

Sommaire

Table des figures **ix**

Liste des tableaux **xi**

Chapitre 1

Introduction **1**

- 1.1 Historique 2
- 1.2 Domaines d'application 2
- 1.3 Problématique 3
- 1.4 Plan du rapport 4

Chapitre 2

Méthodes de détection et de reconnaissance des visages **5**

- 2.1 Détection de visages 5
 - 2.1.1 Algorithme de Viola et Jones 7
- 2.2 Prétraitement 12
 - 2.2.1 Normalisation Géométrique 13
 - 2.2.2 Normalisation Photométrique 13
- 2.3 Reconnaissance de visages 14
 - 2.3.1 Méthodes globales 15
 - 2.3.2 Méthodes locales 16
 - 2.3.3 Méthodes hybrides 18
- 2.4 Reconnaissance par Eigenfaces 18
- 2.5 Bases de données 23
 - 2.5.1 La base XM2VTS 23
 - 2.5.2 La base FERET 24
 - 2.5.3 La base AR 25

2.5.4	La base CVL	26
2.5.5	La Base de donnée ORL	26
2.5.6	Base Labeled Faces in the Wild (LFW)	30

Chapitre 3	
Expérimentation	31

3.1	Environnement du travail	31
3.2	OpenCV	31
3.3	Implémentation	32
3.3.1	Comment détecter un visage à l'aide du détecteur OpenCV	32
3.3.2	Comment faire le prétraitement des images faciales pour la reconnaissance faciale :	34
3.3.3	Comment Eigenfaces peut être utilisée pour la reconnaissance des visages :	34
3.3.4	Implémentation du système de reconnaissance faciale Hors ligne (en ligne de commande)	34
3.3.5	Comment utiliser le système FaceRec en ligne de commande :	37
3.3.6	La mise en œuvre de reconnaissance faciale en temps réel à partir d'une caméra :	37
3.3.7	Comment utiliser le système FaceRec de webcam en temps réel :	38

Chapitre 4	
Conclusion	

Bibliographie	45
----------------------	-----------

Table des figures

1.1	Schéma général d'un système de reconnaissance des visages.	3
2.1	Exemple de détection de visage.	6
2.2	caractéristiques pseudo-haar à seulement deux caractéristiques	9
2.3	caractéristiques pseudo-haar avec différentes orientations.	10
2.4	Illustration de l'architecture de la cascade : les fenêtres sont traitées séquentiellement par les classifieurs, et rejetées immédiatement si la réponse est négative (F).	12
2.5	Histogramme de l'image avant et après égalisation	13
2.6	Quand les visages ne sont pas vus dans leur état naturel, la capacité du système visuel humain à les distinguer est dégradée.	14
2.7	EBGM.	17
2.8	visage moyen	20
2.9	Quelques faces propres	21
2.10	Exemples d'images de la base XM2VTS.	24
2.11	Extrait de la base Color FERET : Les images sont transformées en niveau de gris.	25
2.12	Extrait de la base AR. Ensembles des vues collectées pour un individu de la base.	25
2.13	Images faciales extraites de la base CVL.	26
2.14	Base de donnée ORL	27
2.15	Exemple de changements d'orientations du visage	27
2.16	Exemple de changements d'éclairage	27
2.17	Exemple de changements d'échelle	28
2.18	Exemple de changements des expressions faciales	28
2.19	Exemple de port de lunettes	28
2.20	Exemple de changements de coiffure et de port de barbe	29
2.21	Exemple d'individus de différents ages races et sexes	29
2.22	Paires de la base LFW aligned	30
3.1	Base de données au format XML	33
3.2	Organigramme du prétraitement	35
3.3	Exemple d'image moyenne, le premier visage propre et le dernier visages propres d'une collection de 30 images de 4 personnes	36

Table des figures

3.4	Cela dit au programme que la personne 1 est nommé <code>personA</code> , et les 3 photos du visage prétraitées de <code>personA</code> sont dans le dossier <code>data/s1</code> , et personne 2 est appelé <code>personB</code> avec 3 images dans le dossier <code>data/s2</code> . . .	36
3.5	Capture d'écran de l'application en fonctionnement.	39
3.6	Reconnaissance de visage.	40

Liste des tableaux

1.1	Les biométries couramment utilisées.	2
2.1	Comparaison des propriétés des caractéristiques locales et des caractéristiques globales.	18

1

Introduction

Identifier une personne à partir de son visage est une tâche aisée pour les humains. En est-il de même pour une machine ? Ceci définit la problématique de la reconnaissance automatique de visages, qui a engendré un grand nombre de travaux de recherche au cours des dernières années.

Le visage peut être considéré comme une donnée biométrique¹. Une donnée biométrique est une donnée qui permet l'identification d'une personne sur la base de ce qu'il est (caractéristiques physiologiques ou comportementales)^{2, 3}. Les indices biométriques physiologiques sont des traits biologiques/chimiques innés, alors que les indices biométriques comportementaux sont associés à des comportements appris ou acquis. Le tableau 1.1 fournit une liste des indices biométriques couramment utilisés.

Les données biométriques sont devenues des données incontournables pour le problème de l'identification sécurisée et de la vérification de personnes. Les méthodes d'identification ou de vérification d'identité basées sur une donnée biométrique offrent certains avantages par rapport aux méthodes basées sur un mot de passe ou un code PIN.

Premièrement, les données biométriques sont des données individuelles, alors que les mots de passe peuvent être utilisés ou volés par quelqu'un d'autre que l'utilisateur autorisé.

Deuxièmement, une donnée biométrique est très pratique car il n'y a rien à porter ou à mémoriser.

Troisièmement, les technologies biométriques deviennent de plus en plus précises avec un coût en constante diminution.

Parmi les éléments biométriques figurant dans le tableau 1.1, seules les données visage et parole appartiennent à la fois aux deux catégories physiologique et comportementale.

Cela veut dire entre autre que la biométrie faciale nous permet d'exploiter de nombreuses informations relatives à une personne. Dans la vie quotidienne, le visage est probablement le trait biométrique le plus utilisé par les humains afin de reconnaître les autres, le visage a de grands avantages par rapport aux autres biométries, parce qu'il est naturel, et facile à acquérir.

1. National Institute of Standards and Technologies (NIST), <http://www.nist.gov/biometrics>.

2. Biometric Catalog. <http://www.biometriccatalog.org>

3. Biometric Consortium. <http://www.biometrics.org>.

Biométries physiologiques	ADN, visage, empreinte, forme de la main, rétine, odeur, voix
Biométries comportementales	Démarche, visage, voix, écriture, signature

TABLE 1.1 – Les biométries couramment utilisées.

1.1 Hitorique

La reconnaissance faciale automatique est un concept relativement nouveau. Le premier système semi-automatisé de la reconnaissance faciale a été développé dans les années 1960, il nécessite à l'administrateur de localiser les yeux, les oreilles, le nez et la bouche sur la photo et de saisir les distances calculées et les ratios à un point de référence commun, qui ont ensuite été comparés aux données de référence.

Dans les années 1970, Goldstein, Harmon et Lesk [1] ont utilisé 21 marqueurs spécifique tels que la couleur des cheveux et l'épaisseur de la lèvre pour automatiser la reconnaissance. Le problème avec ces deux premières solutions, c'est que les mesures et les emplacements ont été calculés manuellement.

En 1988, Kirby et Sirovich [2] ont appliqué l'analyse en composantes principales (ACP), une technique standard de l'algèbre linéaire. Cela a été considéré en quelque sorte comme une étape importante car elle a montré qu'au moins une centaine de valeurs ont été nécessaires pour coder convenablement et avec précision une image alignée et normalisée.

En 1991, Turk et Pentland [3] ont découvert que lorsque vous utilisez la technique Eigenfaces (ACP), l'erreur résiduelle peut être utilisée pour détecter un visage dans une image, une découverte qui a permis la reconnaissance faciale automatique en temps réel. Bien que l'approche était quelque peu limitée par des facteurs environnementaux, elle a néanmoins créé un intérêt significatif pour promouvoir le développement des technologies de la reconnaissance faciale automatique. Cette technologie a été mise en essai en janvier 2001 lors de la finale du championnat de football américain SUPER BOWL en capturant des images de surveillance puis comparées à une base de données numérique [4].

Aujourd'hui la reconnaissance faciale est utilisée dans plusieurs domaines.

1.2 Domaines d'application

La sécurité reste le domaine d'application principal. Dans ce domaine, la reconnaissance des visages est nécessaire à l'identification et à l'authentification. Un bon exemple de cet usage est l'aéroport de Francfort (Allemagne) où on utilise la reconnaissance pour automatiser le contrôle des passagers [5].

L'analyse des vidéos capturées par les systèmes de caméras externes gérés par les grandes villes est un autre exemple, un suspect peut être identifié [6].

Les agences de publicité sont en train de développer des panneaux d'affichage intelligents, qui grâce à la reconnaissance du visage des personnes qui passent adaptent le contenu affiché [7].

Google et Facebook ont implémenté des algorithmes pour identifier les personnes dans leurs bases de données de photos. En octobre 2013, Facebook a abandonné la reconnais-

sance faciale en Europe suite aux critiques visant cette technologie. Il est interdit de détecter les visages des gens sur les photos, cela met en péril leur vie privée. Aux États-Unis, cette technologie continue à se développer, elle est même utilisée par les agences gouvernementales (FBI) ou par des entreprises privées [8].

Au Brésil, la police se prépare à la coupe du monde de football de 2014 à sa façon : elle utilisera des lunettes équipées d'une caméra capable de filmer 400 images par seconde et les comparer avec une base de données numérique de 13 millions de photos [9].

La nouveauté dans la reconnaissance faciale arrive grâce au développement des nouvelles caméras de type 3D. Ces caméras obtiennent de meilleurs résultats que les caméras classiques, parce qu'elles acquièrent une image tridimensionnelle de chaque visage.

1.3 Problématique

Le problème de la reconnaissance de visages peut être formulé comme suit : étant données une ou plusieurs images d'un visage, la tâche est de trouver ou de vérifier l'identité d'une personne par comparaison de son visage à l'ensemble des images de visage stockées dans une base de données.

En général, un système de biométrie faciale est constitué de deux modules : un module de segmentation de visages (détection ou localisation de visage), et un module de reconnaissance qui se déroule en trois étapes : normalisation ou prétraitement, extraction de caractéristiques faciales, classification (Figure 1.1).

