * u_1 + b_2 * u_2 + \dots + b_n * u_n$$

The diagram shows a grayscale face image on the left, followed by an equals sign, and then a sum of terms. Each term consists of a coefficient (b_1, b_2, \dots, b_n) multiplied by a small grayscale image (u_1, u_2, \dots, u_n) representing an independent component. The images u_i show isolated facial features like eyes, nose, and mouth.

Figure 2.8 : La représentation d'ICA = (b_1, b_2, \dots, b_n)

La représentation d'*ICA* est constituée des coefficients b de la combinaison linéaire des images de base U qui forment chaque image de visage. Le nombre de composantes indépendantes extraites par l'algorithme d'*ICA* est égal à la dimension de l'entrée [45] (le

nombre d'images d'apprentissage), et si on veut contrôler ce nombre, au lieu d'exécuter *ICA* sur les n images originales, on l'exécute sur un ensemble de m combinaisons linéaires de ces images, où $m < n$.

Communément, on choisit pour ces combinaisons linéaires un sous ensemble formé des premières composantes principales de la matrice de donnée, ainsi l'exécution d'*ICA* sur l'ensemble de ces m composantes principales produit m images sources U_i statistiquement indépendantes.

Dans cette première architecture d'*ICA*, les images de base U_i sont statistiquement indépendantes, mais les coefficients b_i ne le sont pas. Au lieu de séparer les images de visages en ensembles d'images indépendantes, il existe une deuxième architecture d'*ICA*, dans laquelle les coefficients sont statistiquement indépendants, en séparant les pixels en un ensemble de variables indépendantes codant les images.

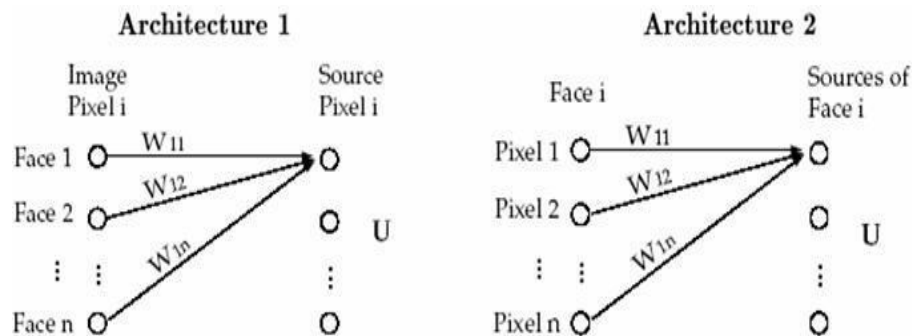


Figure 2.9 : Les deux architectures d'ICA : l'architecture 1 pour trouver des images de base statistiquement indépendantes (la séparation des images de visages produit des images composantes indépendantes), et l'architecture 2 pour trouver un code factoriel (la séparation des pixels produit un code factoriel).

$$\text{Target Face Image} = u_1 * a_1 + u_2 * a_2 + \dots + u_n * a_n$$

The figure shows a target grayscale face image on the left. To its right is an equals sign followed by a sum of terms. Each term consists of a coefficient (u_1, u_2, \dots, u_n) multiplied by a small grayscale face image (a_1, a_2, \dots, a_n). The images a_i represent the basis images.

Figure 2.10 : Le modèle de synthèse d'image de l'architecture 2 d'ICA.

Dans l'architecture 2 d'*ICA*, chaque image de l'ensemble de données est considérée comme étant créée à partir d'un ensemble d'images de base en A , et d'un vecteur de causes d'images "image causes" fondamentales statistiquement indépendantes en S . L'algorithme d'*ICA* essaye d'inverser les images de base en trouvant un ensemble de filtres W_i qui produisent des sorties statistiquement indépendantes.

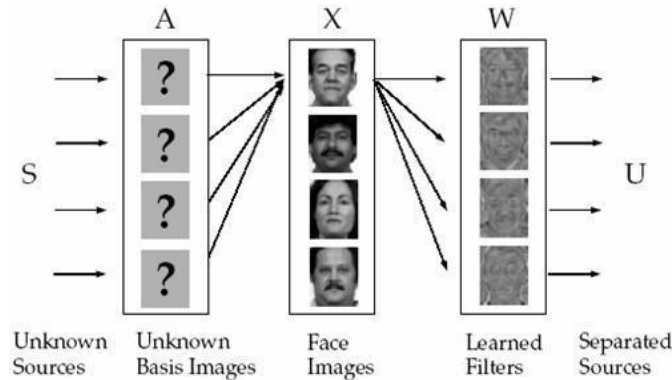


Figure 2.11 : La représentation factorielle d'ICA = (u_1, u_2, \dots, u_n) .

5.6. Le réseau de neurones

5.6.1. Introduction

L'application des réseaux de neurones artificiels dans la reconnaissance de visages a visé plusieurs problèmes : la classification Homme Femme, la reconnaissance de visages et la classification des expressions faciales [46]. Un réseau de neurones est un système de traitement de l'information qui a été développé comme généralisations des modèles mathématiques assortissant la connaissance humaine. Ils se composent d'un grand nombre d'unités de traitement hautement reliées appelées neurones, travaillant ensemble pour exécuter une tâche de classification donnée.

Il est un processeur parallèle distribué, ayant une prospérité naturelle pour stocker une connaissance expérimentale. Il ressemble au cerveau humain en trois aspects : la connaissance est acquise par le réseau par un processus d'apprentissage, des forces de connexion reliées ensemble, connues sous le nom de poids synaptiques, sont employées pour stocker la connaissance, et chaque neurone a un état interne appelé seuil ou fonction d'activation (ou fonction de transfert) utilisée pour classifier les vecteurs.

Cottrell et al. [51] ont cherché des méthodes d'analyse automatique du visage les plus proches possible de la réalité biologique. Ainsi, un réseau de neurones dit auto-supervisé, c'est-à-dire dont la couche d'entrée et la couche de sortie sont identiques et égales à l'image d'un visage, effectue une analyse en composantes principales.

Le nombre de composantes principales est donné par le nombre de neurones de la couche cachée. Chaque neurone de la couche cachée correspond aux valeurs propres de la décomposition. Lawrence et al. [52] choisissent de classer les vecteurs de caractéristiques, extraites par cartes de Kohonen, à l'aide d'un réseau de neurones convolutionnel. Ce type de réseau de neurones est partiellement invariant à des transformations globales telles que la translation, la rotation et les changements d'échelle.

Les résultats expérimentaux ont montré la supériorité des réseaux de neurones convolutionnels sur les réseaux auto-supervisés de Cottrell [51].

Lin et al. [53] ont utilisé un réseau de neurones à décision probabiliste alliant ainsi les avantages des approches statistiques et des réseaux de neurones. Il a été montré que les performances de cette solution sont comparables à la méthode de Lawrence et al. , tout en étant beaucoup moins coûteux en termes de temps de calcul.

On trouve plusieurs types de réseaux de neurones : l'apprentissage compétitif (non-supervisé, parfois appelé *VQ*), le *LVQ* (apprentissage compétitif supervisé), les cartes auto organisatrices, le perceptron multicouches avec apprentissage par rétro propagation, les réseaux probabilistes, les réseaux à fonctions radiales de base (*FRB*), et les réseaux récurrents [47].

Une classification par réseau de neurones comporte les étapes suivantes :

Tout d'abord une phase de prétraitement des images d'apprentissage et l'association à chaque image d'apprentissage (entrée du réseau) un vecteur de sortie, puis vient l'étape d'initialisation (création des couches du réseau). On fait l'apprentissage (supervisé) du réseau, jusqu'à atteindre une certaine erreur minimale (le réseau apprend à bien classifier les images d'apprentissage). On présente ensuite au réseau une nouvelle image à identifier (phase de reconnaissance ou de simulation ou d'activation du réseau) qui sera finalement affectée à une classe donnée [48].

5.6.2. Le réseau de neurones LVQ

Parmi tous les types de réseau de neurones qui existent, le type *LVQ* est le type le plus adapté à l'application de reconnaissance de visage, puisqu'au contraire des autres types qui donnent des performances basses, le *LVQ* quand à lui réalise de hautes performances de classification [47].

L'algorithme de base *LVQ* :

- application d'une entrée
- recherche du neurone vainqueur (sa sortie est égale à 1)
- modification des poids (rapprochement ou éloignement selon que le neurone vainqueur est le neurone désiré ou non)

5.7. Machine à vecteurs de support

Une machine à vecteurs de support est une technique de discrimination, c'est une méthode d'apprentissage supervisé utilisée pour la classification et la régression. Elle consiste à séparer deux ou plusieurs ensembles de points par un hyperplan. Selon les cas et la configuration des points, la performance de la machine à vecteurs de support peut être

supérieure à celle d'un réseau de neurones ou d'un modèle de mixture gaussienne [49].

5.7.1. Le principe de SVM

L'idée originale des SVMs est basée sur l'utilisation de fonctions noyau kernel, qui permettent une séparation optimale des points du plan en différentes catégories. La méthode fait appel à un ensemble de données d'apprentissage, qui permet d'établir un hyperplan séparant au mieux les points [49].

5.7.2. La séparation optimale

Dans un plan à deux dimensions dans lequel sont répartis deux groupes de points associés à deux classes : les points (+) pour $y > 0$ et les points (-) pour $y < 0$, on peut trouver un séparateur linéaire évident, qui est l'axe des abscisses et qui sépare parfaitement les points. Le problème est dit *linéairement séparable*. Pour des problèmes plus compliqués, la caractérisation d'un séparateur linéaire peut être très compliquée et tout à fait non optimale. Par exemple dans un plan dans lequel les points (+) sont regroupés en un cercle, avec des points (-) tout autour, aucun séparateur linéaire en deux dimensions ne pourra correctement séparer les groupes. Le problème dans ce cas n'est pas *linéairement séparable*. La SVM utilise les *trucs du noyau* “kernel trick” pour appliquer une classification linéaire à un problème de classification non linéaire [49].

5.7.3. Le concept d'hyperplan

Afin de remédier au problème de l'absence de séparateur linéaire, le concept des SVMs est de reconsidérer le problème dans un espace de dimension supérieure. Dans ce nouvel espace, il existe un séparateur linéaire qui permet de classer au mieux les points dans les deux groupes qui conviennent, avec une distance maximale entre l'hyperplan et les points les plus proches des deux classes, afin de pouvoir classifier plus exactement les nouveaux points, cette distance s'appelle la *marge*. Cet hyperplan est connu sous le nom de l'*hyperplan à marge maximum* ou l'*hyperplan optimal* et les vecteurs (points de dimension n) les plus proches de cet hyperplan s'appellent les *vecteurs de support*. On pourra ensuite projeter le séparateur linéaire dans l'espace d'origine pour visualiser le résultat de la classification. Le séparateur linéaire obtenu peut être un hyperplan, c'est à dire la généralisation à n dimensions d'une ligne (1D) séparant un espace 2D, ou d'un plan (2D) séparant un espace 3D. Le changement d'espace se fait au moyen d'une fonction répondant au critère de Mercer. Ce critère permet un changement dans les deux sens, ce qui permet à partir de l'expression de l'hyperplan dans l'espace complexe de classer les éléments dans l'espace de description initial [49].

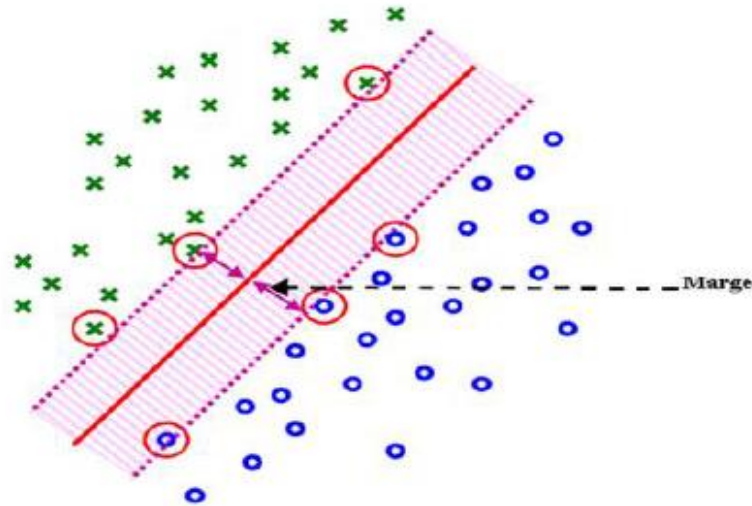


Figure 2.12 : Exemple de deux classes linéairement séparables. L'hyperplan déterminé par la SVM, maximisant la marge, permet de séparer les deux classes de manière optimale.

5.7.4. La classification linéaire :

On considère un ensemble de données $\{(x_1, c_1), (x_2, c_2), \dots, (x_p, c_p)\}$ où c_p est la classe à laquelle appartient le point x_p , c_p est égale à (-1) ou à $(+1)$ et x_p est un vecteur de dimension n de valeurs appartenant à l'intervalle $[0,1]$ ou $[-1,1]$. Ces *données d'apprentissage* désignent la classification correcte qu'on voudra que la SVM fasse par la suite, au moyen de l'hyperplan de division, qui obéit à la forme : $w \cdot x - b = 0$ (2.14)

On s'intéresse aussi aux hyperplans parallèles à l'hyperplan optimal qui sont tangents aux vecteurs de support des classes, et qui sont décrits par les équations : $w \cdot x - b = 1$ et $w \cdot x - b = -1$.

On veut que ces hyperplans maximisent la distance avec l'hyperplan de division et qu'il n'y ait aucun point de données entre les deux. En employant la géométrie, on trouve que la distance entre les hyperplans est $2 / |w|$, ainsi le but est de réduire au minimum $|w|$.

Pour exclure les points de données, on doit s'assurer que pour chaque i on a $w \cdot x_i - b \geq 1$ ou $w \cdot x_i - b \leq -1$ ou sous une autre équation $c_i (w \cdot x_i - b) \geq 1$ avec $1 \leq i \leq p$. Le problème maintenant est de minimiser $|w|$ sous cette dernière contrainte, ceci est un problème d'optimisation de *programme quadratique QP*. Il existe plusieurs algorithmes spécialisés pour résoudre rapidement les problèmes *QP* et la méthode la plus commune est l'algorithme *SMO*.

Après l'apprentissage, la SVM peut classifier les nouvelles données (données de test) en utilisant la règle de décision suivante :

$$\hat{C} = \begin{cases} 1, & \text{if } w \cdot x + b \geq 0 \\ -1, & \text{if } w \cdot x + b \leq 0 \end{cases} \quad (2.15)$$

On peut constater d'après la règle de classification qu'elle est seulement en fonction des vecteurs de supports (les données d'apprentissage se trouvant sur la marge) [49].

5.7.5. La classification non linéaire

L'algorithme original de l'hyperplan optimal est un classificateur linéaire, et pour avoir un classificateur non linéaire on applique le *kernel trick* à l'hyperplan à marge maximum, l'algorithme résultant est formellement semblable, sauf que chaque produit scalaire est remplacé par une fonction noyau non linéaire.

Parmi les noyaux les plus communs, on trouve :

- le noyau polynomial (homogène)
- le noyau polynomial (inhomogène)
- la fonction radiale de base RBF
- la fonction radiale de base de type gaussien
- la fonction sigmoïde

5.8. Modèle de Markov caché

Les modèles de Markov cachés *HMMs* sont massivement utilisés en reconnaissance de formes, en intelligence artificielle ou encore en traitement automatique du langage naturel. Les *HMMs* sont généralement employés pour la modélisation statistique de séries de vecteur de temps non stationnaire. En considérant l'information du visage comme étant une séquence variable dans le temps, le *HMM* peut être appliqué à la reconnaissance de visages [46]. Les caractéristiques faciales les plus significatives d'une image de visage frontale, à savoir les cheveux, le front, les yeux, le nez et la bouche, se présentent dans un ordre naturel de haut en bas, même dans le cas où il y'a des petites rotations. En se basant sur cette observation, l'image d'un visage peut être modélisée en utilisant un *HMM* unidimensionnel en assignant à chacune de ces régions un état. Le *HMM* classifie une caractéristique par la propriété de la *chaîne de Markov*. Une séquence des valeurs des pixels forme une *chaîne de Markov*, si la probabilité que le système à l'instant $n+1$ soit à l'état x_{n+1} dépend uniquement de la probabilité que le système à l'instant n soit à l'état x_n .

Dans une *chaîne de Markov*, la transition d'un état à un autre est probabiliste ce qui donne une distribution de probabilité de l'ensemble des sorties de chaque état. Ce résultat est employé pour faire la comparaison entre deux visages [46] [49].

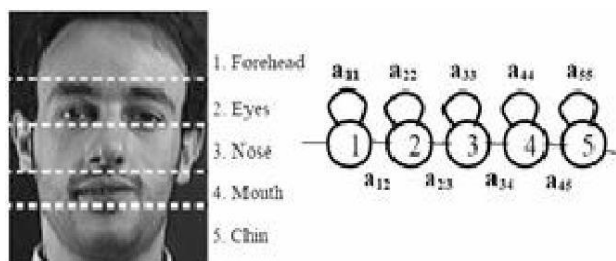


Figure 2.13 : Les 5 états du HMM (de haut en bas)

Pour une image de visage d'une personne donnée de l'ensemble d'apprentissage, le but de l'étape d'apprentissage est d'optimiser les paramètres pour mieux décrire l'observation. La reconnaissance est effectuée en assortissant l'image de test contre chaque modèle d'apprentissage (chaque *HMM* représente une personne différente). Et dans la fin de cette procédure, l'image est convertie en une séquence d'observation et la probabilité est calculée pour chaque modèle enregistré.