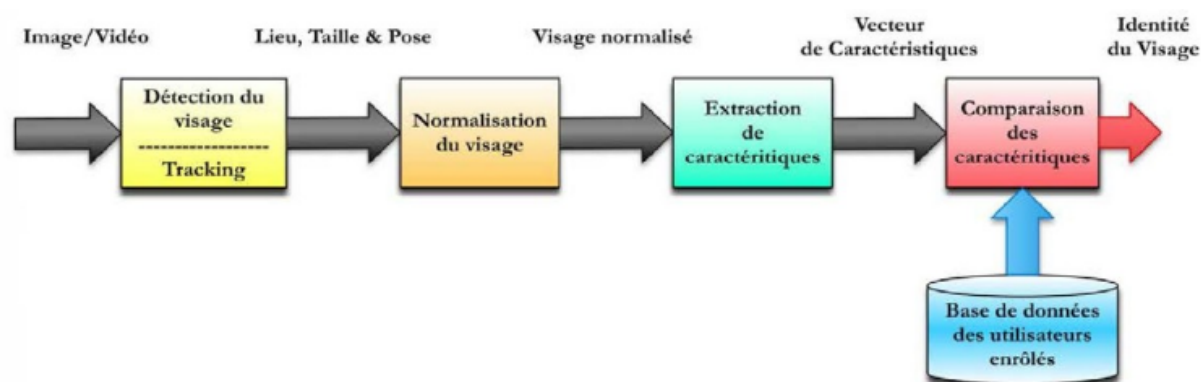


FIGURE 1.1 – Schéma général d'un système de reconnaissance des visages.

La reconnaissance faciale pose de nombreux défis car les visages sont des objets déformable 3D. Nous nous limitons dans ce travail à une reconnaissance à partir d'une image 2D de visage en environnements non contraints (contexte de vidéosurveillance). De tels systèmes doivent pouvoir s'affranchir des problèmes suivants :

- **Influence des variations de la pose :**

Les changements d'orientation et les changements de l'angle d'inclinaison du visage engendrent de nombreuses modifications d'apparence dans les images collectées. Les rotations en profondeur engendrent l'occlusion de certaines parties du visage comme pour les vues de trois-quarts. D'autre part, elles amènent des différences de

profondeur qui sont projetées sur le plan 2D de l'image, provoquant des déformations qui font varier la forme globale du visage. Ces déformations qui correspondent à l'étirement de certaines parties du visage et la compression d'autres régions font varier aussi les distances entre les caractéristiques faciales.

- **Influence des changements d'éclairage :**

L'intensité et la direction d'éclairage lors de la prise de vue influent énormément sur l'apparence du visage dans l'image. En effet, dans la plupart des applications courante, des changements dans les conditions d'éclairage sont inévitables, notamment lorsque les vues sont collectées à des heures différentes, en intérieur ou en extérieur. Etant donnée la forme spécifique d'un visage humain, ces variations d'éclairage peuvent y faire apparaître des ombres accentuant ou masquant certaines caractéristiques faciales.

- **Influence des expressions faciales :**

Les visages sont des éléments non rigides. Les expressions faciales véhiculant des émotions, combinés avec les déformations induites par la parole, peuvent produire des changements d'apparence importants, et le nombre de configurations possibles est trop important pour que celles-ci soient décrites *in extenso* de façon réaliste.

- **Influence des occultations :**

Un visage peut être partiellement masqué par des objets ou par le port d'accessoires tels que lunettes, un chapeau, une écharpe. Les occultations peuvent être intentionnelles ou non. Dans le contexte de la vidéosurveillance, il peut s'agir d'une volonté délibérée d'empêcher la reconnaissance. Il est clair que la reconnaissance sera d'autant plus difficile que peu d'éléments discriminants seront simultanément visibles.

1.4 Plan du rapport

Notre travail s'intéresse au problème général de la reconnaissance de visages en contexte de vidéosurveillance. Ainsi, dans ce mémoire, nous aurons à concevoir et à implémenter une application permettant la reconnaissance des visages des personnes à partir d'une séquence vidéo, la sortie est une valeur qui représente le score de comparaison des caractéristiques des visages extraits de la base avec celle qui existe dans la séquence vidéo.

Dans le chapitre 2, nous évoquerons l'état de l'art des algorithmes de reconnaissance de visage 2D, en particulier, nous détaillons les méthodes de détection qui ont connu une avancée notable. A la fin de ce chapitre nous présenterons quelques bases de données disponibles sur le marché, et également celle qui a été utilisée dans ce mémoire. Dans la chapitre 3, la conception du système de reconnaissance faciale sera discuté.

2

Méthodes de détection et de reconnaissance des visages

2.1 Détection de visages

La détection de visage 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 [10, 11].

L'intérêt de la localisation faciale va au-delà de l'application de ce présent mémoire. Quelques appareils photos numériques récents emploient la détection de visages pour la mise au point automatique. Elle est également recherchée dans le domaine des économies d'énergie ; les télévisions et les ordinateurs peuvent économiser l'énergie en réduisant l'éclat. Le système peut identifier la direction de visage de l'utilisateur de TV. Quand l'utilisateur ne regarde pas l'écran, l'éclat de TV est abaissé, et quand le visage revient à l'écran, l'éclat est augmenté [12].

Les premières difficultés rencontrées par les méthodes de détections de visages sont les variations de pose (vue de profil ou de face), d'expression, de rotation du visage, d'âge et d'illumination. Ce dernier type de difficulté pouvant être surmonté par un prétraitement de normalisation et de compensation de l'illumination, ce sera présenté à la fin de ce chapitre.

Méthodes de détections réparties en catégories

Une classification des méthodes de localisation faciale a été proposée par Yang *et al*[11]. Les méthodes sont divisés en quatre catégories. Ces catégories peuvent se chevaucher si



FIGURE 2.1 – Exemple de détection de visage.

un algorithme peut appartenir à deux ou plusieurs catégories. Cette classification peut être faite comme suit :

- ***Knowledge-based methods :***

Ces méthodes se basent sur la connaissance des différents éléments qui constituent un visage et des relations qui existent entre eux. Ainsi, les positions relatives de différents éléments clés tels que la bouche, le nez et les yeux sont mesurées pour servir ensuite à la classification 'visage' 'nonvisage' chez Chiang *et al.* [13]. Le problème dans ce type de méthode est qu'il est difficile de bien définir de manière unique un visage. Si la définition est trop détaillée, certains visages seront ratés tandis que si la description est trop générale, le taux de faux positifs montera en flèche.

- ***Feature invariant approaches :***

Ces approches utilisent les éléments invariants aux variations d'illumination, d'orientation ou d'expression tels que la texture ou la signature de couleur de la peau pour la détection.

- ***Template matching methods :***

Des modèles caractéristiques d'un visage entier ou de sous-partie de visage (bouche,

oeil, nez) sont créés. La localisation se fait ensuite sur base de la corrélation de ces modèles avec les candidats [21].

- ***Appearance-based methods*** :

Ces méthodes utilisent le même principe que présenté au point précédent mais se basent sur des modèles appris à partir d'un ensemble d'essai. Ces méthodes présentent l'avantage de s'exécuter très rapidement mais demandent un long temps d'entraînement. Les méthodes appartenant à cette catégorie ont montré de bons résultats par rapport aux trois autres types de méthodes [14]. On peut citer parmi celles-ci, la méthode basée sur les réseaux de neurones de Rowley *et al.* [15], la méthode de Schneiderman et Kanade [16] basée sur un classifieur de Bayes naïf ainsi que le fameux algorithme de Viola et Jones [17] fonctionnant en temps réel, et ce dernier sera détaillé ci-dessous.

2.1.1 Algorithme de Viola et Jones

Une avancée majeure dans le domaine a été réalisée par les chercheurs Paul Viola et Michael Jones en 2001 [17]. Ces derniers ont proposé une méthode basée sur l'apparence (*Appearance-based methods*)⁴.

La méthode de Viola et Jones est une méthode de détection d'objet dans une image numérique, elle fait partie des toutes premières méthodes capables de détecter efficacement et en temps réel des objets dans une image. Inventée à l'origine pour détecter des visages, elle peut également être utilisée pour détecter d'autres types d'objets comme des voitures ou des avions. La méthode de Viola et Jones est l'une des méthodes les plus connues et les plus utilisées, en particulier pour la détection de visages et la détection de personnes.

En tant que procédé d'apprentissage supervisé, la méthode de Viola et Jones nécessite de quelques centaines à plusieurs milliers d'exemples de l'objet que l'on souhaite détecter, pour entraîner un classifieur. Une fois son apprentissage réalisé, ce classifieur est utilisé pour détecter la présence éventuelle de l'objet dans une image en parcourant celle-ci de manière exhaustive, à toutes les positions et dans toutes les tailles possibles.

Considérée comme étant l'une des plus importantes méthodes de détection d'objet, la méthode de Viola et Jones est notamment connue pour avoir introduit plusieurs notions reprises ensuite par de nombreux chercheurs en vision par ordinateur, à l'exemple de la notion d'image intégrale ou de la méthode de classification construite comme une cascade de classifieurs boostés.

Cette méthode bénéficie d'une implémentation sous licence BSD dans OpenCV, la bibliothèque utilisée dans notre application.

Principe

La méthode de Viola et 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

4. L'article Robust Real-Time Face Detection [24], est cité plus de 1600 fois selon Google Scholar.

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.

Il existe d'autres méthodes mais celle de Viola et 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.

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.

Dans le cas présent, Viola et Jones ont entraîné leur classifieur à l'aide d'une banque d'images du MIT. Il en résulte un classifieur sensible aux visages humains. Il se présente sous la forme d'un fichier XML.

Dans l'absolu, on serait en mesure de détecter n'importe quel signe distinctif à partir d'un classifieur entraîné à cela.

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.

Prenons un exemple : Voici 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. 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

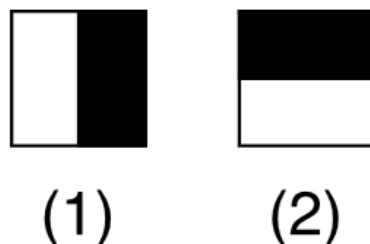


FIGURE 2.2 – caractéristiques pseudo-haar à seulement deux caractéristiques

généralisé, Viola et Jones donnant l'exemple d'une fenêtre de taille 24×24 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. Malheureusement, le calcul de ces caractéristiques de manière *classique* coûte cher en terme de ressources processeur, c'est là qu'interviennent les images intégrales.

L'image intégrale

Pour calculer rapidement et efficacement ces caractéristiques sur une image, les auteurs proposent également une nouvelle méthode, qu'ils appellent *image intégrale*. C'est une représentation sous la forme d'une image, de même taille que l'image d'origine, elle contient en chacun de ses points la somme des pixels situés au-dessus et à gauche du pixel courant. Plus formellement, l'image intégrale ii au point (x, y) est définie à partir de l'image i par :

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.1)$$

Le calcul de la somme des valeurs des pixels appartenant à une zone rectangulaire s'effectue donc en accédant seulement à quatre pixels de l'image intégrale : soit un rectangle $ABCD$ dont les sommets sont nommés dans le sens des aiguilles d'une montre en commençant par le sommet supérieur gauche et soit x la valeur sous la représentation intégrale d'un sommet X du rectangle ($X \in \{A, B, C, D\}$). La somme des valeurs des pixels appartenant à $ABCD$ est, quelle que soit sa taille, donnée par $c - b - d + a$.

Une caractéristique de Haar étant une combinaison linéaire de tels rectangles $ABCD$, son calcul se fait alors en un temps indépendant de sa taille.

Algorithme d'apprentissage basé sur Adaboost

Le deuxième élément clé de la méthode de Viola et Jones est l'utilisation d'une méthode de boosting afin de sélectionner les meilleures caractéristiques. Le boosting est un principe qui consiste à construire un classifieur *fort* à partir d'une combinaison pondérée de classifieurs *faibles*, c'est-à-dire, donnant en moyenne une réponse meilleure qu'un tirage aléatoire. Viola et Jones adaptent ce principe en assimilant une caractéristique à un

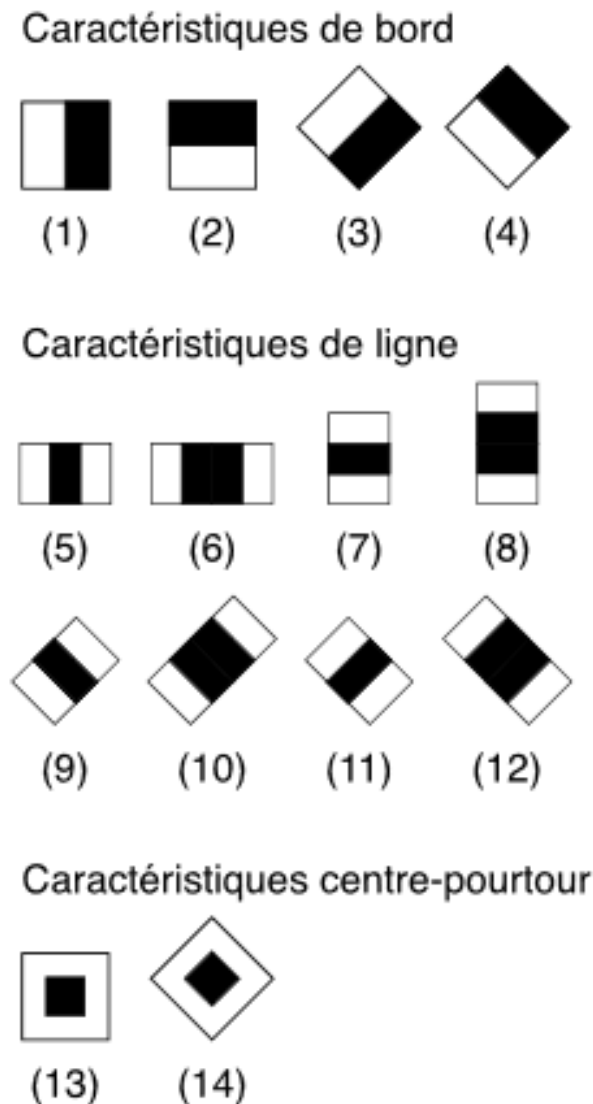


FIGURE 2.3 – caractéristiques pseudo-haar avec différentes orientations.

classifieur faible, en construisant un classifieur faible qui n'utilise qu'une seule caractéristique. L'apprentissage du classifieur faible consiste alors à trouver la valeur seuil de la caractéristique qui permet de mieux séparer les exemples positifs des exemples négatifs. Le classifieur se réduit alors à un couple (caractéristique, seuil).

L'algorithme de boosting utilisé est en pratique une version modifiée d'AdaBoost, qui est utilisée à la fois pour la sélection et pour l'apprentissage d'un classifieur *fort*. Les classifieurs faibles utilisés sont souvent des arbres de décision. Un cas remarquable, fréquemment rencontré, est celui de l'arbre de profondeur 1, qui réduit l'opération de classification à un simple seuillage.

L'algorithme est de type itératif, à nombre d'itérations déterminé. À chaque itération, l'algorithme sélectionne une caractéristique, qui sera ajoutée à la liste des caractéristiques sélectionnées aux itérations précédentes, et le tout va contribuer à la construction du classifieur fort final. Cette sélection se fait en entraînant un classifieur faible pour toutes

les caractéristiques et en sélectionnant celui avec l'erreur la plus faible sur l'ensemble d'apprentissage. L'algorithme tient également à jour une distribution de probabilité sur l'ensemble d'apprentissage, réévaluée à chaque itération en fonction des résultats de classification. En particulier, plus de poids est attribué aux exemples difficiles à classer, c'est-à-dire ceux dont l'erreur est élevée. Le classifieur *fort* final construit par AdaBoost est composé de la somme pondérée des classifieurs sélectionnés.

Plus formellement, on considère un ensemble de n images (x_1, \dots, x_n) et leurs étiquettes associées (y_1, \dots, y_n) , qui sont telles que $y_i = 0$ si l'image x_i est un exemple négatif et $y_i = 1$ si x_i est un exemple de l'objet à détecter. L'algorithme de boosting est constitué d'un nombre T d'itérations, et pour chaque itération t et chaque caractéristique j , on construit un classifieur faible h_j . Idéalement, le but est d'obtenir un classifieur h qui prédise exactement les étiquettes pour chaque échantillon, c'est-à-dire $y_i = h(x_i) \forall i \in \{1, \dots, n\}$. En pratique, le classifieur n'est pas parfait et l'erreur engendrée par ce classifieur est donnée par :

$$\epsilon_j = \sum_{i=1}^n w_i |h_j(x_i) - y_i|, \quad (2.2)$$

les w_i étant les poids associés à chaque exemple et mis à jour à chaque itération en fonction de l'erreur obtenue à l'itération précédente. On sélectionne alors à l'itération t le classifieur h_t présentant l'erreur la plus faible : $\epsilon_t = \min(\epsilon_j)$.

Le classifieur fort final $h(x)$ est construit par seuillage de la somme pondérée des classifieurs faibles sélectionnés :

$$h(x) = \begin{cases} 1 & \text{si } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{sinon} \end{cases} \quad (2.3)$$

Les α_t sont des coefficients calculés à partir de l'erreur ϵ_t :

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t} \quad (2.4)$$

Cascade de classifieurs

La méthode de Viola et Jones est basée sur une approche par recherche exhaustive sur l'ensemble de l'image, qui teste la présence de l'objet dans une fenêtre à toutes les positions et à plusieurs échelles. Cette approche est cependant extrêmement coûteuse en calcul. L'une des idées-clés de la méthode pour réduire ce coût réside dans l'organisation de l'algorithme de détection en une cascade de classifieurs. Appliqués séquentiellement, ces classifieurs prennent une décision d'acceptation ; la fenêtre contient l'objet et l'exemple est alors passé au classifieur suivant, ou de rejet ; la fenêtre ne contient pas l'objet et dans ce cas l'exemple est définitivement écarté. L'idée est que l'immense majorité des fenêtres testées étant négatives (c.-à-d. ne contiennent pas l'objet), il est avantageux de pouvoir les rejeter avec le moins possible de calculs. Ici, les classifieurs les plus simples, donc les plus rapides, sont situés au début de la cascade, et rejettent très rapidement la grande

majorité des exemples négatifs. Cette structure en cascade peut également s'interpréter comme un arbre de décision dégénéré, puisque chaque nœud ne comporte qu'une seule branche [25].

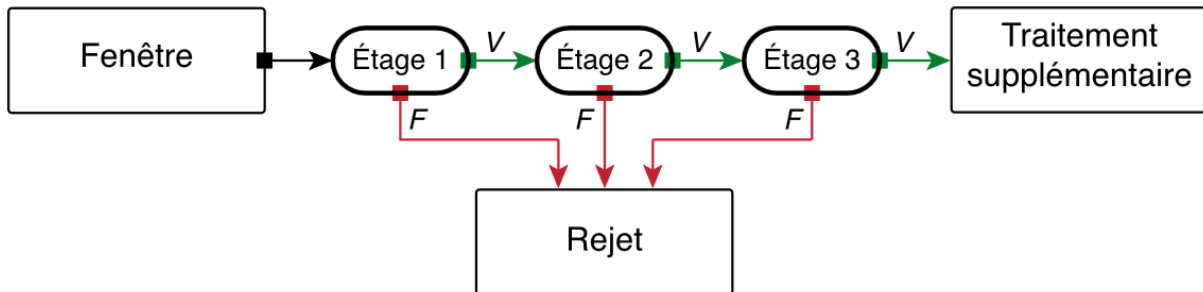


FIGURE 2.4 – Illustration de l'architecture de la cascade : les fenêtres sont traitées séquentiellement par les classifieurs, et rejetées immédiatement si la réponse est négative (F).

Une des limitations de la méthode de Viola et Jones est son manque de robustesse à la rotation, et sa difficulté à apprendre plusieurs vues d'un même objet. En particulier, il est difficile d'obtenir un classifieur capable de détecter à la fois des visages de face et de profil. Viola et Jones ont proposé une amélioration qui permet de corriger ce défaut[18], qui consiste à apprendre une cascade dédiée à chaque orientation ou vue, et à utiliser lors de la détection un arbre de décision pour sélectionner la bonne cascade à appliquer. Plusieurs autres améliorations ont été proposées par la suite pour apporter une solution à ce problème [19, 20].

2.2 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. 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.

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

2.2.1 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.

2.2.2 Normalisation Photométrique

Dans le domaine de la reconnaissance par le visage, un certain nombre de méthodes de normalisation de l'illumination ont été présentées. 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 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 (figure 2.5).



FIGURE 2.5 – Histogramme de l'image avant et après égalisation

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 :

$$p(x_i) = \frac{n_i}{n}, i \in 0, \dots, L \quad (2.5)$$

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 :

$$c(i) = \sum_{j=0}^i p(x_j) \quad (2.6)$$

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 :

$$y_i = T(x_i) = L \frac{c_i}{n} \quad (2.7)$$

avec L le niveau maximal de gris.