Le modèle avec la probabilité la plus élevée indique l'identité de la personne inconnue [39] [46].

L'approche *HMM* a des performances de reconnaissance satisfaisantes, cependant il nécessite des calculs intenses ce qui augmente les temps d'exécution.

5.9. Méthodes basées sur l'apparence locale (Local appearance-based methods)

Ces techniques sont utilisées de manière modulaire pour les différentes régions faciales. Un modèle global est alors défini à partir de la combinaison des différents modèles locaux. Ainsi, les différentes régions faciales ne seront plus affectées de la même manière par les différentes sources de variabilité. Par exemple, le port de lunettes de soleil change considérablement l'aspect des yeux, tandis qu'un sourire affecte plus la région de la bouche.

Deux paramètres sont utilisés pour définir les régions locales du visage : la forme (rectangle, ellipse) et la taille. Les caractéristiques des régions locales sont déterminées à partir d'une analyse des valeurs de niveau gris [54]. Des techniques comme les Ondelettes de Gabor [55] ou de Har [56] et l'analyse fractale sont aussi utilisées pour l'extraction de caractéristiques. En général, les caractéristiques à base de valeurs de gris préservent l'information de texture, tandis que les caractéristiques de Gabor sont plus robustes face aux changements d'illumination et aux transformations géométriques [55].

5.9.1. ACP modulaire

Pentland et al. [12] ont introduit l'approche des « Modular Eigenspace ». Les régions faciales utilisées sont: la totalité du visage, les yeux et le nez. Une ACP est appliquée sur chacune de ces régions et les résultats de classifications obtenus sont agrégés. La bouche étant trop sensible à des changements d'expression faciale, sa prise en compte engendre une baisse du taux de reconnaissance. Cette approche peut être qualifiée d'hybride, puisqu'elle

utilise à la fois des caractéristiques globales et locales.

Les auteurs ont montré qu'elle est plus efficace que les techniques globales ou strictement locales appliquées séparément.

5.10. Template matching

Le template matching est une technique globale de reconnaissance de visage. La corrélation est généralement exploitée pour mesurer la similitude entre un template (masque) stocké et l'image à reconnaître. Les templates devraient être délibérément conçus pour couvrir la variété de variations possibles d'image. Pendant la recherche dans l'image, l'échelle et la rotation devraient également être soigneusement considérées pour accélérer le processus. Cette technique a aussi été exploitée pour localiser les caractéristiques saillantes "salient features" comme les yeux le nez et la bouche, dans une image de visage. D'affines transformations ont été introduites dans le processus de reconnaissance, lors de changement de pose [50].

5.10.1. L'algorithme du template matching

On commence par l'acquisition, la lecture et la normalisation des images d'apprentissage. On calcule ensuite le visage moyen de chaque classe de personnes (les templates). Dans l'étape de reconnaissance, on fait l'acquisition, la lecture et la normalisation de l'image de vérification. Puis on calcule les différences entre l'image de vérification et les templates (les différences sont des images). On calcule après les sommes de ces différences (somme des pixels de l'image différence). Et finalement la somme minimale va référencer la classe auquel appartient l'image de vérification.

6. Performances d'un système de reconnaissances de visage

Les performances d'un système de reconnaissance dépendent de plusieurs facteurs qui interviennent à plusieurs niveaux et qui peuvent limiter le degré de précision.

Cependant, il serait judicieux de s'intéresser à ces facteurs avant de mesurer la performance d'un système de reconnaissance.

Nous citons ici les principaux facteurs :

- L'environnement au moment de l'acquisition.
- Les différentes positions des capteurs.
- La qualité des capteurs.
- La mauvaise interaction entre l'utilisateur et les capteurs.

6.1. Performance d'un système d'identification

Pour évaluer les performances d'un système d'identification, on calcule le taux de reconnaissance du système.

On effectue n fois le test et on voit, si la réponse du système (la personne la plus proche résultante de la comparaison entre le test et la base d'apprentissage) est juste alors le taux est incrémenté et à la fin on aura le pourcentage $\text{taux}/100$ qui indique sur 100 tests combien de bonnes réponses sont retournées par le système.

Ainsi plus ce taux s'approche de 100 plus le système est performant, sans oublier qu'on est dans le cas d'un système temps réel qui doit être rapide, ce qui ajoute le facteur temps dans l'évaluation des systèmes biométriques d'identification.

6.2. Performance d'un système de vérification

Un système de vérification ne peut donner, lors d'une comparaison entre deux échantillons qu'un résultat sous forme de probabilité de coïncidence ou score (S), ce score est comparé à un seuil de décision (T) que l'on détermine grâce à de nombreux tests; Si S est supérieur à T le sujet en question est accepté et dans le cas contraire le sujet est considéré comme imposteur.

Dans ce cas, deux types d'erreurs peuvent être commis :

Soit le sujet est rejeté alors qu'il s'agit bien d'un utilisateur enregistré, c'est ce qu'on appelle le Faux Rejet « False rejection FR», soit le sujet est accepté alors qu'il s'agit d'un imposteur, on l'appelle Fausse Acceptation « False acceptance FA».

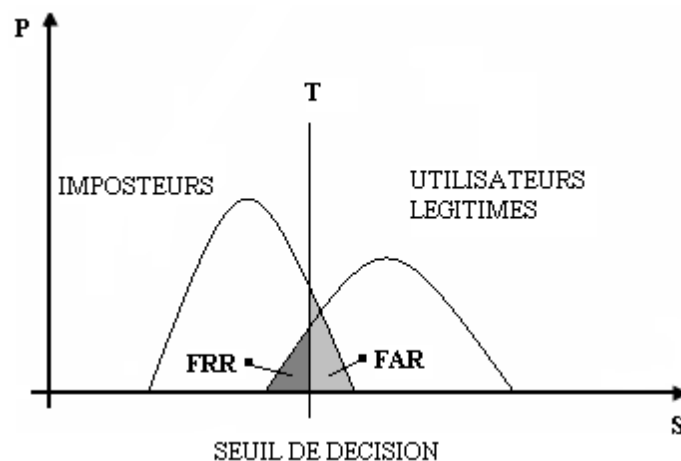


Figure 2.14 : La distribution du taux de vraisemblances.

Donc les performances d'un système de vérification se mesurent à l'aide des critères suivants :

- La rapidité : qu'on peut évaluer par la vitesse d'exécution puisqu'il s'agit d'une application temps réel.

- La précision : ce critère est évalué par les taux d'erreurs (taux de faux rejet et taux de fausse acceptation) qui sont eux-mêmes fonction du seuil de décision; et puisque le résultat d'une comparaison est toujours une estimation (un score), tous les systèmes biométriques donnent la possibilité de paramétrer le seuil d'acceptabilité :

a) soit en exigeant du système un contrôle strict, en mettant par exemple le seuil à 99,8%, signifiant par là que 2 échantillons ne seront considérés comme provenant d'un même individu que si le score de similitude est supérieur à 99,8% ; la conséquence sera d'augmenter le nombre de « faux rejets ».

b) soit en étant plus tolérant, en autorisant par exemple que le système réponde positivement si le score de similitude n'est pas en dessous de 95% ; La conséquence sera d'augmenter le taux de « fausses acceptations ».

Dans un système idéal $FRR = FAR = 0$, mais ce n'est pas le cas dans la pratique; quand FR augmente, FA diminue et vice versa. Par conséquent, il faut trouver un compromis entre les deux taux qui est la jonction des courbes, c'est-à-dire le point X où le couple (FAR, FRR) est minimal.

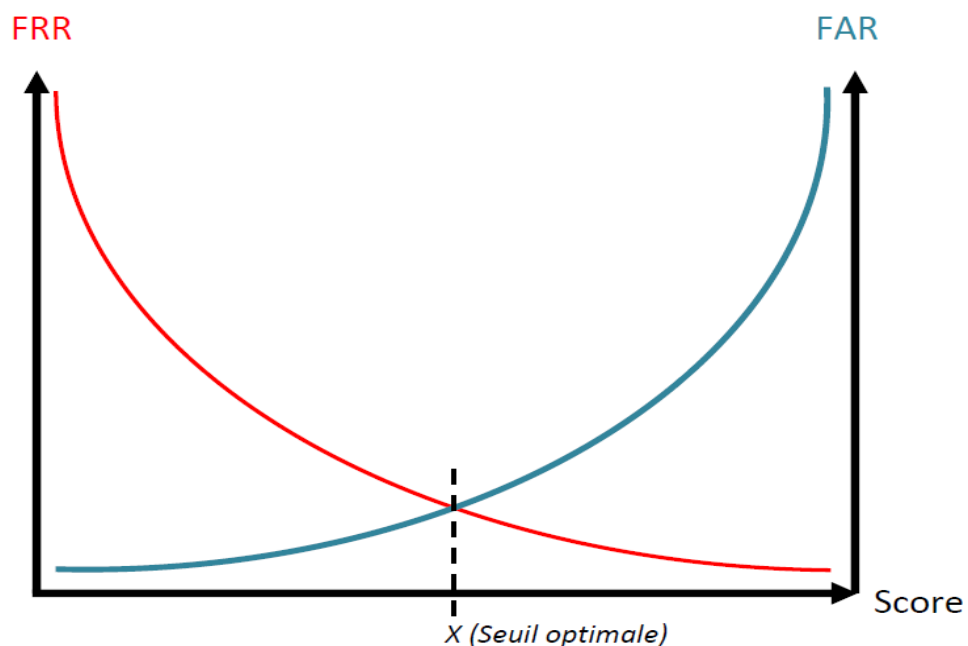


Figure 2.15 : Relation entre FAR et FRR.

7. Conclusion

Ces dernières années, de nombreuses approches pour la reconnaissance automatique de visages ont été proposées. Elles peuvent être regroupées en deux types d'approches principales. Les approches globales pour lesquelles les caractéristiques sont extraites directement de l'image du visage en entier et les approches locales basées sur l'extraction de traits locaux du visage, tel que les yeux, le nez ou la bouche.

Dans ce chapitre nous avons passé en revue les méthodes les plus connues de la détection de visage puis nous nous sommes focalisés sur les principaux algorithmes de la reconnaissance faciale, qui se divisent en trois catégories à savoir les méthodes globales, locales et hybrides.

Nous avons décrit également leurs avantages et leurs inconvénients puis dégager les problèmes liés à leur application dans un environnement réel. En effet, un système de reconnaissance automatique de visages est efficace dans le cas où les images faciales seraient bien cadrées et prises dans de bonnes conditions de luminosité. Sa performance se dégrade fortement lorsque ces images, utilisées pour l'apprentissage ou lors de la reconnaissance, présentent des variations importantes de luminosité et/ou de pose du visage.

Chapitre 3

Détection par Viola et Jones

Reconnaissance par ACP (Eigenface)

1. Introduction

La reconnaissance du visage a atteint un haut niveau de performances, même avec l'utilisation d'un seul exemple d'apprentissage. Dans les conditions actuelles des travaux sur la reconnaissance faciale, le système visuel humain reste encore le plus robuste face aux diverses variations pouvant altérer le processus d'identification : changement des conditions d'éclairage, variations de l'expression faciale, modifications de l'apparence du visage à travers la présence ou l'absence de lunettes, barbe, maquillage.

De nombreuses techniques ont été développées ces dernières années, nous avons passées en revue les plus connues dans le chapitre deux. Pour la détection de visage nous utilisons l'algorithme Viola et Jones et pour la reconnaissance la technique Eigenface, qui sont des techniques particulièrement prisées par les chercheurs de la communauté de la biométrie.

2. Explication de la méthode de Viola & Jones

2.1 Principe

La méthode de Viola & Jones consiste à balayer une image à l'aide d'une fenêtre de détection de taille initiale 24px par 24px (dans l'algorithme original) et de déterminer si un visage y est présent. Lorsque l'image a été parcourue entièrement, la taille de la fenêtre est augmentée et le balayage recommence, jusqu'à ce que la fenêtre fasse la taille de l'image. L'augmentation de la taille de la fenêtre se fait par un facteur multiplicatif de 1,25. Le balayage, quant à lui, consiste simplement à décaler la fenêtre d'un pixel. Ce décalage peut être changé afin d'accélérer le processus, mais un décalage d'un pixel assure une précision maximale.

Cette méthode est une approche basée sur l'apparence, qui consiste à parcourir l'ensemble de l'image en calculant un certain nombre de **caractéristiques** dans des zones rectangulaires qui se chevauchent. Elle a la particularité d'utiliser des caractéristiques très simples mais très nombreuses [8].

Il existe d'autres méthodes mais celle de Viola & Jones est la plus performante à l'heure actuelle. Ce qui la différencie des autres est notamment :

- l'utilisation d'**images intégrales** qui permettent de calculer plus rapidement les caractéristiques
- la **sélection par boosting** des caractéristiques
- la combinaison en **cascade de classifieurs** boostés, apportant un net gain de temps d'exécution

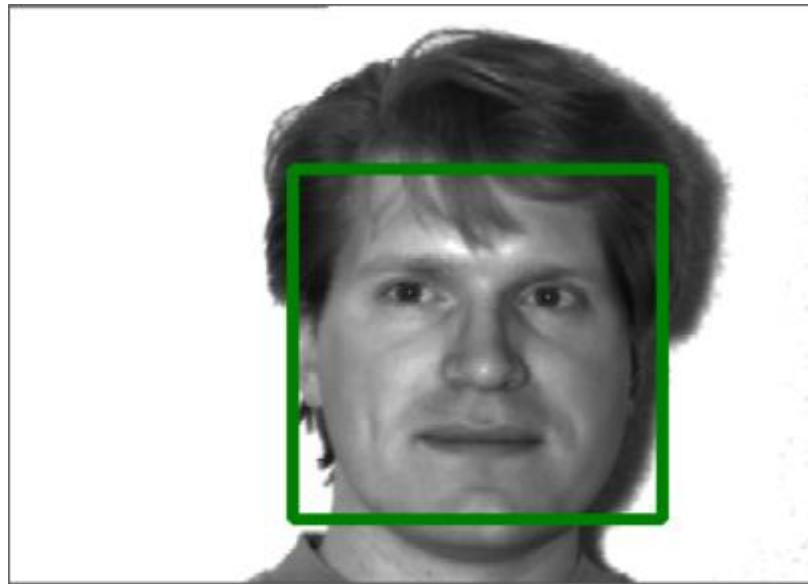


Figure 3.1 : Exemple de détection de visage.

2.2 Apprentissage du classifieur

Une étape préliminaire et très importante est l'apprentissage du classifieur. Il s'agit d'entraîner le classifieur afin de le sensibiliser à ce que l'on veut détecter, ici des visages. Pour cela, il est mis dans deux situations.

La première où une énorme quantité de cas positifs lui sont présentés et la deuxième où, à l'inverse, une énorme quantité de cas négatifs lui sont présentés. Concrètement, une banque d'images contenant des visages de personnes est passée en revue afin d'entraîner le classifieur. Ensuite, une banque d'images ne contenant pas de visages humains est passée.

2.3 Les caractéristiques :

Une caractéristique est une représentation synthétique et informative, calculée à partir des valeurs des pixels. Les caractéristiques utilisées ici sont les caractéristiques pseudo-haar. Elles sont calculées par la différence des sommes de pixels de deux ou plusieurs zones rectangulaires adjacentes [8].

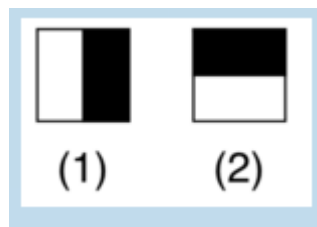


Figure 3.2 : deux zones rectangulaires adjacentes, la première en blanc, la deuxième en noire.

Les caractéristiques seraient calculées en soustrayant la somme des pixels noirs à la somme des pixels blancs. Pas si compliqué que ça finalement !

Les caractéristiques sont calculées à toutes les positions et à toutes les échelles dans une fenêtre de détection de petite taille, typiquement de 24x24 pixels ou de 20x15 pixels. Un très grand nombre de caractéristiques par fenêtre est ainsi généré, Viola et Jones donnant l'exemple d'une fenêtre de taille 24 x 24 qui génère environ 160 000 caractéristiques.

L'image précédente présente des caractéristiques pseudo-haar à seulement deux caractéristiques mais il en existe d'autres, allant de 4 à 14, et avec différentes orientations.

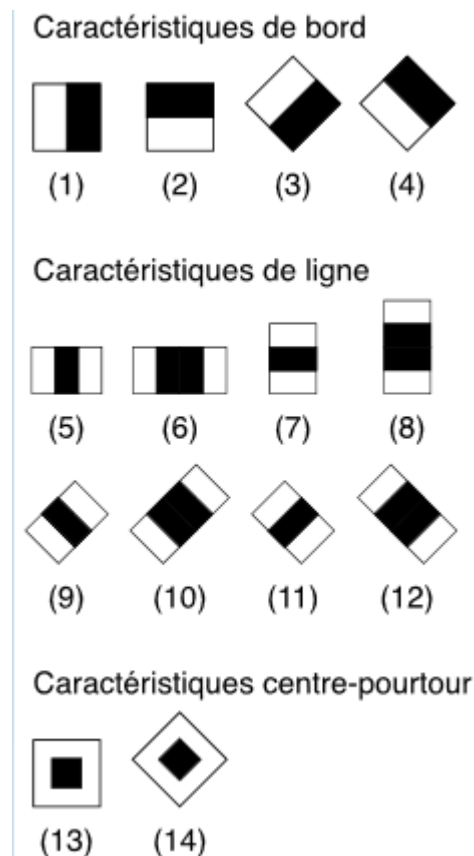


Figure3.3 : caractéristiques pseudo-haar [8].

Malheureusement, le calcul de ces caractéristiques de manière "classique" coûte cher en terme de ressources processeur, c'est là qu'interviennent les images intégrales.