2.3 Reconnaissance de visages

De nombreuses méthodes de reconnaissance de visages ont été proposées au cours des trente dernières années. La reconnaissance faciale automatique est un challenge tel qu'il a suscité de nombreuses recherches dans des disciplines différentes : psychologie, neurologie, mathématiques, physique et informatique (reconnaissance des formes, réseaux de neurones, vision par ordinateur). C'est la raison pour laquelle la littérature sur la reconnaissance de visages est vaste et diversifiée.

Les neurologistes étudient depuis longtemps le mécanisme par lequel le cerveau reconnaît les visages, et beaucoup croient que le cerveau perçoit les visages d'une manière spécifique et très différente des autres objets visuels. Par exemple, les études ont constaté qu'une rotation d'image faciale de 180° dégrade la reconnaissance beaucoup plus qu'une même rotation pour des objets quelconques (voir figure 2.6). Dans un travail innovant [22], Moscovitch et al. ont démontré que le cerveau des hommes traite les visages et les objets dans des aires séparées à savoir que les visages sont traités dans une aire spéciale. Cette approche a fait office de référence pendant la dernière décennie jusqu'au travail récent de Jiang *et al.* [23] qui montre que le traitement des visages et des objets ne reposent finalement pas sur des mécanismes différents.



FIGURE 2.6 – Quand les visages ne sont pas vus dans leur état naturel, la capacité du système visuel humain à les distinguer est dégradée.

Les systèmes de reconnaissance de visages sont très souvent classés à partir des conclusions d'études psychologiques sur la façon dont les hommes utilisent les caractéristiques faciales pour reconnaître les autres. De ce point de vue, on distingue les trois catégories : les méthodes globales, les méthodes locales et les méthodes hybrides.

2.3.1 Méthodes globales

Le principe de ces méthodes est de représenter une image faciale par un seul vecteur de grande dimension en concaténant les niveaux de gris de tous les pixels du visage. Cette représentation, appelée description basée sur *l'apparence globale*, a deux avantages. Premièrement, elle conserve implicitement toutes les informations de texture et de forme utiles pour différencier des visages. Deuxièmement, elle peut tenir compte des aspects d'organisation structurelle globaux du visage. Toutefois, son inconvénient majeur réside dans la dimension très grande de l'espace image qu'elle nécessite ce qui rend très difficile la classification.

Pour traiter le problème des données de grande dimension, des techniques de réduction de la dimensionnalité peuvent être utilisées. L'une des techniques les plus courantes pour la reconnaissance de visages est la description par visages propres[3], qui est basée sur l'analyse en composantes principales (ACP).

Analyse en composantes principales (ACP)

Une méthode très populaire, basée sur la technique ACP, est la méthode *Eigenface*[3]. Son principe est le suivant : étant donné un ensemble d'images de visages exemples, il s'agit tout d'abord de trouver les composantes principales de ces visages. Ceci revient à déterminer les vecteurs propres de la matrice de covariance formée par l'ensemble des images exemples. Chaque visage exemple peut alors être décrit par une combinaison linéaire de ces vecteurs propres. Pour construire la matrice de covariance, chaque image de visage est transformée en vecteur. Chaque élément du vecteur correspond à l'intensité lumineuse d'un pixel. Cette méthode sera présentée avec davantage de détails dans le chapitre suivant.

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 optimisée pour la séparabilité (discrimination) de classe. Une alternative qui est l'analyse discriminante linéaire LDA tient compte de ceci.

Analyse Discriminante Linéaire (ADL)

Une autre méthode très connue est celle basée sur l'ADL (Analyse discriminante linéaire). L'objectif de la plupart des algorithmes basés sur l'ADL [26, 27], est de trouver les directions de projection les plus discriminantes dans l'espace propre, en maximisant le ratio entre les variations inter-personnelles et les variations intra-personnelles. Comme les variations intra-personnelles peuvent être petites (notamment quand il n'y a pas beaucoup d'images par individu), ce ratio est difficile à maximiser puisque il est déjà grand.

Ce problème est encore appelé *Small Sample Size*. Pour l'éviter, on peut utiliser tout d'abord l'ACP et ensuite l'ADL, et cette méthode est appelée *Fisherfaces*. Voilà pourquoi les méthodes basées sur l'ADL ne fonctionnent bien que lorsque beaucoup d'images par personne sont disponibles dans la base d'apprentissage. En revanche, quand il n'y a pas beaucoup d'images par personne, les méthodes basées sur l'ADL marchent moins bien que celles basées sur l'ACP [26].

Bien que les méthodes globales aient eu beaucoup de succès, leur inconvénient majeur réside dans le fait qu'elles utilisent uniquement des photos 2D d'apparence faciale. Or, on sait qu'une telle représentation est sensible aux changements d'expression, d'illumination et de poses. Une manière d'éviter ce problème consiste à utiliser des représentations faciales locales. En effet, les caractéristiques locales ne sont généralement pas aussi sensibles aux changements d'apparence que les caractéristiques globales.

2.3.2 Méthodes locales

Les méthodes locales peuvent être classées en deux catégories, les méthodes basées sur les points d'intérêt et celles basées sur l'apparence du visage.

Méthodes locales basées sur les points d'intérêts

On détecte tout d'abord les points d'intérêts et ensuite on extrait des caractéristiques localisées sur ces points d'intérêt. Les méthodes les plus anciennes en reconnaissance de visages appartiennent à cette catégorie [28, 29]. Elles s'appuient toutes [29, 30] sur l'extraction de caractéristiques géométriques spécifiques telles que la largeur de la tête, les distances entre les yeux,... Ces données sont ensuite utilisées par des classificateurs afin de reconnaître des individus. Ces méthodes présentent les deux inconvénients suivants :

1. les caractéristiques géométriques sont difficiles à extraire dans certains cas puisque la tâche de la détection précise de points caractéristiques n'est pas facile, en particulier dans les cas où des occultations ou des variations (pose, expression) de visages sont présentes.
2. les caractéristiques géométriques seules ne sont pas suffisantes pour représenter réellement un visage, alors que d'autres informations utiles telles que les valeurs des niveaux de gris de l'image sont complètement écartées.

Ces deux limites ont engendré deux directions de recherche. La première se concentre sur les performances des détecteurs de points caractéristiques du visage. Brunelli et Poggio [30] ont proposé d'utiliser un ensemble d'apprentissage pour détecter la position de l'oeil dans une image. Ils ont tout d'abord calculé pour chaque point des coefficients de corrélation entre l'image de test et les images de l'ensemble d'apprentissage et ensuite ils ont cherché les valeurs maximales. Rowley *et al.* [15] ont utilisé plusieurs détecteurs de traits spécifiques correspondant à chaque partie du visage, tels que les yeux, le nez, la bouche,... Malgré toutes ces recherches, il n'existe pas encore de détecteur de points caractéristiques qui soit suffisamment précis.

Dans la deuxième direction, les méthodes se concentrent sur des représentations plus élaborées des informations portées par les points caractéristiques du visage, plutôt que simplement sur des caractéristiques géométriques. Manjunath *et al.* [31] ont proposé des algorithmes pour détecter et représenter des caractéristiques faciales à partir d'ondelettes de Gabor⁵. Pour chaque point détecté, deux types d'information sont stockées : sa position et ses caractéristiques.

5. Le filtre de Gabor, défini par Dennis Gabor, est largement utilisé en traitement d'images car les ondelettes de Gabor présente deux propriétés intéressantes : la localisation fréquentielle et la sélection en

Pour modéliser la relation entre les points caractéristiques, un graphe topologique est construit pour chaque visage. Plus tard, Wiskott *et al.* [34] ont proposé une méthode très connue appelée Elastic Bunch Graph Matching (EBGM), où les nœuds des graphes sont situés sur un certain nombre de points sélectionnés du visage (voir la figure 2.7) De manière similaire à la méthode décrite dans l'étude de Manjunath *et al.* [31], Wiskott *et al.* ont utilisé les ondelettes de Gabor pour extraire les caractéristiques des points détectés car les filtres de Gabor sont robustes aux changements d'illumination, aux distorsions et aux variations d'échelle.

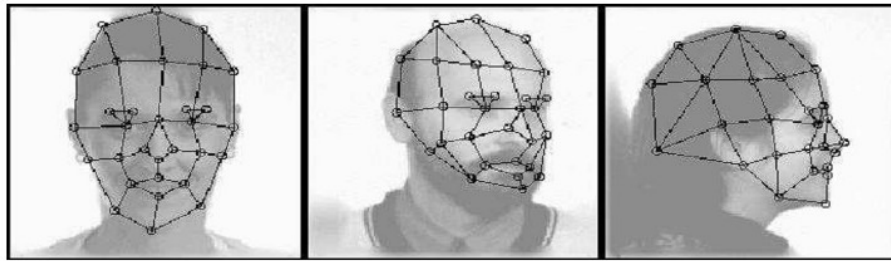


FIGURE 2.7 – EBGM.

Les méthodes locales basées sur l'apparence du visage

Dans les méthodes locales basées sur l'apparence du visage, on divise le visage en petites régions (ou patches) sur lesquelles les caractéristiques locales sont extraites directement. Martinez a présenté une approche probabiliste locale pour la reconnaissance de visages qui sont occultés partiellement et avec des variations d'expression dans le cas où une seule image de référence est disponible [32]. De nombreux échantillons virtuels sont d'abord générés à l'aide d'une méthode de perturbation de l'image, puis, chaque visage est divisé en six régions locales en forme d'ellipse. Ensuite, tous les patches locaux à la même position de chaque visage sont regroupés séparément dans un sous-espace de visages (donc six sous-espaces au total). Dans la phase d'identification, les images de test sont également divisées en six zones locales et sont projetées sur les espaces propres calculés ci-dessus, respectivement. Une approche probabiliste est utilisée pour mesurer la similarité d'une paire d'images. Des expériences sur un ensemble de 2600 images montrent que l'approche probabiliste locale ne réduit pas le taux de reconnaissance même lorsque $\frac{1}{6}$ du visage est occulté. Cependant, les complexités de calcul et de stockage ainsi que la procédure de génération des échantillons virtuels sont très compliquées (6615 échantillons par individu), en particulier si on a une base de référence avec de nombreux visages. La variation de pose est l'une des questions les plus importantes et difficiles en reconnaissance automatique de visages, en particulier dans le cas où une seule image de référence est disponible. Pour traiter ce problème, Kanade et Yamada [36] ont proposé une méthode probabiliste qui est similaire à celle de Moghaddam et Pentland [33].

orientation. Les représentations en fréquence et en orientation du filtre Gabor s'apparentent à celles du système visuel humain [35].

Variations	Caractéristiques locales	Caractéristiques globales
Petites variations	Pas sensible	Sensible
Grandes variations	Sensible	Très sensible
Illuminations	Pas sensible	Sensible
Expressions	Pas sensible	Sensible
Pose	Sensible	Très sensible
Bruit	Très sensible	Sensible
Occultations	Pas sensible	Très sensible

TABLE 2.1 – Comparaison des propriétés des caractéristiques locales et des caractéristiques globales.

Les méthodes mentionnées ci-dessus ne considèrent pas explicitement les relations existants entre les caractéristiques locales. Il est concevable que l'utilisation de cette information soit bénéfique pour le système de reconnaissance. Une solution possible est de construire un modèle flexible de géométrie sur les caractéristiques locales comme cela se fait dans la méthode EBGm.

2.3.3 Méthodes hybrides

Les méthodes hybrides (ou méthodes de fusion) sont des approches utilisant à la fois des caractéristiques globales et des caractéristiques locales. Les facteurs clés qui influent les performances des méthodes de fusion comprennent le choix des caractéristiques pour la combinaison et la manière de les combiner de telle sorte que leurs avantages soient préservés et que leurs inconvénients soient évités.

Les caractéristiques locales et les caractéristiques globales ont des propriétés très différentes et peuvent offrir des informations complémentaires utiles à la tâche de classification. Notons aussi que d'un certain point de vue, les méthodes locales peuvent être considérées comme des méthodes hybrides car des informations globales sont généralement prises en compte. Dans la méthode probabiliste locale [32] de nouveaux échantillons d'apprentissage sont d'abord produits pour chaque personne par méthode globale, puis une méthode locale est utilisée pour la reconnaissance.

2.4 Reconnaissance par Eigenfaces

La méthode de reconnaissance faciale Eigenfaces emploie la technique de l'analyse en composante principale, qui marque une différence notable avec les méthodes plus classiques, appelées méthodes locales, qui se basent sur les particularités du visage analysé, et dont les défauts résident dans son manque de précision, ainsi que sa sensibilité aux informations qui ne sont pas pertinentes. La méthode que nous avons utilisé est qualifiée de globale, puisque l'ensemble du visage est alors analysé. Notre technique de reconnaissance va donc utiliser la méthode d'analyse en composantes principales (également dite ACP). De manière simple, nous visons la diminution de la dimension de l'espace dans lequel nous

allons travailler, et nous pourrions alors simplifier les données à notre disposition et leur interprétation.

L'algorithme ACP, PCA en anglais (Principal Component Analysis) est né des travaux de M.A. Turk et A.P. Pentland au MIT Media Lab, en 1991 [3]. Il est aussi connu sous le nom de 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 reste sensible aux problèmes d'éclairage [45], de pose et d'expression faciale.

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 [46].

Nous allons chercher à trouver les visages propres ; tout d'abord, nous devons prendre un nombre M de visages d'apprentissage. Chacune de ces images, qui sont en pratique des matrices $N \times N$ sont alors transformées en un unique vecteur colonne de longueur N^2 .

Matrice $N \times N$ initiale :

$$\begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{1i} & \alpha_{1N} \\ \alpha_{21} & \alpha_{22} & \alpha_{2i} & \alpha_{2N} \\ \dots & \dots & \dots & \dots \\ \alpha_{N1} & \alpha_{N2} & \alpha_{Ni} & \alpha_{NN} \end{pmatrix}$$

transformée en :

$$\begin{pmatrix} \alpha_{11} \\ \vdots \\ \alpha_{N1} \\ \vdots \\ \alpha_{1N} \\ \vdots \\ \alpha_{NN} \end{pmatrix}$$

Nous devons par la suite déterminer le visage moyen, déduit des M visages d'apprentissages

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (2.8)$$



FIGURE 2.8 – visage moyen

Ce visage moyen va servir dans l'analyse d'images, on soustrait en effet ce visage moyen aux visages d'apprentissages, ce qui nous laisse alors les informations propres à ce visage, nous récupérons alors dans Φ_i uniquement les informations qui sont particulières à ce visage d'apprentissage.

$$\Phi_i = \Gamma_i - \Psi \quad (2.9)$$

Où Φ_i représente le i^{eme} visage auquel on a soustrait le visage moyen. A présent, nous devons calculer la matrice de covariance D . Elle correspond à

$$D = QQ^T \quad (2.10)$$

avec

$$Q = [\Phi_1, \Phi_2, \dots, \Phi_M] \quad (2.11)$$

Nous devrions calculer les vecteurs propres d_i de la matrice D . Mais cela représente pour nous N^2 vecteurs propres de dimension N^2 chacun.

C'est à présent que nous allons réduire l'information en limitant les composantes avec lesquelles nous travaillerons, en accord avec le principe de l'analyse en composantes principales. Nous allons donc considérer la matrice $E = Q^T Q$, dont nous trouverons les vecteurs propres e_i . Cette matrice est de taille $M \times M$ ce qui nous simplifiera donc les choses étant donné que nous aurons M vecteurs propres de taille M chacun. Le passage de la matrice D à la matrice E n'est pas anodin, nous utilisons en effet le fait que les vecteurs propres de ces deux matrices sont liés de manière assez proche. En effet, nous

avons comme relation,

$$Ee_i = Q^T Qe_i = \lambda_i e_i \quad (2.12)$$

avec λ_i la valeur propre associée au vecteur propre e_i .

En multipliant cette équation par la matrice Q , il vient

$$QEe_i = QQ^T Qe_i \quad (2.13)$$

Nous voyons alors apparaître la matrice D

$$QEe_i = DQe_i = \lambda_i Qe_i \quad (2.14)$$

Nous en déduisons donc qu'avec e_i vecteur propre de la matrice E associé à la valeur propre λ_i , nous avons par conséquent Qe_i est un vecteur propre de la matrice D associé à la même valeur propre λ_i . Ainsi, nous avons d_i vecteur propre de D , avec

$$d_i = Qe_i \quad (2.15)$$

Ce sont les valeurs propres qui leur sont associées qui nous permettent ensuite de classer les vecteurs propres en fonction de leur capacité à caractériser les variations entre les images. Lorsque l'on les visualise (ces vecteurs sont à l'origine des matrices de taille $N \times N$), les faces propres sont ce que l'on pourrait appeler des images aux airs fantomatiques. Mais gardons à l'esprit que ce sont les vecteurs propres de la matrice de covariance des images d'apprentissage des visages. (Figure 2.9)

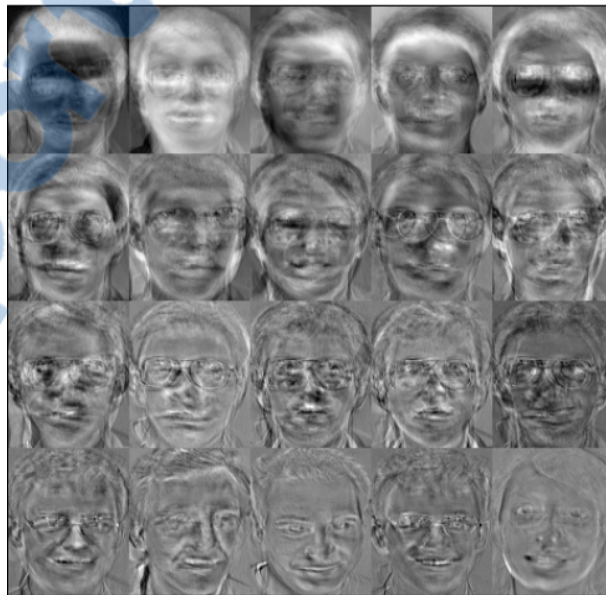


FIGURE 2.9 – Quelques faces propres

Les M vecteurs propres que nous avons alors obtenus nous permettrons donc d'approximer au mieux les visages d'apprentissage en utilisant les visages propres de plus grande importance. L'avantage de réduire le nombre de visages propres est d'une part de

nécessiter de moins d'espace mémoire, mais aussi de réduire les calculs, leur temps d'exécution ; cependant nous perdons sans aucun doute de l'information et donc l'information moins précise, mais les résultats ne s'en verront pas vraiment modifiés, étant donné que nous ne nous donnons qu'une mission d'identification. Nous ne cherchons pas à reconstruire le visage du sujet à partir de nos visages propres, mais seulement à le reconnaître. Parmi les M vecteurs propres trouvés, nous allons seulement conserver un nombre L , qui seront les plus significatifs. Nous allons trouver maintenant le poids associé à chacun des visages propres. Les images servant à l'apprentissage, auquel on a enlevé l'image moyenne, sont en fait combinaison linéaire des visages propres.

$$\Phi_i = \sum_{i=1}^L P_i d_i \quad (2.16)$$

Pour trouver le poids associé, nous faisons pour chacune des coordonnées correspondant à un visage d'apprentissage

$$P_i = d_i^T \Phi_i \quad (2.17)$$

Ce qui nous permet d'obtenir pour chacun des M visages d'apprentissages un vecteur π_i , où i représente le i^{eme} visage, et qui nous informe sur le coefficient appliqué à chacun des visages propres.

$$(2.18)$$

$$\Pi_i = \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_L \end{pmatrix}$$

Passons à présent au travail à effectuer pour la reconnaissance d'un visage d'un sujet. Une fois l'image prise, l'image (vecteur colonne Γ) obtenue est soustraite à l'image moyenne Ψ :

$$\Phi = \Gamma - \Psi \quad (2.19)$$

Puis nous trouvons les coordonnées de cette image dans l'espace réduit des faces propres

$$P_i = d_i^T \Phi_i \quad (2.20)$$

Ce qui nous donne au final :

$$\Pi_i = \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_L \end{pmatrix}$$

Il nous faut maintenant interpréter la projection de l'image à analyser pour identifier le sujet. Pour cela nous allons utiliser une mesure de distance particulière, la distance de

Mahalanobis. L'intérêt de cette distance réside dans le fait qu'elle va accorder un poids moins important aux composantes bruitées, et qu'elle permet de séparer efficacement les axes pour lesquels l'information peut être mieux classifiée.

Elle est définie par :

$$d(a, b) = \sqrt{(a - b)^T Q^{-1} (a - b)} \quad (2.21)$$

avec Q covariance des variables.