2.4 Les images intégrales

Comme je vous l'ai dit précédemment, les images intégrales permettent de gagner du temps quant au calcul des caractéristiques. Il s'agit d'une image construite à partir de l'image d'origine, et de même taille qu'elle. Elle contient en chacun de ses points la somme des pixels situés au-dessus et à gauche du pixel courant. Regardez cette figure:

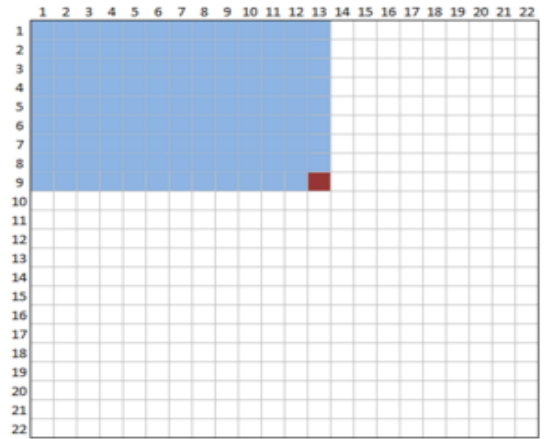


Figure3.4: Le pixel rouge est égale à la somme de tous les pixels bleu, soient ceux à gauche et au dessus.

Ok, mais je ne vois pas en quoi ça nous aide à calculer les caractéristiques...

Prenons un exemple. Nous souhaitons calculer la somme des pixels de la zone rectangulaire ABCD suivante:

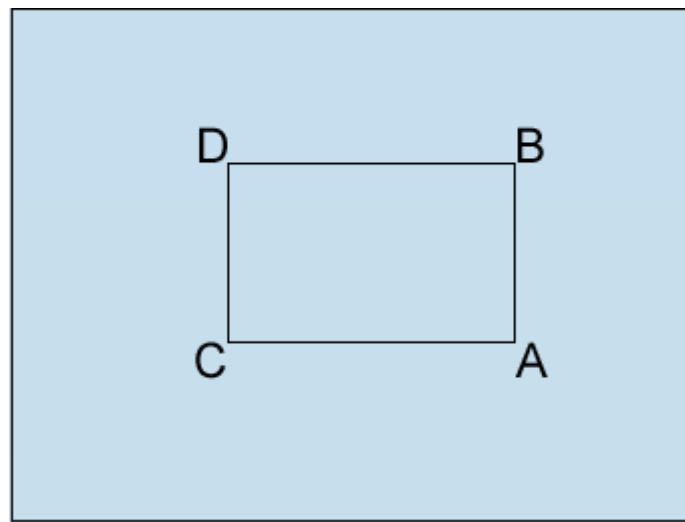


Figure3.5 : zone rectangulaire ABCD.

Grâce à l'image intégrale, nous connaissons la valeur de la somme des pixels en chacun des quatre points. Il suffit donc de faire : $A - B - C + D$.

En seulement trois opérations nous avons réussi à calculer notre somme de pixels ! Ainsi, on est en mesure de trouver la somme de pixels de n'importe quelle zone rectangulaire de l'image en seulement 3 opérations et 4 accès à l'image intégrale (un accès par point).

Une caractéristique pseudo-Haar à deux rectangles peut alors être déterminée en seulement 6 accès (2 points sont partagés) à l'image, et une caractéristique à 3 rectangles en seulement 8 accès.

1.5 Sélection par boosting

Nous arrivons maintenant à la dernière partie concernant la théorie : la sélection par boosting !

La sélection par boosting consiste à utiliser plusieurs classifieurs "faibles" mis en cascade plutôt que d'utiliser un seul classifieur "fort". En effet, avec un seul classifieur dit "fort" qui se présenterait de la sorte :

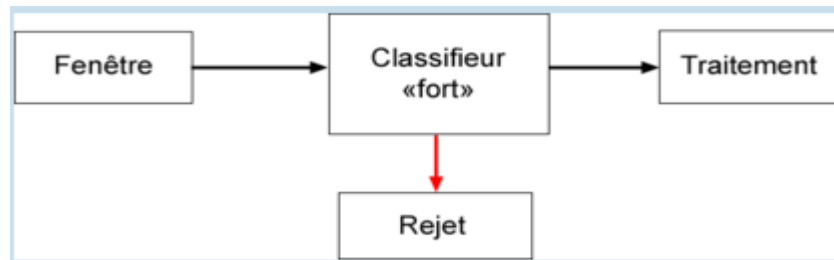


Figure 3.6 : présentation de classifieur fort.

Il faudrait attendre que le classifieur est analysé toute la fenêtre afin de savoir si un visage est présent dans l'image ou non. Une mise en cascade de classifieurs dont le critère de sélection serait moins sévère se présenterait de la sorte :

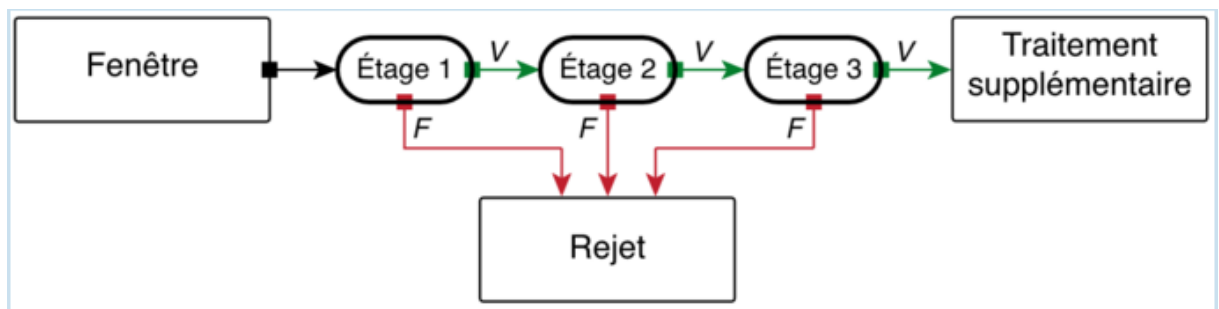


Figure 3.7 : présentation de classifieur selection [8].

Ainsi dès que l'un des étages estime qu'il n'y a pas de visage, la fenêtre est rejetée et l'algorithme passe à la suite ce qui permet un gain de temps considérable.

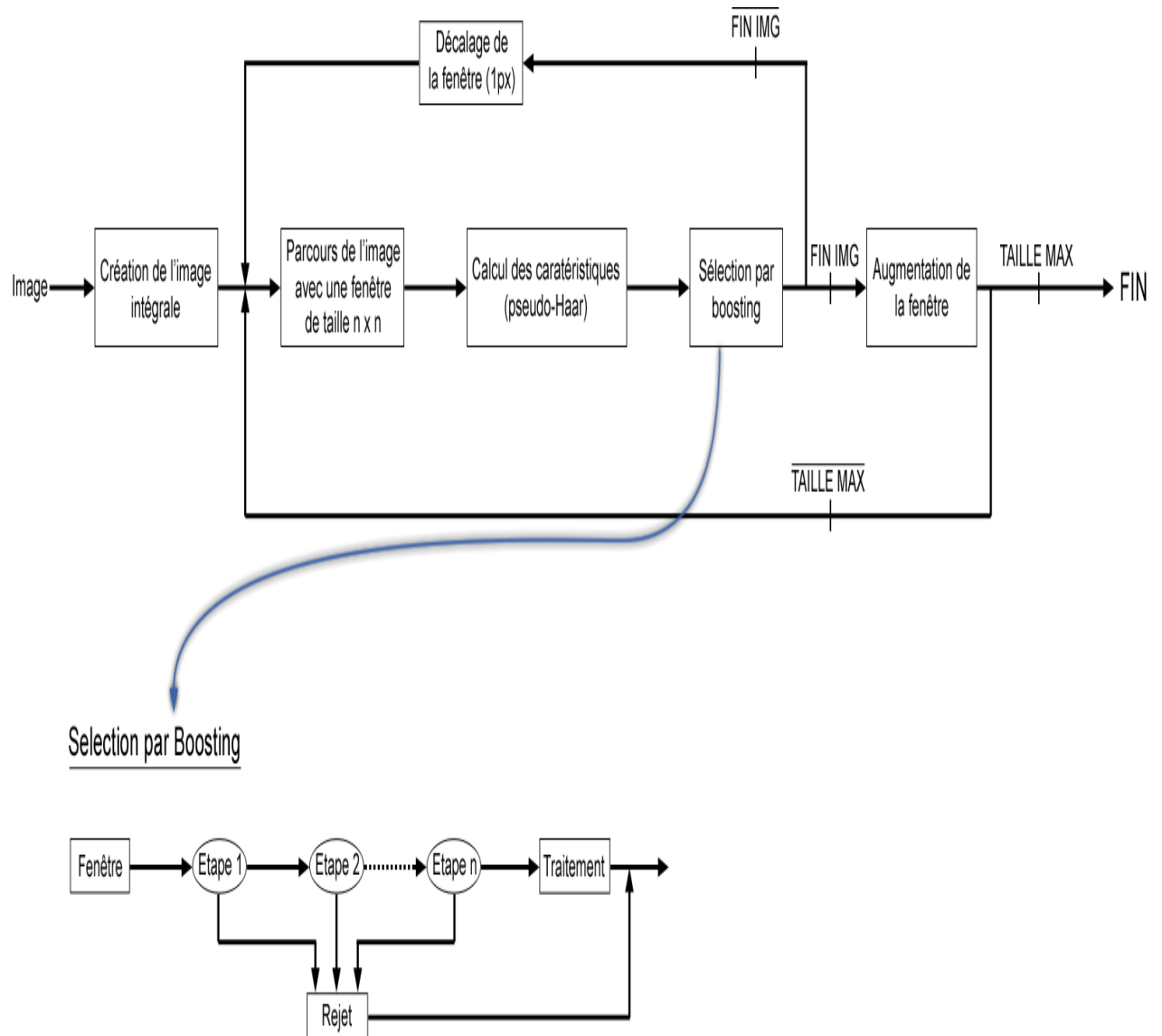


Figure 3.8 : schéma fonctionnel de l'algorithme Viola et Jones.

3. Présentation de la méthode Eigenface

L'algorithme ACP, PCA en anglais (Principal Component Analysis) est né des travaux de MA. Turk et AP. Pentland au MIT Media Lab, en 1991. Il est aussi connu sous le nom d'Eigenfaces car il utilise des vecteurs propres et des valeurs propres. Cet algorithme s'appuie sur des propriétés statistiques bien connues et utilise l'algèbre linéaire. Il est relativement rapide à mettre en œuvre mais il est sensible aux problèmes d'éclairément [12], de pose et d'expression faciale. Il est à la base de nombreux algorithmes globaux actuels.

L'idée principale consiste à exprimer les M images d'apprentissage selon une base de vecteurs orthogonaux particuliers, contenant des informations indépendantes d'un vecteur à l'autre. Ces nouvelles données sont donc exprimées d'une manière plus appropriée à la reconnaissance du visage.

Nous voulons extraire l'information caractéristique d'une image de visage, pour l'encoder aussi efficacement que possible afin de la comparer à une base de données de modèles encodés de manière similaire. En termes mathématiques, cela revient à trouver les vecteurs propres de la matrice de covariance formée par les différentes images de notre base d'apprentissage [39].

Notre travail consiste ces différentes étapes :

3.1. Création et gestion d'une base de données d'images (Prétraitement)

Création et gestion d'une base de données d'images faciales, la base de données utilisée est Yale qui contient 165 images en niveaux de gris au format GIF de 15 personnes. Il y a 11 images par sujet, une par l'expression du visage différent ou configuration « center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink [10]. »



Figure 3.9 : Les visages de la base de données Yale.

3.2. Convertir les images faciales de la base de données à des vecteurs faciaux

Une image I_i (m, n) est traitée comme un vecteur i ($m \times n, 1$) dans un espace vectoriel de grande dimension ($N = m \times n$), par concaténation des colonnes.

Après avoir rassemblé nos M images dans une unique matrice, nous obtenons une matrice d'images Γ , où chaque colonne représente une image Γ_i .

$$\Gamma = \begin{pmatrix} a_{1,1} & b_{1,1} & \dots & z_{1,1} \\ \vdots & \vdots & \dots & \vdots \\ a_{n,1} & b_{n,1} & \dots & z_{n,1} \\ \vdots & \vdots & \dots & \vdots \\ a_{1,m} & b_{1,m} & \dots & z_{1,m} \\ \vdots & \vdots & \dots & \vdots \\ a_{n,m} & b_{n,m} & \dots & z_{n,m} \end{pmatrix}$$

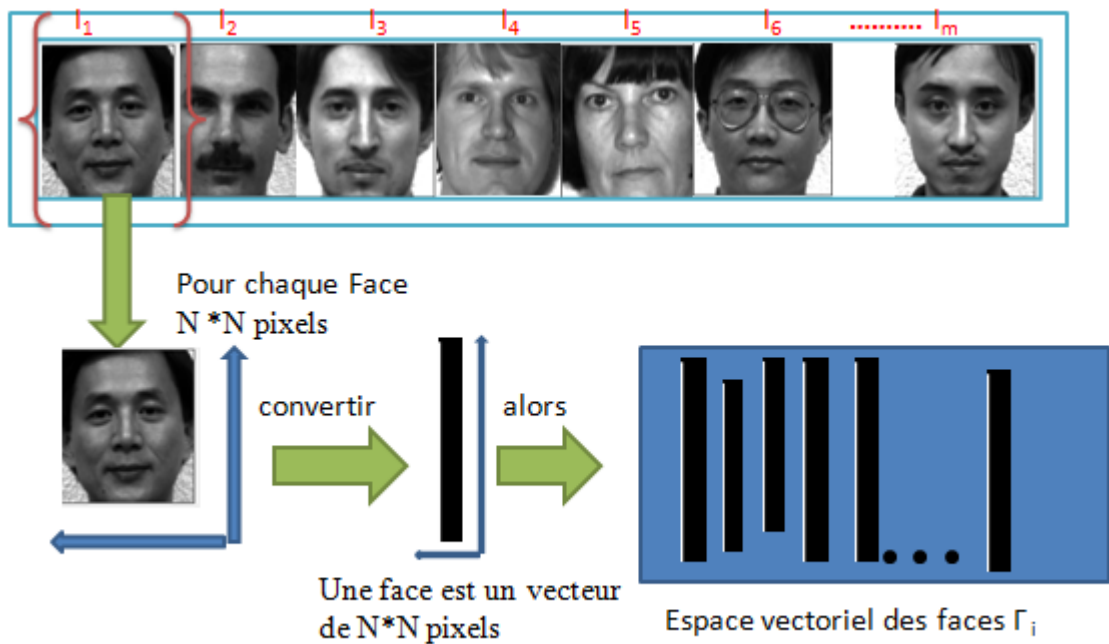


Figure 3.10 : Convertir les images faciales de notre base d'apprentissage à des vecteurs facials.

3.2. Normaliser les vecteurs facials

3.2.1. Calcule l'image moyenne Ψ de toutes les images collectées

On calcule ensuite l'image moyenne de toutes les images collectées cette image peut être vue comme le centre de gravité du jeu d'images.

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (3.1)$$

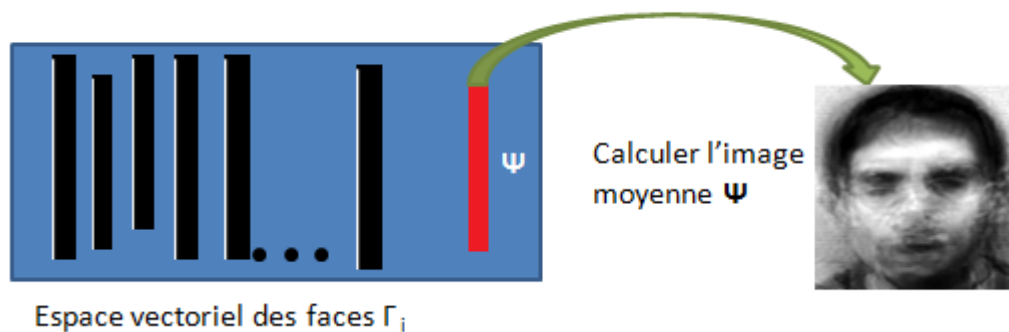


Figure 3.11. Image moyenne.

L'image moyenne est alors soustraite de chaque image avec la formule suivante:

$$\Phi_i = \Gamma_i - \Psi \quad \text{avec} \quad i=1 \dots M \quad (3.2)$$

Alors on trouve :

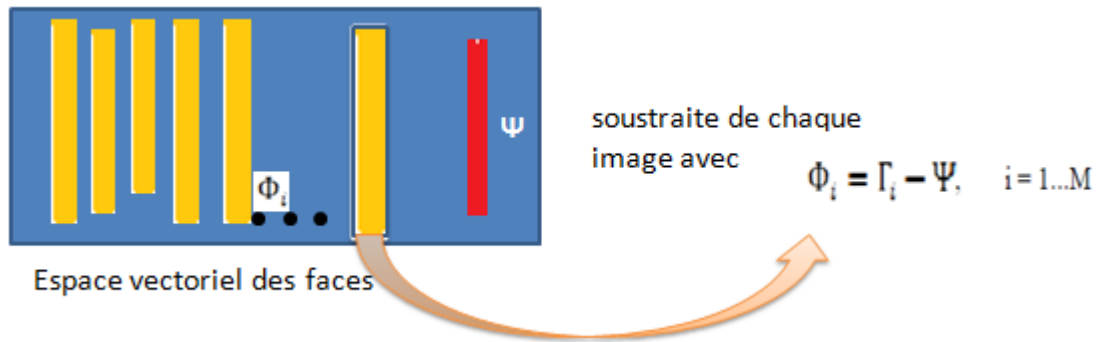


Figure 3.12 : soustraite de chaque image.