Nous cherchons donc :

$$m = \min ||\Pi - \Pi_i|| \quad (2.22)$$

Puis, nous comparons la valeur de m trouvée à une valeur seuil Δ , qui aurait du être déterminée à partir d'essai sur des images choisies aléatoirement, puis en comparant ces valeurs aux valeurs obtenues avec des visages d'apprentissage, et déterminer si oui ou non l'image analysée correspond à un visage présent dans la base de données. Le choix de ce seuil dépend de trop nombreuses conditions (prise de vue des images, niveau de précision souhaité pour la reconnaissance, etc..

2.5 Bases de données

Plusieurs bases de données contenant des informations qui permettent l'évaluation des systèmes de reconnaissance de visages sont disponibles sur le marché. Toutefois, ces bases de données sont généralement adaptées aux besoins de quelques algorithmes spécifiques de reconnaissance, chacune d'elle a été construite avec des conditions d'acquisition d'images de visages diverse (changements d'illumination, de pose, d'expressions faciales) ainsi que le nombre de sessions pour chaque individu.

Les bases les plus anciennes (ORL et YALE) ont été le plus utilisées et permettent de comparer plus facilement de nouvelles méthodes à celles de l'état de l'art. Les plus récentes (Color FERET, FRGC, CVL, AR et IV²) contiennent plus de personnes et sont donc utiles pour des évaluations à plus grande échelle. D'autres bases de visages sont disponibles et destinées à des évaluations adaptées à certaines variabilités du visage telles que les bases UMIST, BANCA, PF01, Yale et PIE. Ces trois dernières bases par exemple (PF01, Yale et PIE) disposent d'un nombre important de poses différentes mais renferment seulement quelques dizaines de personnes acquises lors d'une seule session.

Notre choix s'est arrêté sur les bases XM2VTS, Color FERET, AR, CVL en plus de la base ORL pour développer notre corpus de travail. Nous détaillons principalement ces bases d'images ci-dessous. Une liste plus complète et plus détaillée est disponible dans[37].

2.5.1 La base XM2VTS

La base XM2VTS [38] (The Extended M2VTS Database) est une base de données multimodale qui renferme des images faciales 2D et 3D, ainsi que des séquences vidéo avec des enregistrements de la voix. La base de données a été acquise dans le cadre du projet M2VTS (Multi Modal Verification for Teleservices and Security applications) à l'Université de Surrey (Royaume-Uni). La base de données contient 295 personnes enregistrées

en 4 sessions durant une période de 4 mois. La base de données de visage 2D contient des prises de vue faciales et de profil. Deux prises de vue faciales ont été enregistrées, à chaque session, ce qui fait un total de 8 images pour chaque personne. La figure 2.10 illustre un exemple d'images d'une personne. Les images en colonnes représentent les sessions auxquelles elles appartiennent.

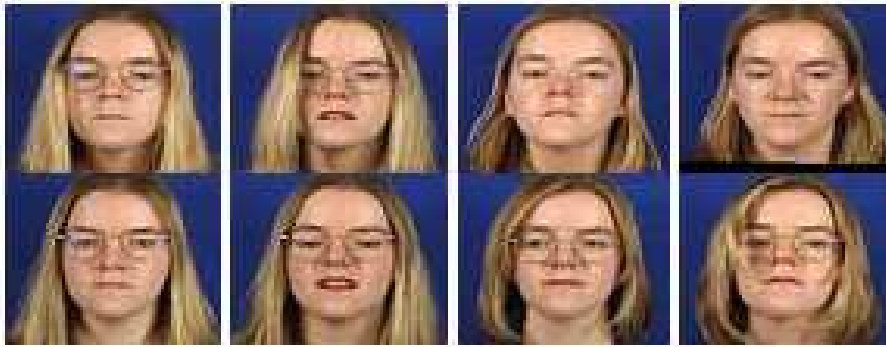


FIGURE 2.10 – Exemples d'images de la base XM2VTS.

2.5.2 La base FERET

La base de données FERET a été collectée dans le cadre du programme *Facial Recognition Technology* [39, 40] mené par le National Institute of Standards and Technology (NIST) Américain. Il s'agit de la plus grande base disponible pour les chercheurs qui a été acquise avec des poses différentes et durant 15 sessions entre 1993 et 1996. Les images, initialement collectées depuis un appareil photographique de 35mm ont ensuite été digitalisées. Une première version de cette base de données a été réalisée en 2001 et contient 14051 images faciales en niveaux de gris avec une résolution de 256 x 384 pixels.

La version la plus récente, réalisée en 2003, renferme des images numériques couleurs de plus grande qualité avec une résolution de 512 x 768 pixels et une compression sans perte de données à la différence des premières images en niveaux de gris. En plus, de multiples erreurs de noms d'images, d'identifiants, et dates de capture, qui figurent sur la première base à niveau de gris ont été corrigées. Cette dernière base contient 11338 images représentant 994 personnes différentes. Pour chaque individu, on dispose d'une vue faciale régulière **fa** et une vue faciale alternative **fb** prise un peu après **fa**. D'autres poses ont été acquises pour la majorité de ces individus allant du profil gauche au profil droit avec des rotations de 15°, 22°, 45°, 67° et 90° en profondeur de la tête. Pour quelques personnes de la base, on dispose d'autres vues *duplicate* collectées dans des conditions similaires à **fa** et **fb** mais dans des sessions ultérieures. Aucune contrainte n'est imposée sur la date de la prise de vue de l'image *duplicate* **I**. par contre, la vue *duplicate* **II** a été collectée au moins 540 jours après la première prise de vue.

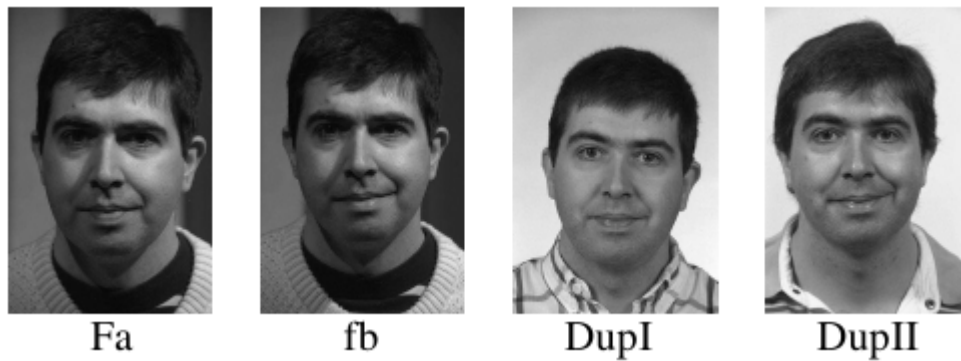


FIGURE 2.11 – Extrait de la base Color FERET : Les images sont transformées en niveau de gris.

2.5.3 La base AR

La base AR[41] a été constituée en 1998 au sein du laboratoire Computer Vision Center (CVC) à Barcelone, en Espagne. 116 personnes (63 hommes et 53 femmes) sont enregistrées. Les images sont en couleur de taille 768 x 576 pixels. 13 vues de chaque sujet ont été collectées. Pour la majorité de ces personnes, 13 autres vues ont été acquises lors d'une deuxième session à deux semaines d'intervalle.

Un extrait des images collectées lors de la première session est donné en figure 2.12. Ces vues renferment des changements d'expression faciale, d'éclairage, ainsi que des occultations partielles des yeux (lunettes de soleil) et de la partie basse du visage (cache-col). Lors de la seconde session, les 13 vues sont collectées dans les mêmes conditions que pour la première.

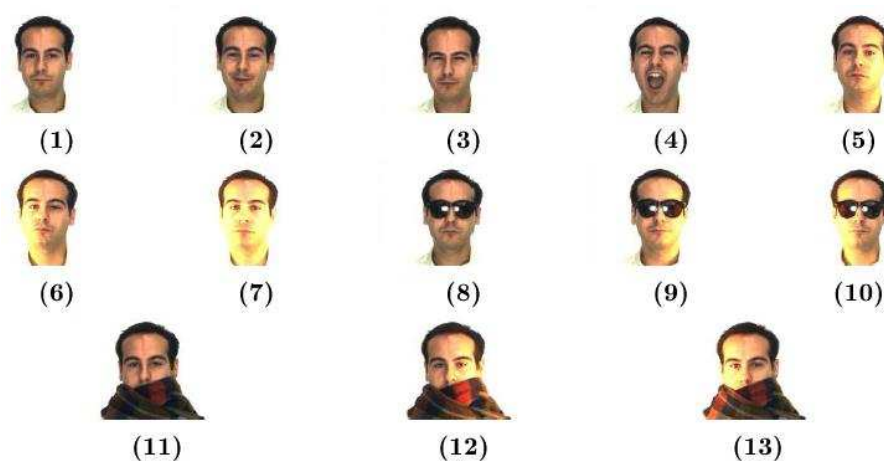


FIGURE 2.12 – Extrait de la base AR. Ensembles des vues collectées pour un individu de la base.

2.5.4 La base CVL

Cette base de données de visage a été créée au Computer Vision Laboratory (CVL) à l'université de Ljubljana (Slovénie) [42]. 114 personnes y sont enregistrées. Les images sont en couleur de taille 640 x 480 pixels. 7 vues de chaque sujet ont été collectées. Ces vues renferment différentes poses et expressions prises dans des conditions de luminosité uniformes. Parmi ces images, trois seulement sont de face. La figure 2.13 illustre un exemple d'images de face de la base CVL.



FIGURE 2.13 – Images faciales extraites de la base CVL.

2.5.5 La Base de donnée ORL

Conçu par AT&T laboratoires de l'université de Cambridge en Angleterre, la base de donnée ORL (Olivetti Research Laboratory) est une base de donnée de référence pour les systèmes de reconnaissances automatique des visages. En effet tous les systèmes de reconnaissances de visages trouvés dans la littérature ont été testés par rapport à l'ORL, cette popularité est due au nombre de contraintes imposées par cette base car la plus part des changements possibles et prévisibles du visage ont été pris en compte, comme par exemple : le changement de coiffure, la barbe, les lunettes, les changements dans les expressions faciales, *etc.* Ainsi que les conditions d'acquisition telles que : le changement d'illumination et le changement d'échelle dû à la distance entre le dispositif d'acquisition et l'individu.

La base de donnée ORL est constituée de 40 individus, chaque individu possède 10 poses, donc la base contient 400 images. Les poses ont été prises sur des intervalles de temps différents pouvant aller jusqu'à trois mois. L'extraction des visages à partir des images a été faite manuellement. Nous présenterons dans ce qui suit les figures montrant les spécificités de la base de donnée de référence ORL.