On calcule ensuite la matrice de covariance du jeu de données. Cette matrice peut être vue comme une matrice de moments d'ordre 2 :

$$C = \sum_{i=1}^M \Phi_i \Phi_i^T = A A^T, \quad A = [\Phi_1, \Phi_2, \dots, \Phi_M] \quad (3.3)$$

Avec $A = N^2 * M$

3.2.2. Calculer les vecteurs propres et les valeurs propres de cette matrice de covariance

Calculer les vecteurs propres et les valeurs propres de cette matrice de covariance C de taille $(N \times N)$, c'est-à-dire de l'ordre de la résolution d'une image. Le problème est que cela peut parfois être très difficile et très long.

En effet, si $N > M$ (si la résolution est supérieure au nombre d'images), il y aura seulement $M-1$ vecteurs propres qui contiendront de l'information [40] (les vecteurs propres restants auront des valeurs propres associées nulles).

Par exemple, pour 50 images de résolution 180×200 , nous pourrions résoudre une matrice L de 50×50 au lieu d'une matrice de 36000×36000 pour ensuite prendre les combinaisons linéaires appropriées des images Φ_i . Le gain de temps de calcul serait considérable, nous passerions d'une complexité de l'ordre du nombre de pixels dans une image à celle de l'ordre du nombre d'images.

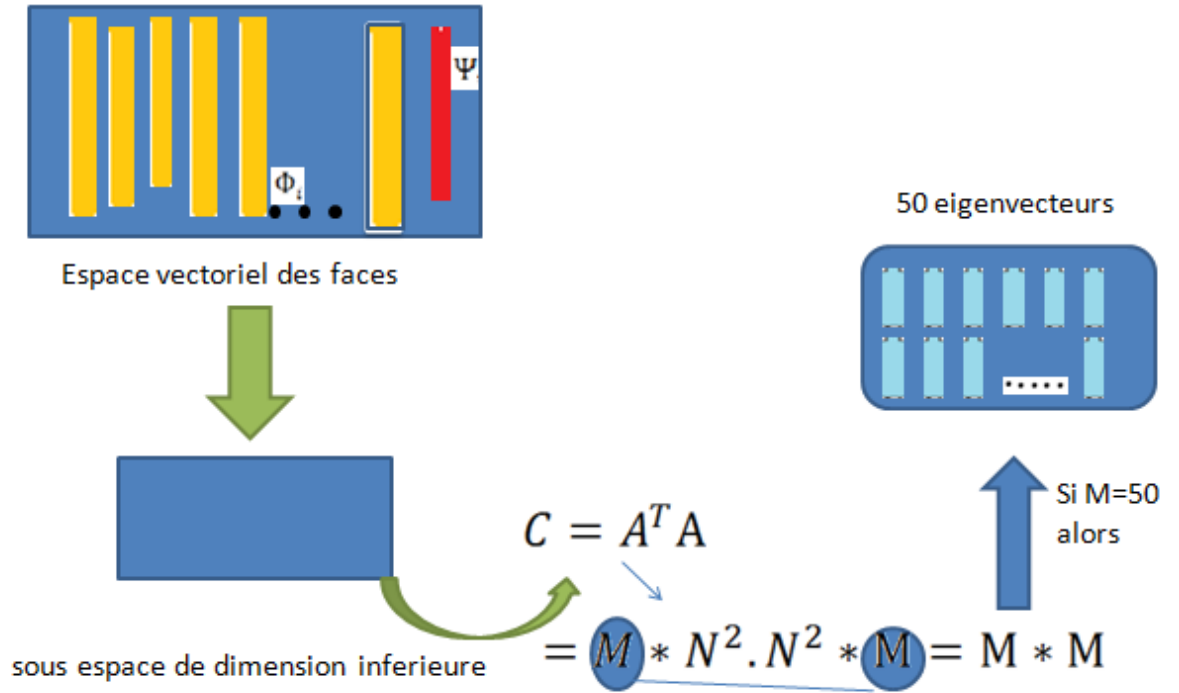


Figure 3.13 : matrice de covariance de M vecteurs.

Les étapes du processus qui nous permettent d'accélérer les calculs sont décrits ci-dessous :

Considérons les vecteurs propres e_i de $C=AA^T$, associés aux valeurs propres λ_i

On a :

$$C e_i = \lambda_i e_i \quad (3.4)$$

Les vecteurs propres v_i de $L = A^T A$, associés aux valeurs propres μ_i sont tels que :

$$L v_i = \mu_i v_i$$

Soit :

$$A^T A v_i = \mu_i v_i \quad (3.5)$$

En multipliant à gauche par A des deux côtés de l'égalité, nous obtenons :

$$A A^T A v_i = A \mu_i v_i$$

Puisque $C=AA^T$, nous pouvons simplifier :

$$C(A v_i) = \mu_i (A v_i) \quad (3.6)$$

De (3.4) et (3.5), nous voyons que $A v_i$ et μ_i , sont respectivement les vecteurs propres et les valeurs propres de C:

$$\begin{cases} e_i = A v_i \\ \lambda_i = \mu_i \end{cases} \quad (3.7)$$

Nous pouvons donc trouver les valeurs propres de cette énorme matrice C en trouvant les valeurs propres d'une matrice L beaucoup plus petite. Pour trouver les vecteurs propres de C , il suffit juste de multiplier les vecteurs propres de L par la matrice A .

Les vecteurs propres trouvés sont ensuite ordonnés selon leurs valeurs propres correspondantes, de manière décroissante. Plus une valeur propre est grande, plus la variance capturée par le vecteur propre est importante. Cela implique que la majeure partie des informations est contenue dans les premiers vecteurs propres.

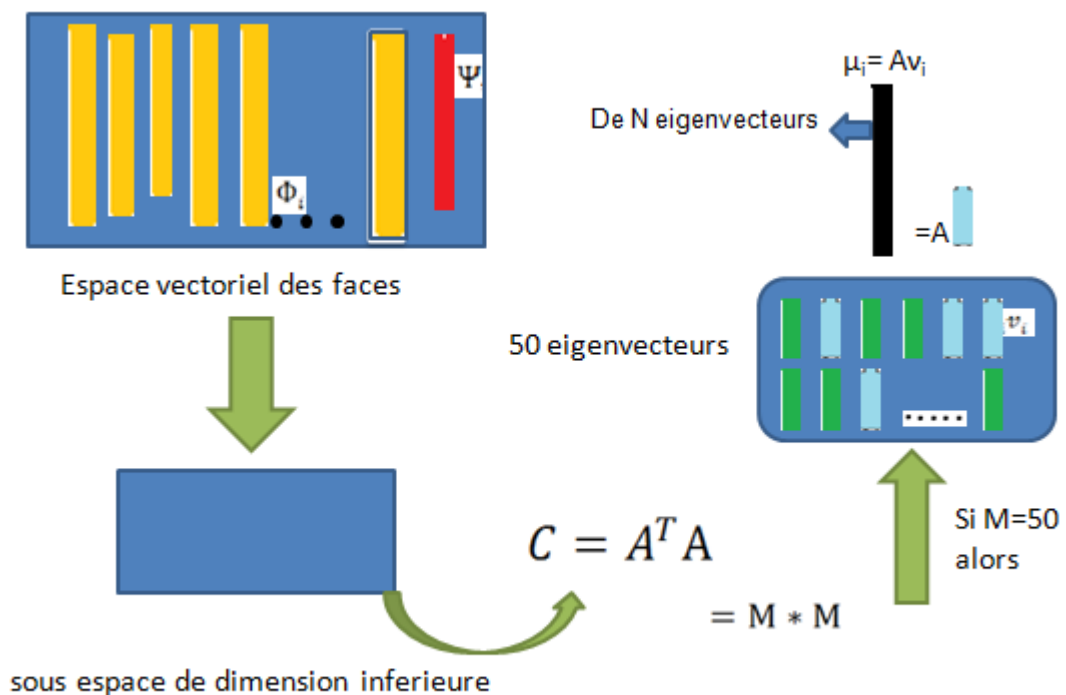


Figure 3.14 : Les vecteurs propres et les valeurs propres.

3.2.3. Choix du nombre de vecteurs propres

L'étape suivante consiste à ne sélectionner que les « k » meilleurs vecteurs propres (ceux avec les « k » plus grandes valeurs propres).

Le nombre de vecteurs propres associés aux plus grandes valeurs propres à retenir est un grand défaut de cette technique. Pour le choisir, les chercheurs ont adopté différentes solutions:

- Pour un ensemble de 115 images, Sirovitch et Kirby ont trouvé que 40 eigenpictures sont suffisantes pour représenter efficacement cet ensemble.
- Turk et Pentland [12] l'ont choisi heuristiquement. Pour leurs tests, sur une base de 16 individus, 7 vecteurs propres ont été retenus,

- Moghaddam [15] a préservé, pour comparer différentes approches de reconnaissance de visages 20 vecteurs propres en justifiant son choix par une erreur de reconstruction raisonnable (0.0012) et un taux de reconnaissance 80% obtenu par eigenfaces sur une base de 1829 images,
- Zhao et al ont retenu 300 vecteurs propres pour une base de 1038 images après avoir observé que pour un nombre très élevé, les eigenfaces ne représentent pas des visages, donc leur choix était basé sur l'allure des eigenfaces au lieu des valeurs propres.

Pratiquement, le choix de « k » peut se faire soit en fixant un certain pourcentage D tel que:

$$\frac{\sum_{i=k+1}^n \lambda_i}{\sum_{i=1}^n \lambda_i} < D \quad (3.8)$$

Avec n le nombre total des valeurs propres ou on se basant sur le fait que la $i^{\text{ème}}$ valeur propre est égale à la variance le long de la $i^{\text{ème}}$ composante principale et on cherche, donc « k » le long du spectre des valeurs propres juste avant qu'il s'annule [12].

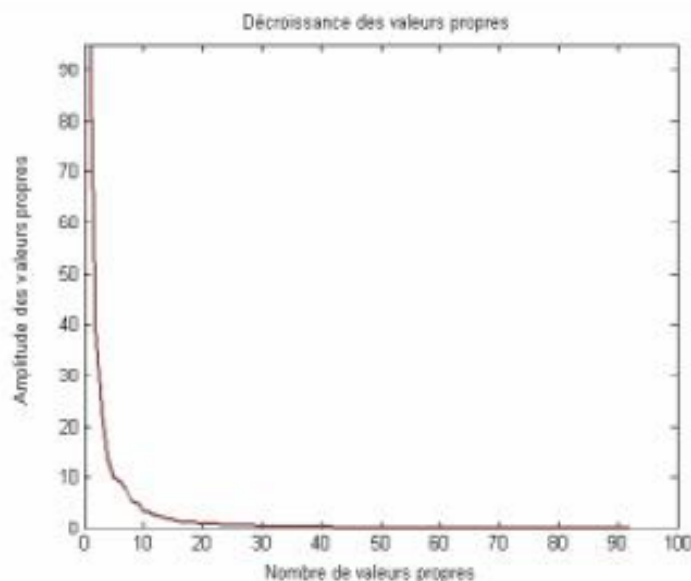


Figure 3.15 : Décroissance des valeurs propres de la base de données utilisée en fonction du nombre de vecteurs propres retenu.

Donc, les « k » premiers vecteurs propres correspondant aux « k » plus grandes valeurs propres sont un paramètre critique sur lequel dépend la performance du système de reconnaissance de visages (temps de calcul et taux de reconnaissance) et aussi un critère d'évaluation [12] des méthodes basées sur l'ACP.

Son choix dépend des contraintes d'application liées au temps de calcul et au nombre

des images de visages disponibles et aussi de la qualité des images de visages reconstruites.

A partir de là, on définit un espace vectoriel engendré par ces k vecteurs propres, que l'on appelle l'espace des visages E_v (Face Space). Les images originales peuvent être reconstituées par combinaison linéaire de ces vecteurs propres. Les représentations graphiques de ces vecteurs rappellent un peu des images fantômes, chacune mettant en avant une partie du visage, on les appelle Eigenfaces.



Figure 3.16 : Images représentant les Eigenface.

3.2.4. Utilisation des Eigenfaces pour la classification des images visage

Nous allons maintenant projeter nos images de départ E_v , une image Γ_i est alors transformée en ses composantes Eigenfaces par une simple opération de projection vectorielle :

$$\omega_k = e_k^T (\Gamma_i - \Psi), K=1, \dots, M \quad (3.9)$$

Les vecteurs ω_k sont appelés poids et forment une matrice [5] $\Omega_T = [\omega_1, \omega_2, \dots, \omega_M]$ qui décrit la contribution de chaque eigenface dans la représentation de l'image d'entrée. La matrice Ω_T est alors utilisée pour trouver quelle est, parmi un nombre prédéfini de classes, celle qui décrit le mieux une image d'entrée.

La méthode la plus simple pour déterminer quelle classe de visage fournit la meilleure description d'une image d'entrée est de trouver la classe de visage k qui minimise la distance Euclidienne.

$$\varepsilon_k^2 = \|\Omega - \Omega_k\|^2 \quad (3.10)$$

Où Ω_T est un vecteur qui décrit la k^{eme} classe de visage.

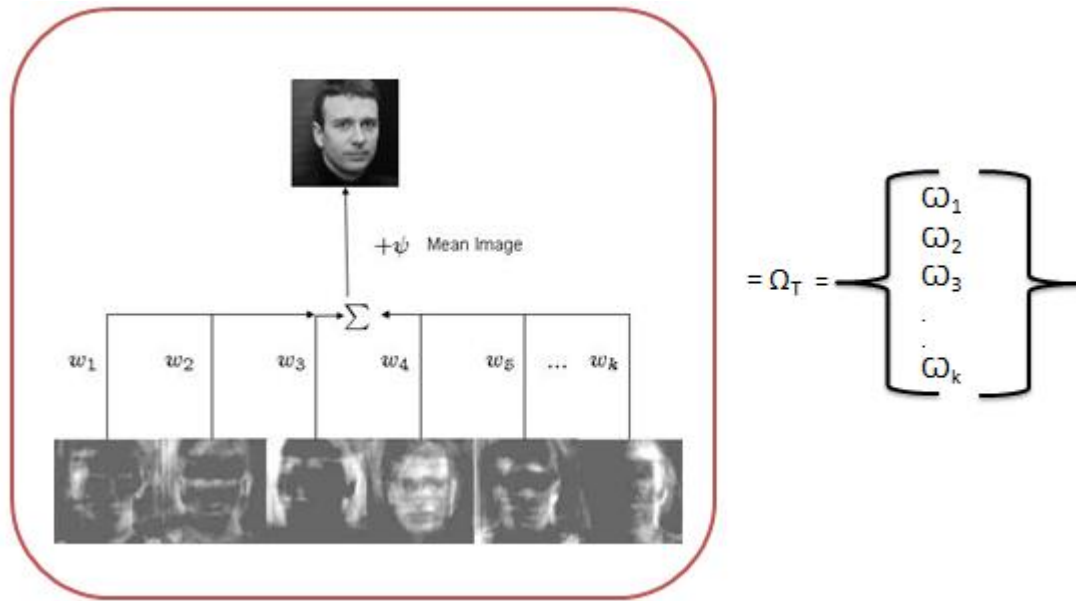


Figure 3.17 : Représentation de k Eigenface + l'image principale.

Un visage appartient à une classe k quand le minimum ε_k est en dessous d'un certain seuil Θ_ε . Dans le cas contraire, le visage est classé comme étant inconnu et peut éventuellement être utilisé pour créer une nouvelle classe de visage.

Il y a quatre possibilités pour une image d'entrée d'être reconnue ou non :

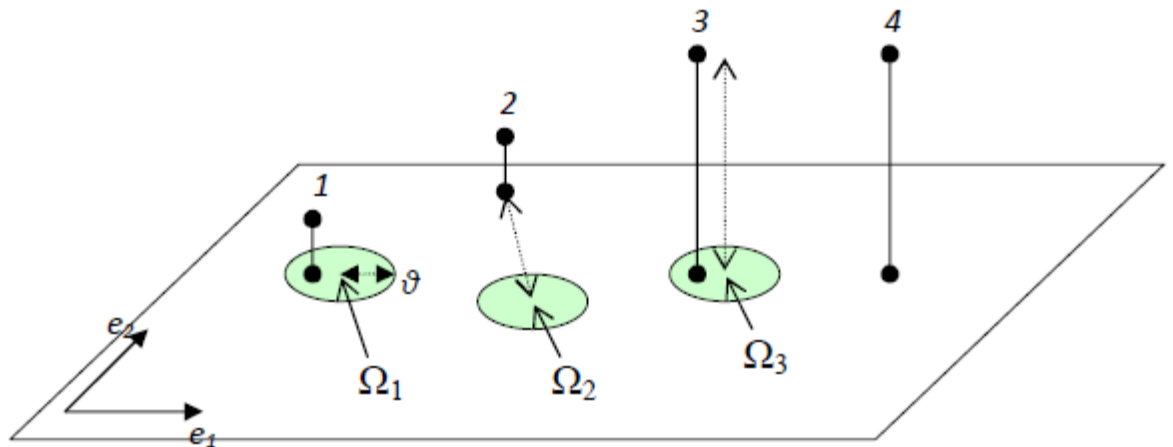


Figure 3.18 : Illustration des cas possibles de classification d'une image.

- Cas 1 : un individu est reconnu et identifié.
- Cas 2 : un individu inconnu du système est présent.
- Cas 3 et 4 : indiquent que l'image n'est pas une image de visage, Cas 3, l'image est éloignée de E_v mais la projection est proche d'une classe connue, risque de fausse acceptation.

3.3. Résumé de la méthode

L'apprentissage des visages propres s'effectue selon les étapes suivantes :

- 1- Collecte des M images faciales et construction de la matrice T de taille M, par concaténation des colonnes des images faciales. Prétraitement des images collectées.
- 2- Calcul du visage moyen en sommant les colonnes de la matrice T et en divisant le vecteur résultant par le nombre d'image d'entrée (M).
- 3- Soustraction du visage moyen de la matrice T pour obtenir la matrice A ; où chaque élément représente la variance des valeurs d'intensité de chaque pixel.
- 4- Calcul de la matrice C, $C = A^T A$.
- 5- Calcul des vecteurs propres de C' et les triés dans un ordre descendant selon les valeurs propres associées.
- 6- Calcul des vecteurs propres de la matrice de covariance C et obtention des visages propres en multipliant les vecteurs propres de C' par la matrice A.
- 7- Choix des K meilleur valeurs propres et les vecteurs propres associés.
- 8- Détermination du poids des images d'entrée en projetant chaque image dans l'espace visage.
- 9- Chaque visage est maintenant représenté par un vecteur qui est utilisé pour reconstruire les images. Et enfin sauvegarde des calculs du visage moyen, des eigenfaces et du poids des images

Les neuf étapes décrites transformeront une base de données d'images faciales en un ensemble de projections dans l'espace visage (face space).

3.4 : Mesures de Distance :

Lorsqu'on souhaite comparer deux vecteurs de caractéristiques issus du module d'extraction de caractéristiques d'un système biométrique, on peut soit effectuer une mesure de similarité (ressemblance), soit une mesure de distance (divergence).

La première catégorie de distances est constituée de distances Euclidiennes et sont définies à partir de la distance de Minkowski d'ordre p dans un espace euclidien R^N (N déterminant la dimension de l'espace euclidien).

Considérons deux vecteurs $X = (x_1, x_2, \dots, x_n)$ et $Y = (y_1, y_2, \dots, y_n)$, la distance de Minkowski d'ordre p notée L_p est définie par :

$$L_p = (\sum_{i=1}^n |x_i - y_i|^p)^{1/p} \quad (3.11)$$

Nous allons présenter quelques mesures de distance dans l'espace original des

images puis dans l'espace de Mahalanobis.

3.4.1. Distances Euclidiennes:

- Distance City Block (L1)

Pour $p = 1$, on obtient la distance City-Block (ou distance de Manhattan) :

$$L_1(x,y)=\sum_{i=1}^n |x_i - y_i| \quad (3.12)$$

- Distance Euclidienne (L2)

Pour $p = 2$, on obtient la distance euclidienne :

$$L_2(x,y)=\sqrt{\sum_{i=1}^n |x_i - y_i|^2} \quad (3.13)$$

Les objets peuvent alors apparaître de façons très différentes selon la mesure de distance choisie.

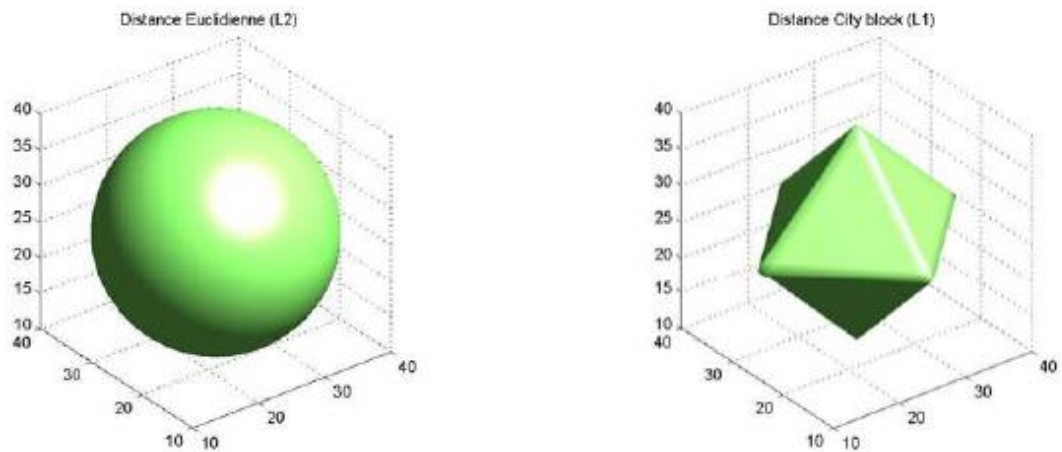


Figure 3.19 : Représentation d'une sphère avec la distance euclidienne (3.4.a) et la distance City-Block (3.4.b).

3.4.2. Distances dans l'Espace de Mahalanobis :

3.4.2.1. De l'espace des images à l'espace de Mahalanobis :

Avant de pouvoir effectuer des mesures de distance dans l'espace de Mahalanobis, il est essentiel de bien comprendre comment l'on passe de l'espace des images \mathbf{I}_m à l'espace de Mahalanobis. \mathcal{E}_{Mah} .

En sortie de l'algorithme PCA, nous obtenons des vecteurs propres associés à des valeurs propres (représentant la variance selon chaque dimension). Ces vecteurs propres définissent une rotation vers un espace dont la covariance entre les différentes dimensions est nulle. L'espace de Mahalanobis est un espace où la variance selon chaque dimension est égale à 1. On l'obtient à partir de l'espace des images \mathbf{I}_m en divisant chaque vecteur propre par son écart-type correspondant.

Soit u et v deux vecteurs propres de I_m , issus de l'algorithme PCA, et m et n deux vecteurs de \mathcal{E}_{Mah} . Soit λ_i les valeurs propres associées aux vecteurs u et v , et σ_i l'écart-type, alors on définit $\lambda_i = \sigma_i$. Les vecteurs u et v sont reliés aux vecteurs m et n à partir des relations suivantes :

$$m_i = \frac{u_i}{\sigma_i} = \frac{u_i}{\sqrt{\lambda_i}} \text{ et } n_i = \frac{v_i}{\sigma_i} = \frac{v_i}{\sqrt{\lambda_i}} \quad (3.14)$$

3.4.2.2. Mahalanobis L1 (MahL1) :

Cette distance est exactement la même que la distance City-Block sauf que les vecteurs sont projetés dans l'espace de Mahalanobis. Ainsi, pour des vecteurs propres u et v de projections respectives m et n sur l'espace de Mahalanobis, la distance Mahalanobis L1 est dénie par :

$$Mah_{L1}(u, v) = \sum_{i=1}^n |m_i + n_i| \quad (3.15)$$

3.4.2.3. Mahalanobis L2 (MahL2) :

Cette distance est identique à la distance euclidienne à part qu'elle est calculée dans l'espace de Mahalanobis. Ainsi, pour des vecteurs propres u et v de projections respectives m et n sur l'espace de Mahalanobis, la distance Mahalanobis L2 est dénie par :

$$L_2(x, y) = \sqrt{\sum_{i=1}^n |x_i + y_i|^2} \quad (3.16)$$

Par défaut, lorsqu'on parle de distance de Mahalanobis, c'est à cette distance que l'on doit se référer.

3.4.2.4. Cosinus de Mahalanobis (MahCosine) :

Il s'agit tout simplement du cosinus de l'angle entre les vecteurs u et v , une fois qu'ils ont été projetés sur \mathcal{E}_{Mah} et normalisées par des estimateurs de la variance.

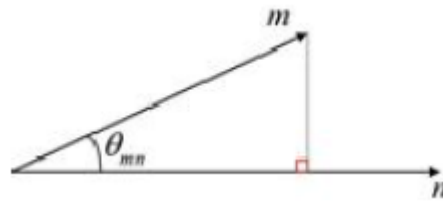


Figure 3.20 : Les deux vecteurs m et n dans l'espace de Mahalanobis.

Nous avons donc par définition :

$$S_{MahCosine}(u, v) = \cos(\theta_{mn}) \quad (3.17)$$

De plus, on peut écrire :

$$\cos(\theta_{mn}) = \frac{|m||n| \cos(\theta_{mn})}{|m||n|} \quad (3.18)$$

D'où la formule finale de la mesure de similarité *MahCosine* :

$$S_{MahCosine}(u, v) = \frac{mn}{|m||n|}, D_{MahCosine}(u, v) = - S_{MahCosine}(u, v) \quad (3.19)$$

Où $D_{MahCosine}(u,v)$, est la mesure de distance équivalente. On peut enfin noter qu'il s'agit principalement de la covariance entre les vecteurs dans l'espace de Mahalanobis.

3.4.3. Choix de la distance de similarité

Dans la communauté de la reconnaissance des formes, le choix d'une distance de similarité est souvent expliqué et argumenté par rapport à l'espace d'attributs et aux points de données utilisés. Nous avons remarqué entre autre, avec les mêmes vecteurs attributs Eigenfaces, que la distance Euclidienne fournit de meilleurs résultats d'identification des individus qu'avec la distance de Manhattan. Ces performances nous ont poussés à utiliser la distance Euclidienne dans ce dernier processus.

4. La méthode générale de reconnaissance

Pour reconnaître une personne, le PCA utilise le même calcul afin d'obtenir une combinaison de ratio en utilisant les mêmes eigenfaces. Cette combinaison est comparée à celle obtenu pour les personnes de la liste d'apprentissage.

La combinaison de ratio la plus proche entre la personne en phase de reconnaissance et les personnes de la liste d'apprentissage nous donne le résultat de cette comparaison.

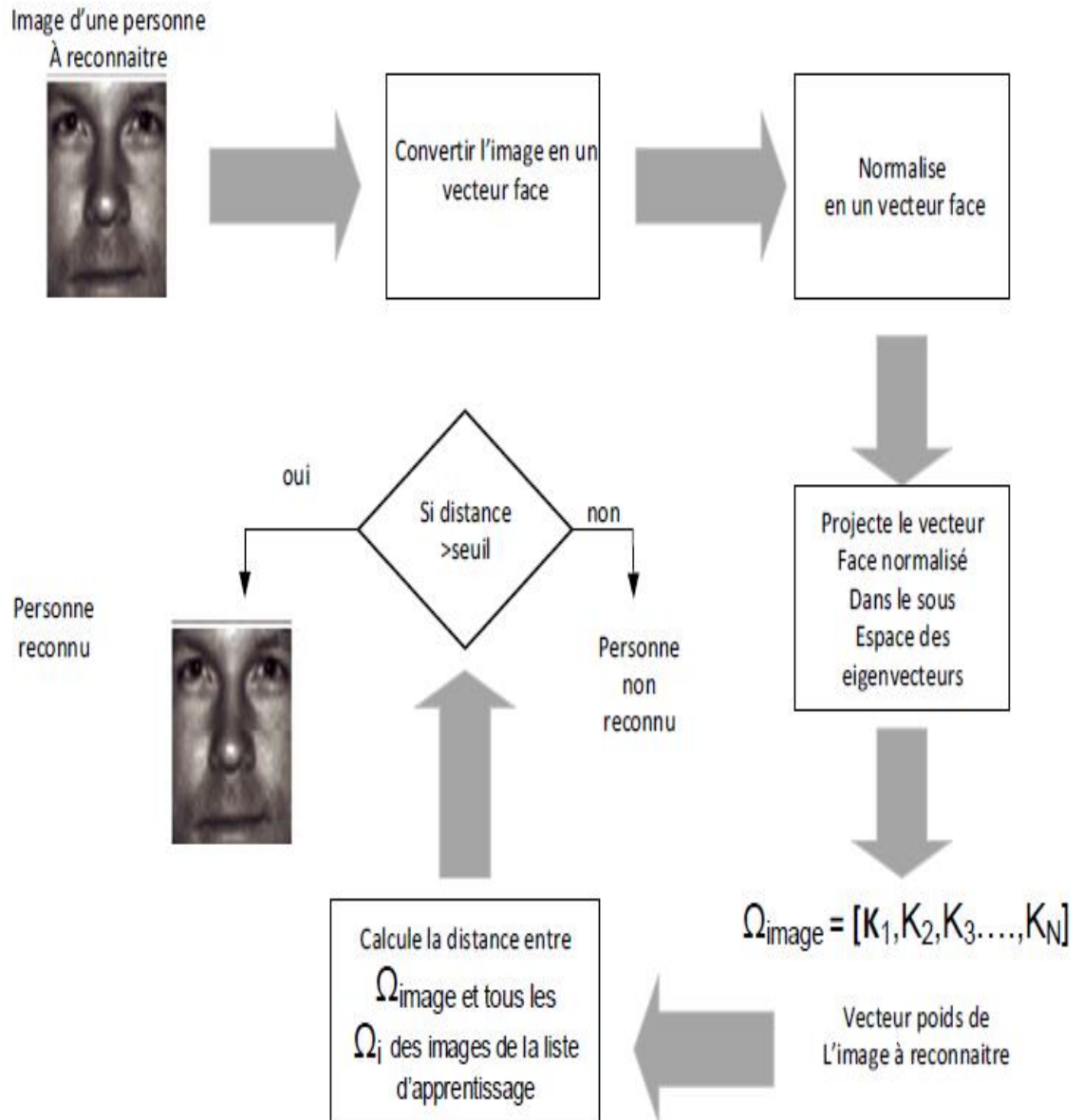


Figure 3.21 : La méthode générale de reconnaissance.

5. Organigramme détaillé de l'approche Eigenface

Notre organigramme se divise en trois parties : la première est le prétraitement suivi de la phase d'apprentissage et se termine par la phase d'identification où la distance euclidienne est retenue pour le calcul de la différence entre les poids de l'image à identifier et les images de base de données puis le programme affiche la plus proche.

5.1. Organigramme du prétraitement :

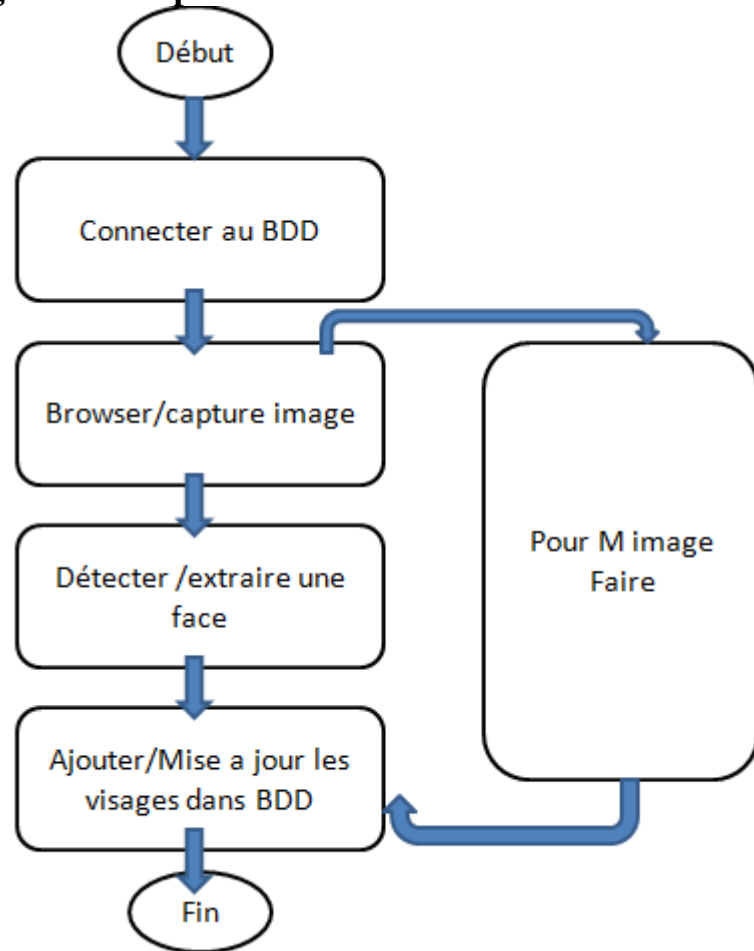


Figure 3.22 : Organigramme du prétraitement.