FIGURE 2.14 – Base de donnée ORL

Voici un exemple où l'acquisition se fait sous différentes orientations du visage et sous différents éclairages :



FIGURE 2.15 – Exemple de changements d'orientations du visage

Voici un exemple où l'acquisition se fait sous différents éclairages :



FIGURE 2.16 – Exemple de changements d'éclairage

Cet exemple montre les changements d'échelle dus à la distance entre le dispositif d'acquisition et l'individu :

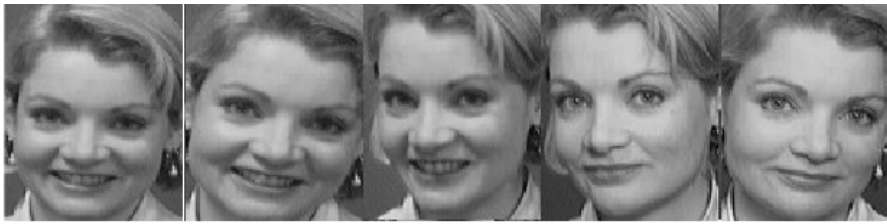


FIGURE 2.17 – Exemple de changements d'échelle

La base de donnée ORL prend aussi en considération les expressions faciales, telles que les grimaces. En voici un exemple :

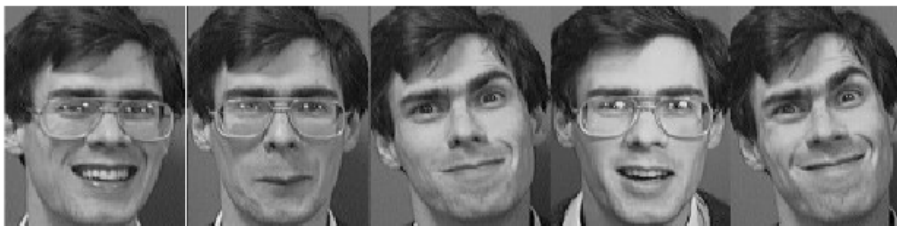


FIGURE 2.18 – Exemple de changements des expressions faciales

La Base ORL prend en compte le fait qu'un individu peut porter ou ne pas porter des lunettes. Cet exemple en est l'illustration :



FIGURE 2.19 – Exemple de port de lunettes

Un individu peut aussi porter une barbe ou changer de coiffure, la base ORL prend en considérations ces particularités :

La base ORL comprend aussi des individus de différents ages, sexe et couleurs de peaux :



FIGURE 2.20 – Exemple de changements de coiffure et de port de barbe



FIGURE 2.21 – Exemple d'individus de différents âges races et sexes

2.5.6 Base Labeled Faces in the Wild (LFW)

Très récemment, une nouvelle base de visages intéressante nommée *Labeled Faces in the Wild*⁶ a été présentée. Cette base de données qui contient 13233 images de 5749 personnes est considérée naturelle car les images ont été collectées directement sur le site d'information *Yahoo! News* en utilisant un détecteur automatique de visage. Cette base est plus difficile pour les chercheurs car aucune contrainte sur les paramètres de prise de vue n'a été imposée (la base contient donc des variations importantes de pose, d'âge, d'expression, des images de mauvaise qualité, etc.). Dans le protocole d'évaluation associé à cette base, la question de la reconnaissance de visages devient une tâche de comparaison de l'identité de deux images dans une paire : étant donnée une paire d'images faciales, il s'agit de déterminer si elles sont issues de la même personne ou pas. Grâce aux étiquettes associées aux images originales, une paire d'images de la base LFW est soit classée dans la catégorie *Same*, si les deux images proviennent d'une même personne, soit classée dans la catégorie *Diff*, si les deux images proviennent de deux personnes différentes. La figure 2.22 montre quelques exemples de paires d'images de cette base. Cette base est disponible sous trois versions : *original* qui contient les images détectées sans alignement, *funneled* qui contient les images alignées automatiquement par l'algorithme de Huang et al. [43], *aligned* qui contient les images alignées automatiquement par l'algorithme de Wolf *et al.* [44].



FIGURE 2.22 – Paires de la base LFW aligned

6. <http://vis-www.cs.umass.edu/lfw/>

3

Expérimentation

Ce dernier chapitre est consacré à la conception de l'application qui permettra d'identifier des personnes par reconnaissance faciale. Le programme de reconnaissance faciale sera codé en C++. Il utilisera les bibliothèques OpenCV pour pouvoir prendre des photos à l'aide de la webcam d'un ordinateur.

Son rôle sera d'identifier, parmi une base de visages connus, lequel se rapproche le plus d'un visage passé en paramètre. Il devra également indiquer à quel point il s'en rapproche, afin de pouvoir valider ou infirmer l'identification. Il utilisera la technique dite *Eigenfaces*, l'Analyse en Composante Principale, qui bien que n'offrant pas les meilleures performances à l'heure actuelle, est relativement simple à mettre en œuvre et reste efficace. Elle affichera dans une fenêtre le visage reconnu ainsi que diverses informations comme le nom de la personne et le score de ressemblance, ce dernier est comparé à un seuil déjà fixé pour vérifier l'existence de la personne recherchée dans la base de données. 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*).

3.1 Environnement du travail

Afin de mener à bien ce projet, nous avons utilisé un ensemble de matériels dont les principales caractéristiques sont les suivantes :

- Processeur : Intel(R) CoreTM i3-M370 @2.4 GHZ
- RAM : 4.00 GO
- OS : Linux Ubuntu 12.04 LTS
- Compilateur : Gnu gcc, Gnu Make

3.2 OpenCV

OpenCV (Open Source Computer vision) est une bibliothèque graphique libre, initialement développée par Intel et maintenant soutenu par la société de robotique Willow

Garage, spécialisée dans le traitement d'images en temps réel [47]. Elle est livrée avec une interface de programmation en C, C++, Python et Android.

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées, elle propose la plupart des opérations classiques en traitement d'images :

- lecture, écriture et affichage d'image ;
- calcul de l'histogramme des niveaux de gris ou d'histogramme de couleur ;
- lissage, filtrage.

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils en traitement vidéo :

- lecture, écriture et affichage d'une vidéo depuis un fichier ou une caméra ;
- détection de visage par la méthode de Viola et Jones, détection de mouvements, poursuite d'objets ;

3.3 Implémentation

3.3.1 Comment détecter un visage à l'aide du détecteur OpenCV

Comme il a déjà été mentionné, la première étape dans la reconnaissance des visages est la détection des visages. Avec la bibliothèque OpenCV, il est assez facile de détecter un visage de face dans une image en utilisant son détecteur de visage Haar Cascade (connu comme la méthode de Viola-Jones).

Etant donné un fichier ou une vidéo en direct, le détecteur de visage examine chaque emplacement de l'image et classe comme *visage* ou *non visage*. Le classifieur utilise des données stockées dans un fichier XML pour décider comment classifier chaque localisation image. OpenCV est livré avec plusieurs classifieur différents pour la détection de visage dans des poses frontales, ainsi que la détection de certains visages de profil, la détection des yeux, détection des corps...*etc.*

Nous pouvons utiliser la fonction `cvHaarDetectObjects` avec l'un de ces autres détecteur, mais le détecteur des visages de face est le seul qui est très fiable.

Pour la détection des données XML, nous pouvons choisir l'un de ces classifieurs Haar Cascade d'OpenCV (dans le répertoire "opencv/data/haarcascade") :

- `haarcascade_frontalface_default.xml`
- `haarcascade_frontalface_alt.xml`
- `haarcascade_frontalface_alt2.xml`
- `haarcascade_frontalface_alt_tree.xml`



```

emacs23@bouchra-Vostro-1540
File Edit Options Buffers Tools XML Help
<?xml version="1.0"?>
<opencv_storage>
<nPersons>10</nPersons>
<personName_1>person</personName_1>
<personName_2>person</personName_2>
<personName_3>person</personName_3>
<personName_4>person</personName_4>
<personName_5>person</personName_5>
<personName_6>person</personName_6>
<personName_7>person</personName_7>
<personName_8>person</personName_8>
<personName_9>person</personName_9>
<personName_10>person</personName_10>
<nEigens>29</nEigens>
<nTrainFaces>30</nTrainFaces>
<trainPersonNumMat type_id="opencv-matrix">
<rows>1</rows>
<cols>30</cols>
<dt>i</dt>
<data>
1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 6 6 6 7 7 7 8 8 8 9 9 9 10 10</data></trainPersonNumMat>
<eigenValMat type_id="opencv-matrix">
<rows>1</rows>
<cols>29</cols>
<dt>f</dt>
<data>
1.93736002e-01 1.60661444e-01 1.19548447e-01 9.83417556e-02

```

FIGURE 3.1 – Base de données au format XML

Nous avons choisi le fichier `haarcascade_frontalface_alt.xml` pour l'application car il correspond parfaitement à notre objectif.

Pour charger des données XML, nous utilisons la fonction `cvLoad()` avec le chemin vers le fichier XML contenant cascade de Haar comme son premier paramètre d'entrée :

```
faceCascade=(CvHaarClassifierCascade*)
cvLoad(haarcascade\_frontalface\_alt.xml, 0, 0, 0);
```

On ouvre le flux caméra avec `cvQueryframe`

```
frame = cvQueryFrame(camera)
```

Il ne reste plus qu'à effectuer une détection de visage sur l'image capturée, en utilisant haar Cascade et retourner un rectangle pour la région détectée dans l'image donnée.

```
rects = cvHaarDetectObjects(detectImg, cascade, storage,
search\_scale\_factor, 3, flags, minFeatureSize);
```

La Fonction utilise en paramètre l'image récupérer pendant la boucle de traitement. C'est sur cette image que toutes les opérations vont être effectuées par la suite.

Dans OpenCV, la détection de visage par la méthode de Viola et Jones est déjà implémentée dans la fonction `cvHaarDetectObjects`. Le résultat de cette fonction est une série d'objets d'un même type qui ont passé les critères de sélection définis par le classifieur. Dans notre cas ce sera nos différents visages détectés.

Nous avons maintenant l'ensemble des visages qui ont été détectés, et afin d'afficher le résultat à l'écran nous allons simplement dessiner un carré autour du visage détecté.

3.3.2 Comment faire le prétraitement des images faciales pour la reconnaissance faciale :

Maintenant que nous avons détecté un visage, nous pouvons utiliser cette image de visage pour la reconnaissance faciale. Toutefois, si nous avons essayé de simplement effectuer une reconnaissance de visage directement sur une image normale de photo, nous aurons une précision inférieure à 10%.