4.2. Organigramme de la phase d'apprentissage

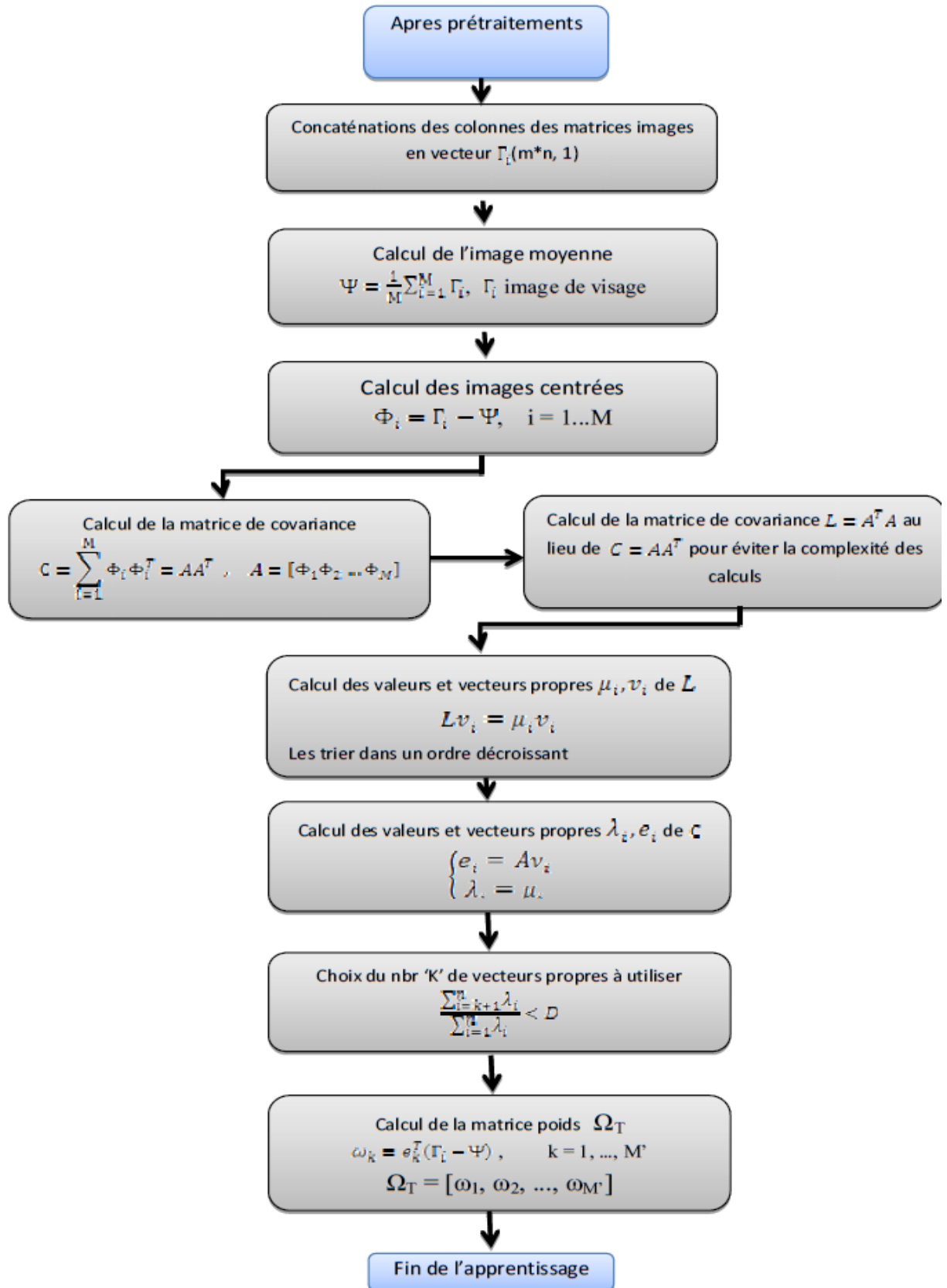


Figure 3.23 : Organigramme de la phase d'apprentissage.

5.3. Organigramme de la phase d'identification :

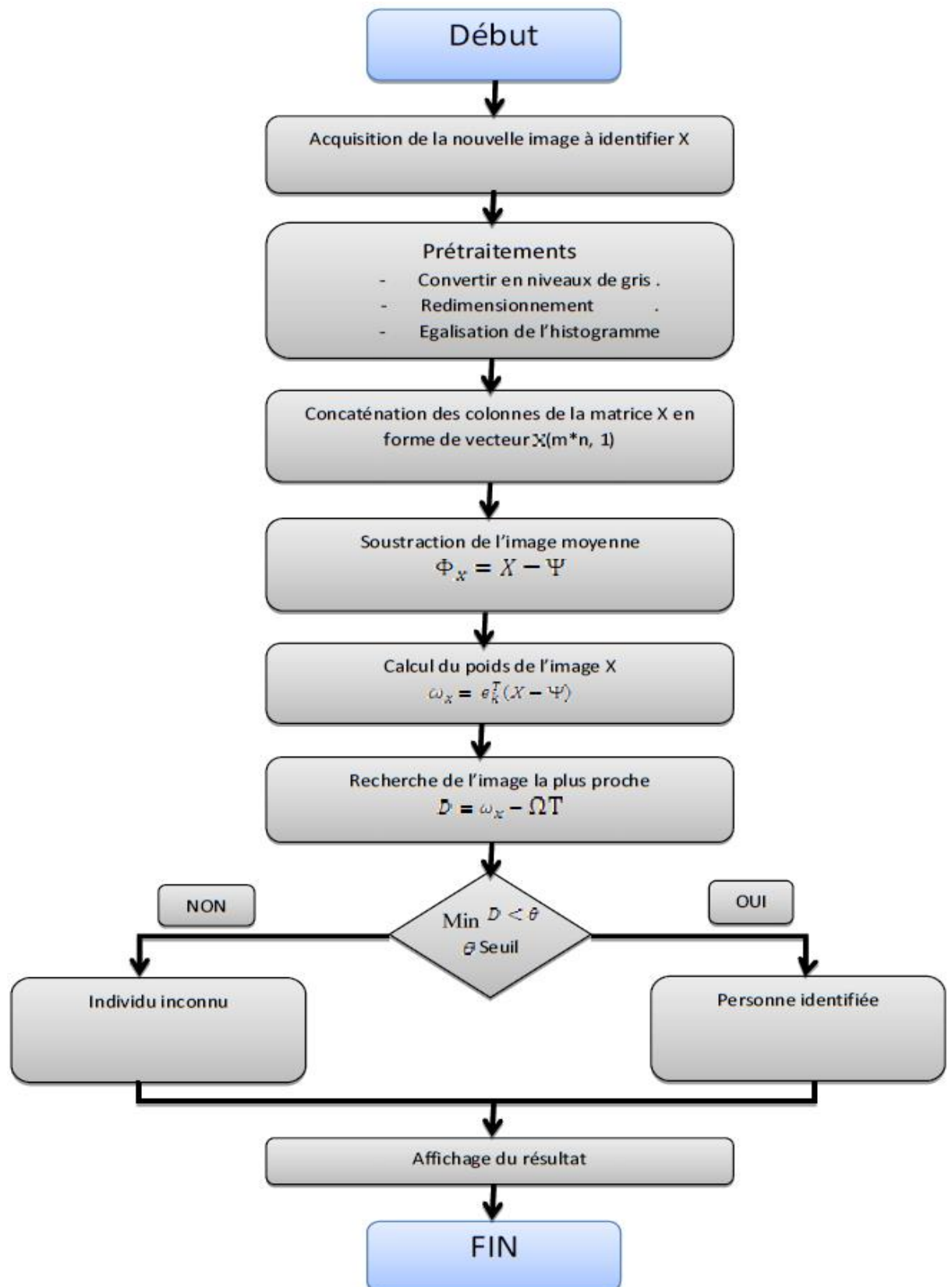


Figure 3.24 : Organigramme de la phase d'identification.

6. Conclusion

Ce chapitre a été consacré en premier lieu à la présentation de la méthode de détection de visage « Viola et Jones », ensuite la méthode reconnaissance faciale choisie qui est « Eigenface », et ensuite, nous avons décrit les différentes étapes de l'implémentation du système.

Dans le chapitre suivant, nous testerons le système avec la BDD Yale.

Chapitre 4

Test et résultat

1. Introduction

Ce chapitre représente les résultats des tests effectués avec la méthode des « Eigenface » sur la base de données Yale. Rappelons que notre travail consiste à concevoir un système d'identification biométrique de personnes par reconnaissance faciale se basant sur une analyse en composante principale qui peut être utilisée pour simplifier un ensemble de données, en réduisant sa dimension. Plusieurs étapes sont nécessaires, l'étape d'extraction des caractéristiques est la plus importante car les performances du système en dépendent (résultats et robustesse, un temps de latence acceptable pour des applications « temps réel »).

2. Environnement du travail

Dans cette section, nous présenterons les environnements matériel et logiciel de notre travail.

2.1. Environnement matériel

Afin de mener à bien ce projet, il a été mis à notre disposition un ensemble de matériels dont les caractéristiques sont les suivantes :

Un ordinateur HP avec les caractéristiques suivantes :

- Processeur : Intel® Core™ i3-3110M CPU @ 2.40Ghz 2.40Ghz
- RAM: 4.00 Go de RAM
- Disque Dur: 500 Go
- OS: Microsoft Windows eight.

2.2. Outils de développement

Nous avons eu recours lors de l'élaboration de notre système à Visual c# d'outils de développement Visual studio 2012 avec l'utilisation de la bibliothèque Emgu cv que nous présenterons ci-dessous.

2.2.1. Visual studio 2012

Visual Studio est un ensemble complet d'outils de développement permettant de générer des [applications web ASP.NET](#), des [services web XML](#), des applications bureautiques et des applications mobiles. [Visual Basic](#), [Visual C++](#), [Visual C#](#) utilisent tous le même [environnement de développement intégré](#) (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du [framework .NET](#), qui fournit un accès à

des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à [Visual Web Developer](#).



	Développeur	Microsoft	
	Dernière version	2015 (14.0.23107.0) (20 juillet 2015)	
	Environnement	Microsoft Windows	
	Langue	Multilingue	
	Type	Environnement de développement intégré	
	Licence	Propriétaire	
2.2.2.	Site web	msdn.microsoft.com/vstudio	Emgu cv
Emgu			CV est une croix
plate-forme			.Net wrapper

pour la bibliothèque OpenCV de traitement d'image. Permettre fonctions OpenCV à appeler des langages compatibles .NET tels que C #, VB, VC ++, IronPython etc, peut être compilée par Visual Studio, Xamarin Studio Unity, il peut fonctionner sur Windows, Linux, Mac OS X, iOS, Android Phone et Windows.

3. Présentation de l'application

On présente dans cette section les différents aspects du système de reconnaissance, notre système est composé de deux sous système connectés entre eux avec une base de données Access.

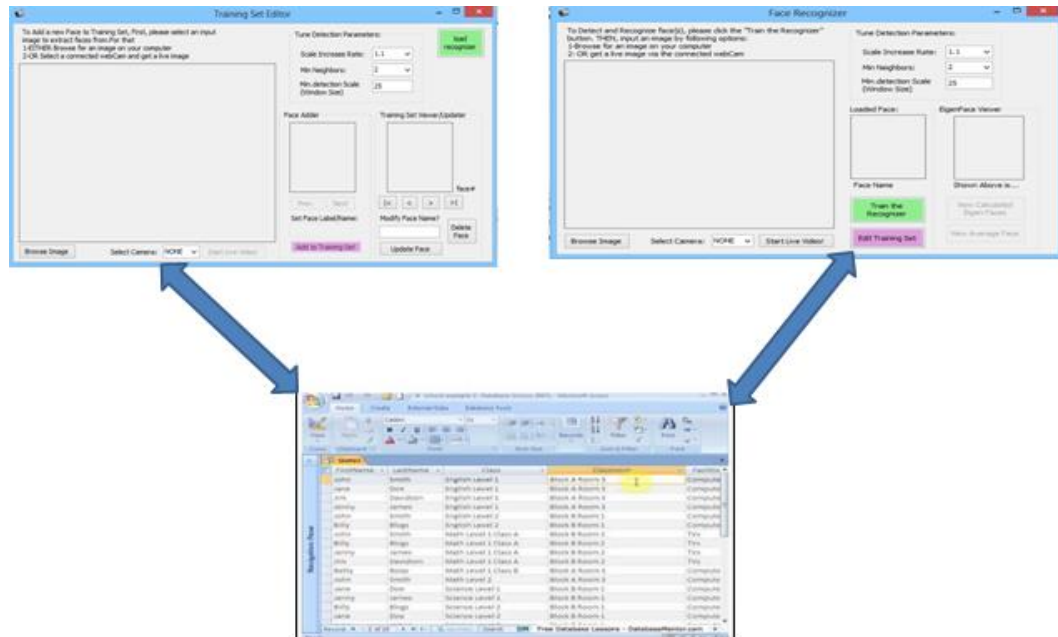


Figure 4.1 : les deux sous systèmes de l'application.

3.1. Sous système du prétraitement

C'est une interface destinée aux utilisateurs, elle est simple pour le prétraitement des images (détection de visage, ajouter, supprimer...etc.).

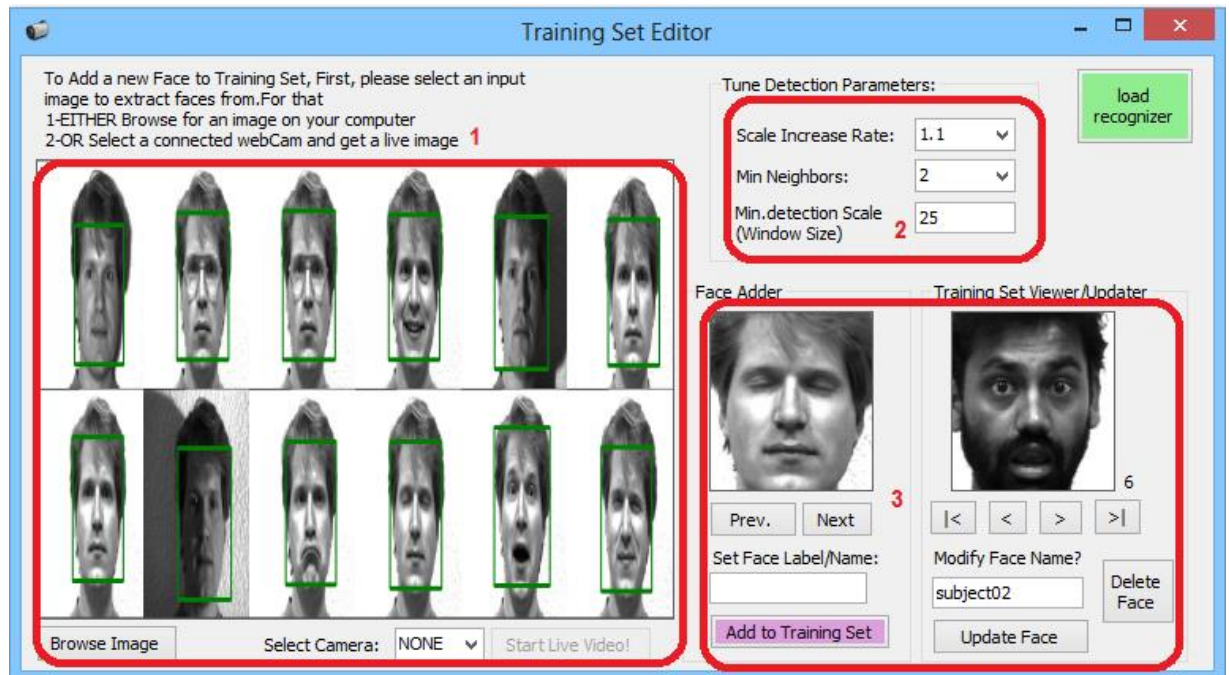


Figure 4.2 : Sous système du prétraitement.

On observe qu'il y a 3 phases principales dans cette interface pour le prétraitement des images.

3.1.1. La détection et l'extraire des visages

D'ajouter un nouveau visage, sélectionnez une image d'entrée pour extraire des visages, soit rechercher une image sur votre ordinateur ou sélectionnez une webcam connectée et obtenir une image en direct avec 3 paramètres de détection.

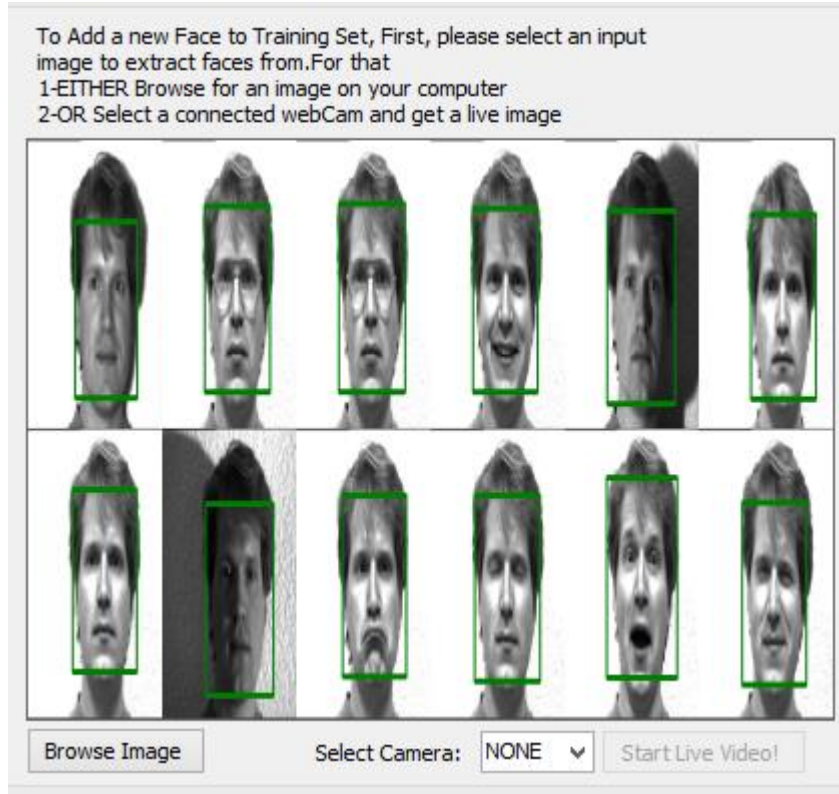


Figure 4.3 : extraire des visages.

3.1.2. Les paramètres de détection

Il y a plusieurs paramètres que vous pouvez régler pour régler le détecteur de visage pour notre application.

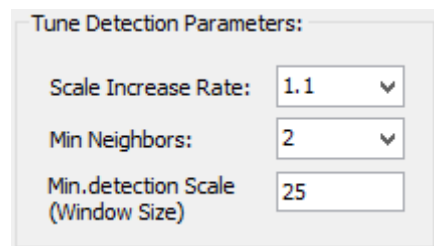


Figure4.4 : Paramètres de détection.

1. Haar Cascade

Un classificateur utilise des données stockées dans un fichier XML pour décider comment classer chaque emplacement d'image. Alors, naturellement, haar aura besoin de fichier XML pour charger des données formés à partir.

Le premier paramètre dans l'appel à DetectHaarCascade () est le fichier XML à partir duquel, les données formé est chargé pour le classificateur Haar.

Nous utilisons haarcascade_frontalface_alt_tree.xml parce qu'il avait les meilleurs résultats, pour autant que nous ayons essayé, pour la détection de face frontale: détecte les visages maximaux possibles en une seule fois avec moins de fausses détections!

La declation en c#

```
Haar = new HaarCascade("haarcascade_frontalface_alt_tree.xml");
```

2. Taux d'augmentation de l'échelle (scale increase rate)

Le second paramètre dans l'appel à DetectHaarCascade () indique la rapidité de EmguCv pour augmenter l'échelle de la détections de visage à chaque passage qu'il fait sur une image, permet le détecteur plus rapide (en exécutant moins de passes), mais si elle est trop élevée, vous pouvez sauter trop rapidement entre les échelles et les visages miss. La valeur par défaut dans EmguCv est de 1,1, en d'autres termes, l'augmentation de l'échelle par un facteur de 1,1 (10%) à chaque passage, ce paramètre peut avoir une valeur de 1,1, 1,2, 1,3 ou 1,4.



Figure 4.5 : Scale increase rate.

3. Minimum Neighbors Threshold

Le troisième paramètre dans l'appel à DetectHaarCascade (), est le «Le seuil minimum-voisins» qui définit le niveau de coupure pour le rejet ou le maintien des groupes de rectangle comme "face" ou non, basé sur le nombre de détections premières sont dans le groupe. Les plages de valeurs de ce paramètre de 0 à 4.

Après nos testes nous concluons que les meilleures paramètres de détection sont scale increse rete = 1.1 et le Minimum Neighbors Threshold = 1.

Un exemple de 12 visages :

Les paramètres scale increse rete = 1.1 et le Minimum Neighbors Threshold = 1, le système détecté les 12 visages.

Les paramètres scale increse rete = 1.4 et le Minimum Neighbors Threshold = 4, le système détecté juste 10 visages.

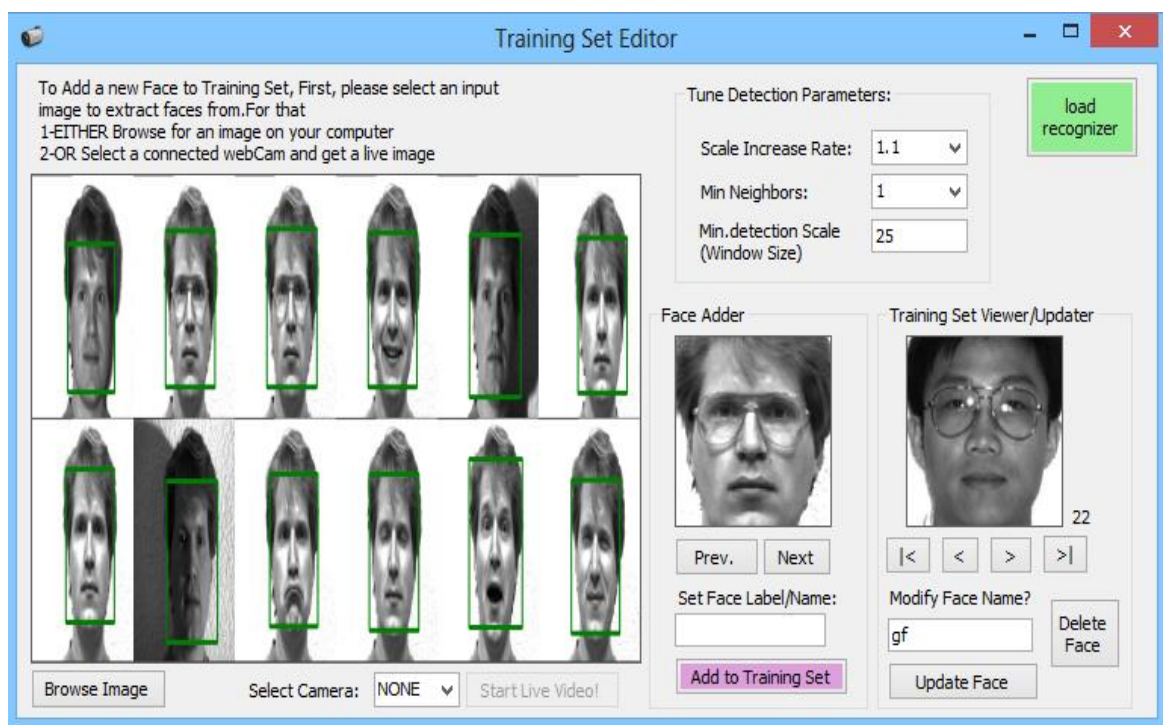


Figure 4.6: Scale increase rate = 1.1, Min neighbors = 1.

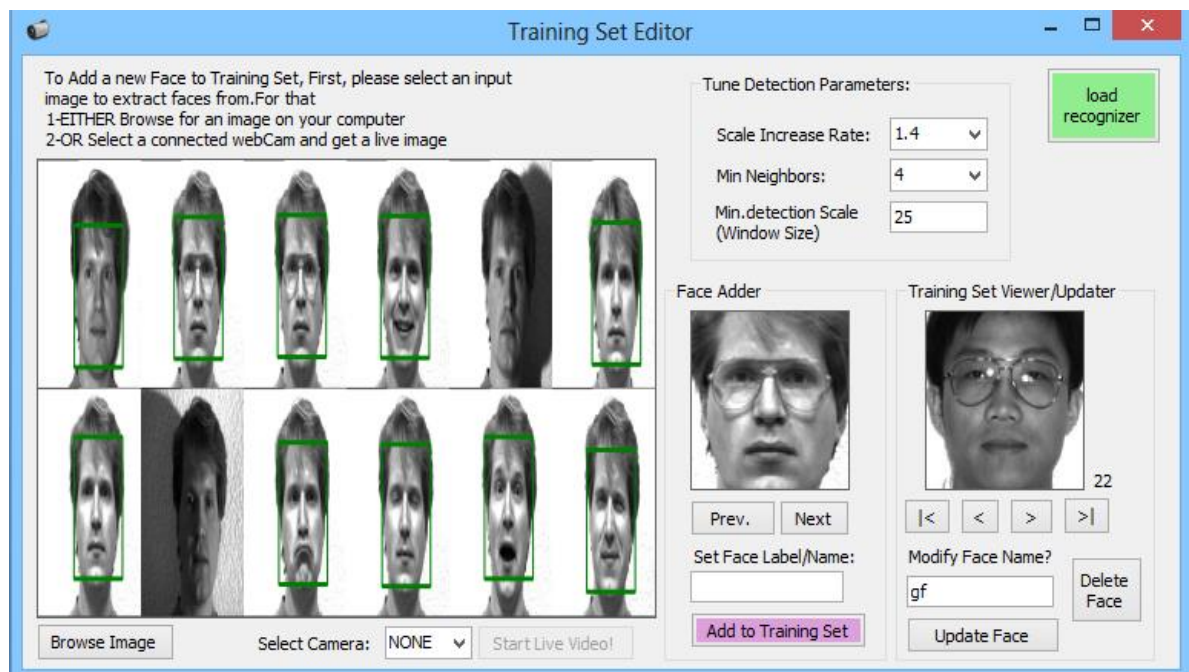


Figure 4.7: Scale increase rate = 1.4, Min neighbors = 4.

La figure 4.8 présente le taux d'erreur de la détection des visages avec les paramètres : scale increase 1.1 et de 1 à 4 de min neighbors.

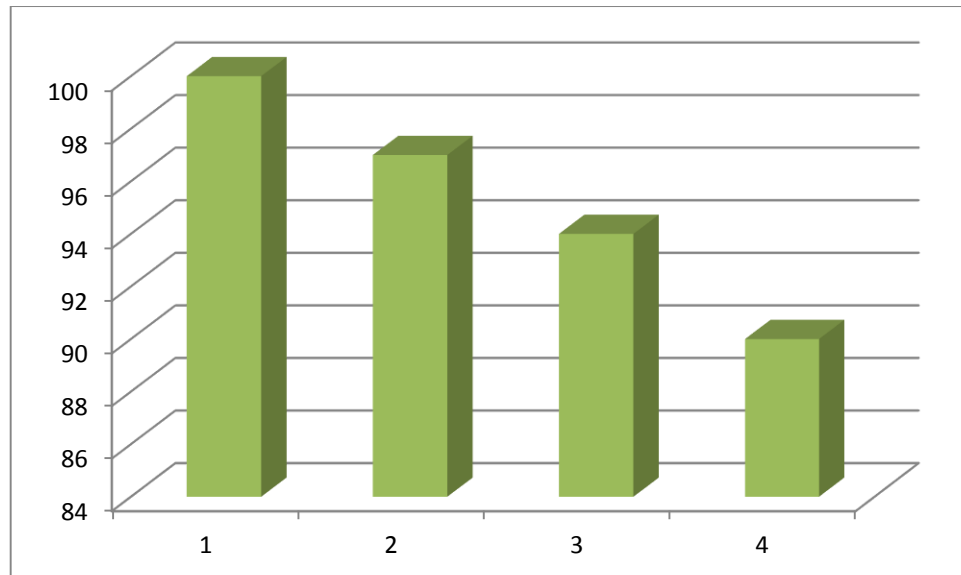


Figure 4.8 : Taux d'erreur de détection des visages.

Avec les tests sur la base de données des images Yale de détection le taux d'erreur de la détection des visages est 100%.

3.1.3. Gestion des visages

La troisième phase de nous premier sous système est la gestion des visages détectés (parcourir les visages détectés ou bien les visages de notre BDD, ajouter, supprimer, mise à jour etc.).

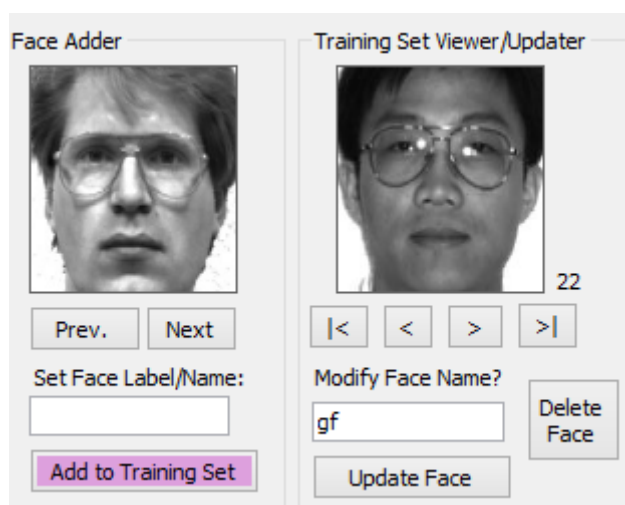


Figure 4.9 : Gestion des visages.

3.2. Sous système de la reconnaissance facial

C'est une interface destinée aux utilisateurs, elle est simple et permet d'illustrer les principaux processus du système de reconnaissance (identification, visage moyen, calculer les Eigenfaces) et pour charger le sous système de reconnaissance on click sur le bouton Load Recognizer dans le sous système du prétraitement.

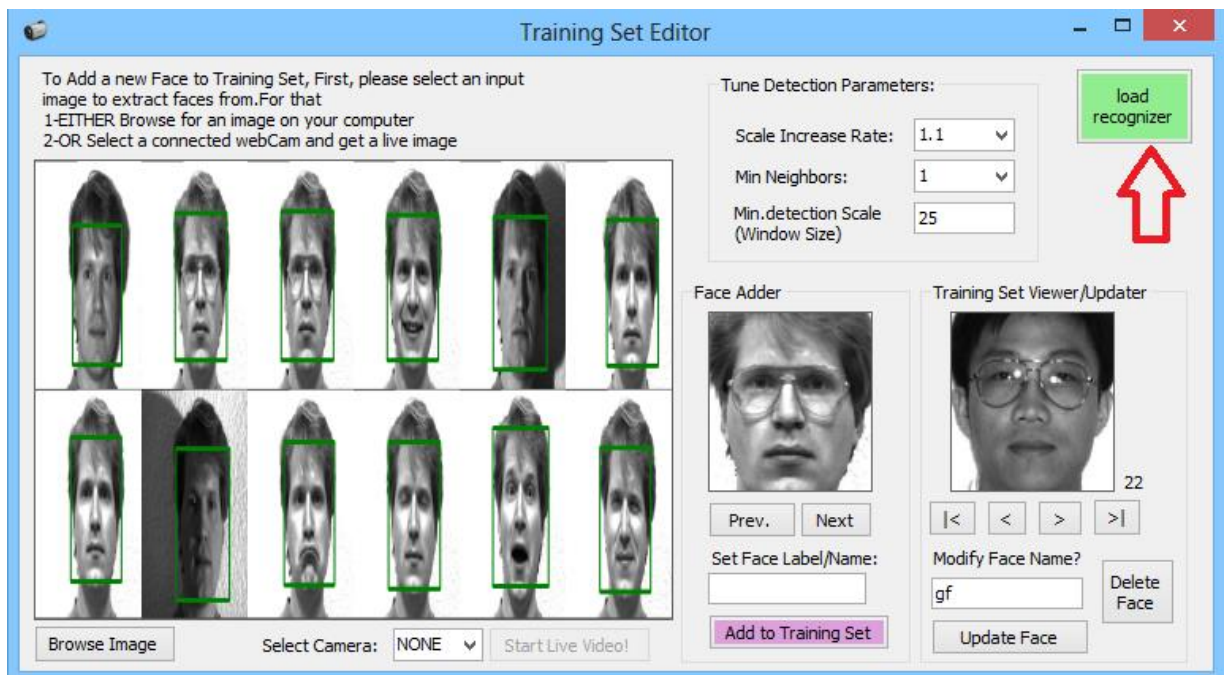


Figure 4.10 : Charger le système de reconnaissance.

Et voilà l'interface de notre sous système de reconnaissance facial.

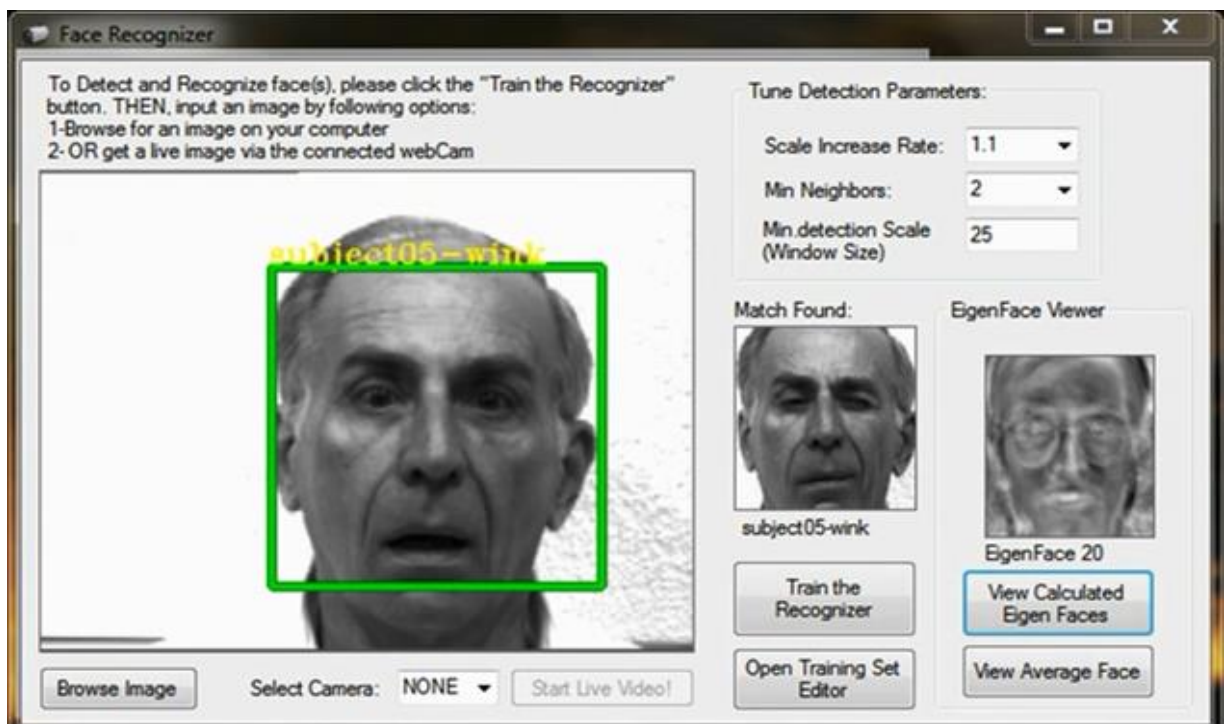


Figure 4.11 : Sous système de reconnaissance facial.

Pour détecter et reconnaître les visages cliquez sur le bouton Train Recognizer puis saisissez une image en navigation pour une image sur l'ordinateur ou d'obtenir une image en direct via webcam.

4. Fonctionnement du système

Notre système permet d'illustrer le processus d'authentification. Nous allons décrire dans ce qui suit les différents événements qui s'y passent lors de ce processus.

Il comporte deux phases : la phase d'apprentissage pendant laquelle les modèles sont construits et la phase d'identification (phase de comparaison).

4.1. Phase d'apprentissage

Dans le cadre d'une méthode supervisée (les modèles doivent être connus à priori) pour l'identification des personnes par le visage, un modèle pour la représentation de chaque classe est construit, à partir des 165 images de la base de données de Yale nous utilisons 135 visages pour cette phase d'apprentissage. Le modèle peut être défini comme suit :

-Le vecteur correspondant à la distance minimale entre le vecteur recherché et l'ensemble de l'échantillon d'apprentissage, ceci permet d'aboutir à un modèle ayant les caractéristiques les plus proches du vecteur recherché au sens de la distance utilisée. Une fois les modèles des individus définis, l'opération d'identification peut s'effectuer, voici les étapes de la phase d'apprentissage :

1- Collecte des M images faciales et construction de la matrice T de taille M , par concaténation des colonnes des images faciales. Prétraitement des images collectées.

2- Calcul du visage moyen en sommant les colonnes de la matrice T et divisant le vecteur résultant par le nombre d'image d'entrée (M).

3- Soustraction du visage moyen de la matrice T pour obtenir la matrice A ; où chaque élément représente la variance des valeurs d'intensité de chaque pixel.

4-Calcul de la matrice \hat{C} , $\hat{C}=A^T A$.

5-Calcul des vecteurs propres \hat{C} et les triés dans un ordre descendant selon les valeurs propres associées.

6-Calcul des vecteurs propres de la matrice de covariance C et obtention des visages propres en multipliant les vecteurs propres \hat{C} par la matrice A .

7- Choix des K meilleurs valeurs propres et les vecteurs propres associés.

8-Détermination du poids des images d'entrée en projetant chaque image dans l'espace visage.

9-chaque visage est maintenant représenté par un vecteur qui est utilisé pour reconstruire les images. Et en fin sauvegarde du calcul du visage moyen, des eigenfaces et du poids des images.

4.2. Phase d'identification

Après la phase d'apprentissage le tour de la phase d'identification, nous utilisons 30 visages pour la phase de test.

- Prétraitement de l'image d'entrée et soustraction du visage moyen.

- Détermination du poids de l'image d'entrée par la projection de celle-ci dans l'espace visage en multipliant le vecteur résultant de l'étape (1) par les eigenfaces de la base de données.

- Récupération du vecteur le plus proche au sens de la distance utilisée (euclidienne, Manhattan...).

- Affichage du résultat (identité trouvée).

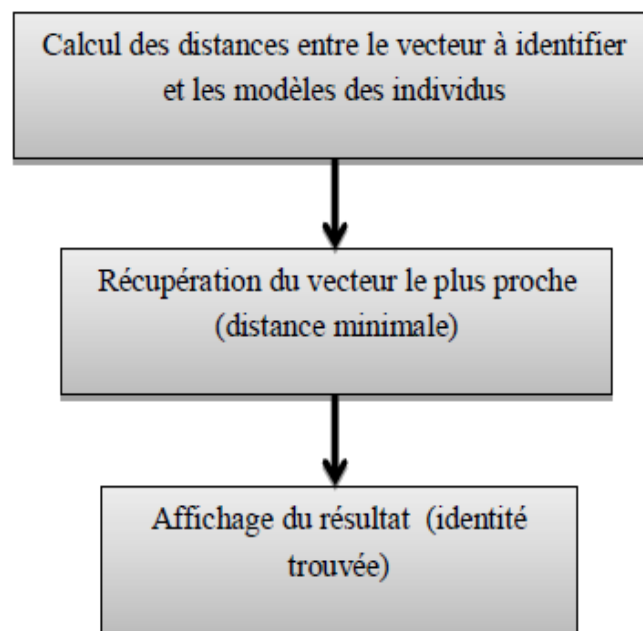


Figure 4.12 : Phase d'identification.

Après les tests notre sous système de reconnaissance facial est identifié 29 visages sur 30, touché de taux d'erreur de 96.66%.

5. Avantages du système

Dans ce chapitre, nous avons décrit les étapes de la conception du système de reconnaissance, et l'interface développée pour lui. En vue de ce qui a été fait dans la conception et la réalisation, on peut dégager plusieurs avantages

- Interface simple et rapide

C'est une interface que toute personne ayant peu de connaissances sur la biométrie peut utiliser.

- Système très interactif

C'est un système très interactif. En effet, en plus du fait qu'il permette l'identification d'individus avec de bons taux, il permet aussi de voir le visage moyen et les « eigenfaces ».

- Système performant

Le système est performant car les taux de reconnaissance sont bons 100% pour la détection des visages et 96.66% pour l'identification.

6. Conclusion

Dans ce chapitre la présentation de nos système, nous avons testé notre système de reconnaissance sur la base de données Yale, avec l'utilisation de l'algorithme Viola et Jones pour la détection des visages et nous avons touché 100% de taux d'erreur et L'algorithme PCA (méthode eigenfaces) pour la phase d'identification avec de taux d'erreur de 96.66%, avec la mention de quelques avantages de nos systèmes.

Conclusion générale

La biométrie est un domaine à la fois passionnant et complexe. Elle tente, par des outils mathématiques souvent très évolués, de faire la distinction entre des individus, nous obligeant à travailler dans un contexte de très grande diversité. Cette diversité se retrouve également dans le nombre considérable d'algorithmes qui ont été proposés en reconnaissance faciale.

Dans ce mémoire, nous sommes intéressés au problème de la reconnaissance faciale. Notre travail consiste à la mise au point d'un algorithme Viola et Jones de détection de visage et d'un algorithme robuste destiné à reconnaître un individu par son visage en utilisant la méthode « Eigenface » qui se base sur une analyse en composante principale (ACP). L'ACP est une méthode mathématique qui peut être utilisée pour simplifier un ensemble de données, en réduisant sa dimension. Elle est utilisée pour représenter efficacement les images de visages, qui peuvent être approximativement reconstruites à partir d'un petit ensemble de poids et d'une image de visage standard.

Une analyse des différentes techniques de reconnaissance développées au cours de ces dernières années a été présentée, et cela pour mettre en évidence les particularités ainsi que les avantages et les inconvénients de chacune d'entre elles.

L'ACP reste une méthode efficace et simple pour gérer ce type de problème. C'est pour toutes ces raisons que nous avons opté pour cette approche de reconnaissance du visage.

Nous estimons avoir réalisé un système répondant à l'objectif que nous sommes fixés au départ, à savoir la mise en œuvre d'un système permettant de détecter des visages avec de taux d'erreur de 0% et la reconnaissance des individus de taux d'erreur 96.66%.

En guise de perspectives, une extension de ce travail peut être réalisée en intégrant un système d'acquisition des images pour éviter que ces opérations d'identification et de vérification soient de simples simulations. Par ailleurs, il serait aussi intéressant d'appliquer, la méthode développée dans ce mémoire, dans un système de sécurité biométrique utilisant l'acquisition vidéo.

Bibliographie

- [1] John D. Woodward, Jr., Christopher Horn, Julius Gatune, and Aryn Thomas, "Biometrics A Look at Facial Recognition", documented briefing by RAND Public Safety and Justice for the Virginia State Crime Commission, 2003.
- [2] Florent Perronnin, Jean-Luc Dugelay, « Introduction à la biométrie : Authentification des individus par traitement audio-vidéo », Institut Eurocom, Multimedia Communications Department, Revue Traitement du signal, Vol. 19, N° 4, 2002.
- [3] <http://www.biometricgroup.com>.
- [4] S. Liu, M. Silveanu, « A practical Guide to Biometric Security Technology », IEEE Computer Society, IT Pro-Security, Janvier-Février 2001.
- [5] A. K. Jain, L. Hong, S. Pankanti, « Biometrics : Promising Frontiers for Emerging Identification Market », Communications of the ACM, pp. 91-98, February 2000.
- [6] C. Fredouille, J. Mariethoz, C. Jaboulet, J. Hennebert, J.-F. Bonastre, C. Mokbel, F. Bimbot, « Behavior of a Bayesian Adaptation Method for Incremental Enrollment in Speaker Verification », International Conference on Acoustics, Speech, and Signal Processing, pp. 1197-1200, Istanbul, Turquie, 5-9 Juin 2000.
- [7] Y. Hori, M. Kusaka, and T. Kuroda. "A 0.79mm² 29mW Real-Time Face Detection Core". Symposium on VLSI Circuits Digest of Technical Papers, pp. 188-189, June 2006.
- [8] Algorithme Viola et Jones <http://www.firediy.fr/article-18.html>.
- [9] W. Bledsoe. "The model method in facial recognition". Tech. Rep. PRI: 15, Stanford University, Palo Alto, CA, USA, 1964.
- [10] Base de données Yale <http://vision.ucsd.edu/content/yale-face-database>.
- [11] L. Sirovich and M. Kirby. "Low-dimensional procedure for the characterization of human faces". Journal of Optical Society of America, Vol. 4, No. 3, pp. 519–524, 1987.
- [12] M. Turk and A. Pentland. "Eigenfaces for recognition". Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71–86, 1991.
- [13] Y. Jian, D. Zhang, A. Frangi, and J.-Y. Yang. "Two-Dimensional PCA : A New Approach to Appearance-Based Face Representation and Recognition". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 1, pp. 131–137, January 2004.
- [14] P. Belhumeur, J. Hespanha, and D. Kriegman. "Eigenfaces vs. Fisherfaces : Recognition Using Class Specific Linear Projection". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, pp. 711–720, 1997.
- [15] M. Bartlett, J. Movellan, and T. Sejnowski. "Face recognition by independent component analysis". IEEE Transactions on Neural Networks, Vol. 13, No. 6, pp. 1450–1464, 2002.
- [16] G. Brown, S. Yamada, and T. Sejnowski. "Independent Component Analysis at the Neural Cocktail Party". Trends in Neuroscience, Vol. 24, pp. 54–63, 2001.
- [17] B. Schölkopf, A. Smola, and K.-R. Müller. "Nonlinear component analysis as a kernel eigenvalue problem". Neural Computation, Vol. 10, No. 5, pp. 1299–1319, 1998.
- [18] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, and K.-R. Müller. "Fisher Discriminant Analysis With Kernels". In : Neural Networks for Signal Processing IX, pp. 41–48, 1999.
- [19] C. Liu and H. Wechsler. "A Unified Bayesian Framework for Face Recognition". In : Proceedings of the IEEE International Conference on Image Processing, pp. 151–155, 1998.
- [20] G. Guo, S. Li, and K. Chan. "Face Recognition by Support Vector Machines". In : Proceedings of the IEEE International Conference on Automatic Face and Gesture

- Recognition, pp. 196–201, 2000.
- [21] T. Ahonen, A. Hadid, and M. Pietikainen. Face Recognition with Local Binary Patterns. 2004.
 - [22] S. Arca, P. Campadelli, and R. Lanzarotti. “A Face Recognition System Based On Automatically Determined Facial Fiducial Points”. Pattern Recognition, Vol. 39, No. 3, pp. 432–443, 2006.
 - [23] P. Penev and J. Atick. “Local features analysis : A general statistical theory for object representation”. Neural Systems, Vol. 7, No. 3, pp. 477–500, 1996.
 - [24] V. Perlibakas. “Face recognition using Principal Component Analysis and Log-Gabor Filters”. March 2005.
 - [25] Y. Adini, Y. Moses, S. Ullman, Face recognition: The problem of compensating for changes in illumination direction. IEEE Trans. Patt. Anal. Mach. Intell. 19, 721–732, 1997.
 - [26] D. Blackburn, M. Bone, P. J Phillips. “Face recognition vendor test 2000”. Tech. rep. <http://www.frvt.org>, 2001.
 - [27] R. Gross, J. Shi, J. Cohn. Quo Vadis Face Recognition?. Third Workshop on Empirical Evaluation Methods in Computer Vision, December, 2001.
 - [28] A.M Martínez, R. Banavente. The AR face database. Tech. Report 24 CVC Barcelone, Espagne, June 1998.
 - [29] D. Blackburn, M. Bone, P. J Phillips. “Face recognition vendor test 2000”. Tech. rep. <http://www.frvt.org>, 2001.
 - [30] Ming-Hsuan Yang, David J. Kriegman et Narendra Ahuja. Detecting faces in images : A survey. Dans IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 24(1), pages 34–58, 2002.
 - [31] H. A. Rowley, S. Baluja, et T. Kanade, "Neural Network based Face Detection", IEEE Trans, Pattern Anal. Mach, Intell, 23-38, January 1998
 - [32] C. Kotropoulos and I. Pitas. Rule-Based Face Detection in Frontal Views. Proc. Int'l Conf. Acoustics, Speech and Signal Processing, vol. 4, pp. 2537-2540, 1997.
 - [33] T. Kanade. “Picture Processing by Computer Complex and Recognition of Human Faces,” PhD thesis, Kyoto Univ., 1973.
 - [34] J.Yang, D. Zhang, A.F. Frangi, J.Yang, Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 131–137,2004.
 - [35] P. Sinha, “Processing and Recognizing 3D Forms,” PhD thesis, Massachusetts Inst. of Technology, 1995.
 - [36] A. Yuille, P. Hallinan, and D. Cohen, “Feature Extraction from Faces Using Deformable Templates,” Int’l J. Computer Vision, vol. 8, no. 2, pp. 99-111, 1992.
 - [37] (Fr) C# et .NET - Gérard Leblanc - 2002 - Éditions Eyrolles - (ISBN 2-212-11066-9)
 - [38] (Fr) C# et .NET Version 2 - Gérard Leblanc - 2006 - Éditions Eyrolles - (ISBN 2-212-11778-7)
 - [39] (Fr) C# Tête la première - Andrew Stellman, Jennifer Greene - 2008 - Éditions Digit Books - (ISBN 978-2-81500-001-7).
 - [40] “Principal components analysis”.
Document available at :
http://en.wikipedia.org/wiki/Principal_components_analysis.
 - [41] A. S. Tolba, A.H. El-Baz, and A.A. El-Harby, “Face Recognition: A Literature Review”, INTERNATIONAL JOURNAL OF SIGNAL PROCESSING VOLUME 2 NUMBER 2 2005 ISSN 1304-4494.
 - [42] Cherng Jye Liou, “A Real Time Face Recognition System”, DSP/IC Design Lab,

- Department of Electrical Engineering, National Taiwan University, June 1997.
- [43] site web <http://fewtutorials.bravesites.com/>
 - [44] “Linear discriminant analysis”.
Document available at : http://en.wikipedia.org/wiki/Linear_discriminant_analysis.
 - [45] Dr. Andrzej Drygajlo, ELE 233, “BIOMETRICS”.
Document available at : <http://scgwww.epfl.ch/courses>.
 - [46] Juwei Lu, Kostantinos N. Plataniotis, and Anastasios N. Venetsanopoulos, “Face Recognition Using LDA-Based Algorithms”, IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 14, NO. 1, JANUARY 2003.
 - [47] M. Bartlett and H. Lades and T. Sejnowski, “Independent component representations for face recognition”, in T. Rogowitz, B. & Pappas, editor, Proceedings of the SPIE Symposium on Electronic Imaging: Science and Technology; Human Vision and Electronic Imaging III, volume 3299, San Jose, CA, January 1998. SPIE Press. 1998.
 - [48] Wen Gao, Shiguang Shan, Face Verification for Access Control, “Biometrics Solutions for Authentication in an E-World”, Edited by David Zhang, Kluwer Academic Publishers, Chapter 13, pp339-376, 2002.
 - [49] “Visual studio 2010”
<http://www.01net.com/telecharger/windows/Programmation/creation/fiches/49822.html>.
 - [48] L. Png, “Morphological Shared-Weight Neural Network For Face Recognition”, A dissertation submitted to the University of Manchester Institute of Science and Technology for the degree of MSc, August 2004.
 - [49] “Support vector machine”.
Document available at : http://en.wikipedia.org/wiki/Support_vector_machine.
 - [50] “Visual Object Recognition using Template Matching”.
Document available at : <http://www.araa.asn.au/acra/acra2004/papers/cole.pdf>.
 - [51] Cotrell and Fleming. Face recognition using unsupervised feature extraction. In Proceedings of International conference on Neural Network, pages 322–325, Paris, France, 1990.
 - [52] S. Lawrence, C.L. Giles, C. Tsoi, and A.D. Back. Face recognition : A convolutional neural network approach. IEEE Transactions on Neural Networks, 8 :98–113, 1997.
 - [53] S.H. Lin, S.Y. Kung, and Lin. Face recognition / detection by probabilistic decision-based neural network. IEEE Transactions on Neural Networks, 8(1) :114–132, 1997
 - [54] S.C. Chen, J. Liu, Z.-H. Zhou, MakinG. FLDA applicable to face recognition with one sample per person, Pattern Recognition 37 (7), 1553–1555, 2004.
 - [55] B.S. Manjunath, R. Chellappa, C.V.D. Malsburg, A feature based approach to face recognition, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 373–378, 1992.
 - [56] H.S. Le, H. Li. Recognizing frontal face images using hidden Markov models with one training image per person, Proceedings of the 17th International Conference on Pattern Recognition (ICPR04), vol. 1, pp. 318–321, 2004.

ملخص

الهدف من هذا العمل المقدم في إطار أطروحة لنيل شهادة الماستر في الإعلام الآلي هو تحديد الهوية (التعرف) باستخدام صور الوجه وذلك باستخدام خوارزمية فيولا وجونز لتحديد الوجه وطريقة Eigenfaces (PCA), على قاعدة البيانات يال باستخدام visual studio 2012.

الكلمات المفتاحية : تحديد الهوية, صور الوجه, فيولا وجونز, Eigenfaces (PCA), يال.

Abstract

The objective of the work presented in the context of obtaining a master memory for computing, is studding identification (recognition) of identity using facial images using the Viola and Jones algorithm for face detection and Eigenface method (PCA) for facials recognition on a standard Yale database face with Visual studio 2012.

Key words: identification, Facial images, Viola and Jones, Eigenface (PCA), Yale.

Résumé

L'objectif du travail présenté dans le cadre d'un mémoire pour obtention du master en l'informatique, est d'étudier l'identification (reconnaissance) de l'identité employant des images faciales on utilisant l'algorithme Viola et Jones pour la détection de visage et la méthode Eigenface(PCA) pour la reconnaissance facials sur une base de données standard de visage Yale avec Visual studio 2010.

Mots clés : l'identification, images faciales, Viola et Jones, Eigenface(PCA), Yale.