Il est extrêmement important d'appliquer diverses techniques de prétraitement d'image pour standardiser les images que nous fournissons à un système de reconnaissance faciale. La plupart des algorithmes de reconnaissance de visage sont extrêmement sensibles aux conditions d'éclairage de sorte que s'il a été formé pour reconnaître une personne dans une pièce sombre, il ne sera probablement pas la reconnaître dans une salle lumineuse.

Il y a aussi d'autres problèmes, tels que le visage doit être dans une position très cohérente dans les images (tels que les yeux se trouvant dans les mêmes coordonnées de pixels), de taille uniforme, angle de rotation, émotion (sourire, colère, *etc*). C'est pourquoi il est si important d'utiliser un prétraitement d'image avant d'appliquer la reconnaissance faciale.

Notre système de reconnaissance est basé sur la méthode Eigenfaces qui utilise des images en niveaux de gris. Le code suivant va convertir une image d'entrée RVB (en couleur) en une image en niveaux de gris, redimensionner l'image à une dimension cohérente, puis appliquer l'égalisation d'histogramme pour standardiser automatiquement la luminosité et le contraste des images.

3.3.3 Comment Eigenfaces peut être utilisée pour la reconnaissance des visages :

Maintenant que nous avons une image faciale prétraitée, nous pouvons effectuer la méthode PCA pour la reconnaissance faciale. OpenCV est livrée avec la fonction `cvEigenDecomposite()`, qui effectue l'opération PCA, mais nous avons besoin d'une base de données (ensemble d'apprentissage) des images.

Donc, nous devons rassembler un groupe d'images faciales prétraitées de chaque personne que nous voulons reconnaître. Par exemple, si vous voulez reconnaître quelqu'un d'une classe de 10 étudiants, alors vous pouvez stocker 20 photos de chaque personne, pour un total de 200 images faciales prétraitées de la même taille. On utilisant l'ACP pour convertir les 200 images a un ensemble de *visages propres* : *Eigenfaces* qui représentent les principales différences entre les images d'apprentissage.

D'abord, il va trouver *l'image moyenne* de nos images on obtenant la valeur moyenne de chaque pixel. Ensuite, les visages propres sont calculées par rapport à cette image moyenne.

3.3.4 Implémentation du système de reconnaissance faciale Hors ligne (en ligne de commande)

Dans la mise en œuvre d'un système de reconnaissance faciale hors ligne, des fichiers sont utilisés comme entrée et la sortie se fait par ligne de commandes. Fonda-

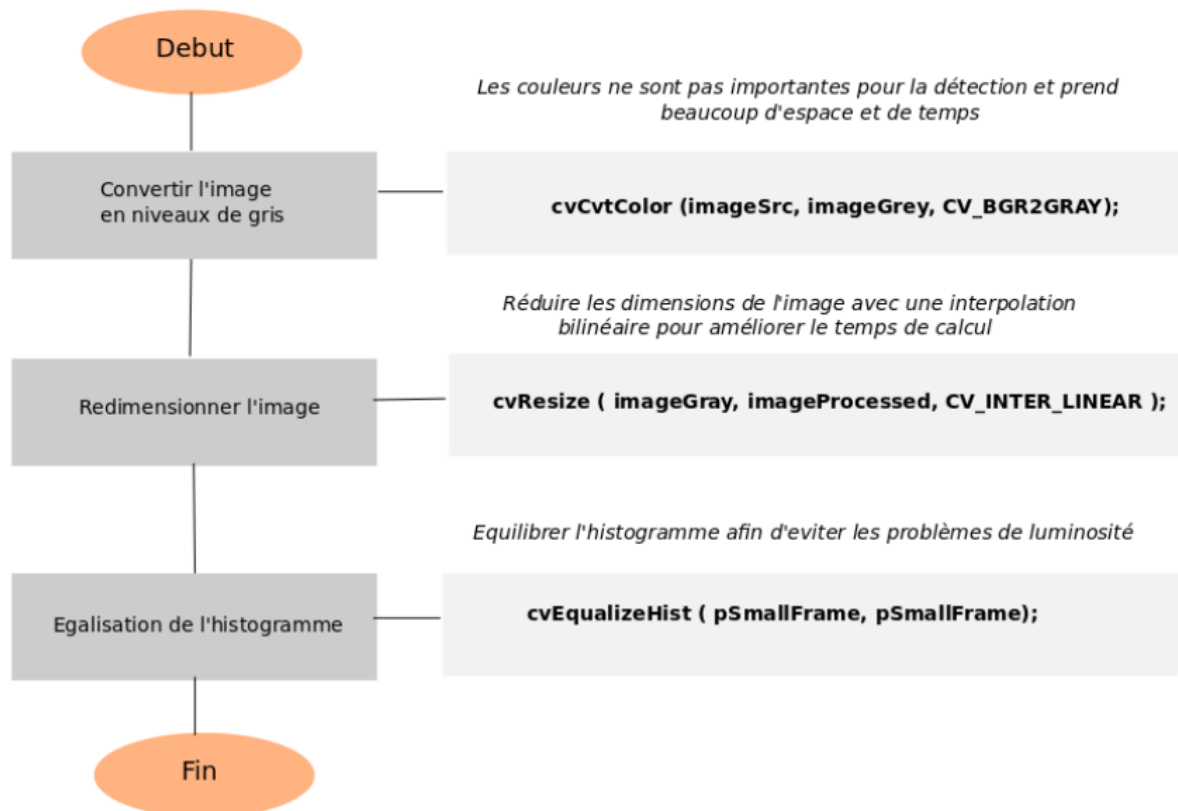


FIGURE 3.2 – Organigramme du prétraitement

mentalement pour créer une base de données à partir d’images d’apprentissages, on va créer un fichier texte qui liste les fichiers d’images et quelle personne de chaque fichier image représente. Par exemple on pourra mettre cela dans un fichier texte appelé : `3_images_pour_2_personnes.txt`.

Le programme peut alors charger tous dans un ensemble d’images en utilisant la fonction `loadFaceImgArray()`. Pour créer la base de données à partir de ces images chargées, nous utilisons la fonction d’OpenCV `cvCalcEigenObject()` qui calcule l’image moyenne, les vecteurs propres (visages propres) et les valeurs propres (ratios) et la fonction `cvEigenDecomposite()` pour projeter chaque image d’essai dans le sous espace de ACP.

Nous avons maintenant l’image moyenne `pAvgTrainImg`, les images des visages propres `eigenVectArr[]` et la matrice de valeurs propres `projectedTrainFaceMat` de chaque image d’apprentissage. Ceux-ci peuvent être stockés dans un fichier, qui sera la base de données de reconnaissance faciale. La fonction `storeTrainingData()` dans le code va stocker ces données dans le fichier `facedata.xml` qui peut être rechargé à tout moment pour reconnaître les gens. Il ya aussi une fonction `storeEigenfaceImages()` qui stocke l’image moyenne dans `out_averageImage.bmp` et les visages propres dans `out_eigenfaces.bmp`.

Le fichier texte répertoriant les images qui doivent être testées, ainsi que la bonne personne dans cette image. Le programme peut alors essayer de reconnaître qui est dans chaque photo, et vérifier la valeur correcte dans le fichier d’entrée pour voir si elle était correcte ou pas.



FIGURE 3.3 – Exemple d’image moyenne, le premier visage propre et le dernier visages propres d’une collection de 30 images de 4 personnes

```
1 personA data/s1/1.pgm
1 personA data/s1/2.pgm
1 personA data/s1/3.pgm
2 personB data/s2/1.pgm
2 personB data/s2/2.pgm
2 personB data/s2/3.pgm
```

FIGURE 3.4 – Cela dit au programme que la personne 1 est nommé `personA`, et les 3 photos du visage prétraitées de `personA` sont dans le dossier `data/s1`, et personne 2 est appelé `personB` avec 3 images dans le dossier `data/s2`

La liste des fichiers d’images (visages prétraitées) et les noms sont chargés dans un tableau d’images, à partir du fichier texte qui est maintenant utilisé pour les tests de reconnaissance (au lieu de l’apprentissage). Cette opération est effectuée dans le code en `loadFaceImgArray ()`.

Le visage moyen, eigenfaces et les valeurs propres (ratios) sont chargés à partir du fichier `facedata.xml` de la base de données de la reconnaissance des visages, par la fonction `loadTrainingData ()`.

Chaque image d’entrée est projeté sur le sous-espace ACP pour voir quel est le meilleur rapport de visages propres pour représenter cette image d’entrée. Mais, maintenant, qu’il a les valeurs propres (ratios d’images Eigenface) pour représenter l’image d’entrée, il cherche l’image initiale qui avait les ratios les plus semblables. Ceci est fait dans le `findNearestNeighbor ()` en utilisant la *distance euclidienne*. Fondamentalement, il vérifie le degré de similarité de l’image d’entrée avec chaque image de l’apprentissage et trouve la plus proche.

La distance entre l’image d’entrée et l’image de l’apprentissage la plus proche est utilisée pour déterminer la valeur de *confiance*, pour être utilisée comme un guide, si une personne a été effectivement reconnue ou non. Une confiance de 1.0 signifierait une bonne correspondance, et une confiance de 0.0 ou négative signifierait une mauvaise correspon-

dance.

Une fois que le programme sait quelle image de l'apprentissage est très semblable à l'image d'entrée, et en supposant que la valeur de confiance n'est pas trop faible (elle devrait être supérieure ou égale à 0.6), il a alors reconnu la personne.

3.3.5 Comment utiliser le système FaceRec en ligne de commande :

Pour utiliser le mode ligne de commande, voici les instructions :

1. Tout d'abord, on a besoin des images de visage. Dans mon travail, j'ai utilisé la base de données de visages "ORL/AT & T".
2. Indiquer les visage de test qu'on voudrez utiliser dans des fichiers texte. Si on a téléchargé la base ORL, on peut utiliser ces exemples de fichiers de texte : `lower2.txt` et `upper6.txt`.
3. Pour lancer la phase d'apprentissage de Eigenface, entrer dans l'invite de commande :

```
>OnlineFaceRec train lower2.txt
```

Cela va créer un fichier base de données `facedata.xml` avec seulement 2 faces chacune de 10 personnes (un total de 20 faces). Il générera également `out_averageImage.bmp` et `out_eigenfaces.bmp`.

```
>OnlineFaceRec test upper6.txt
```

Ce qui permettra de tester la base de données avec 6 faces chacun de 10 personnes (un total de 60 visages). Elle devrait donner un taux de reconnaissance étonnamment élevé de 95% de réponses correctes, à partir de seulement 2 photos de chaque personne.

Il est possible d'essayer d'autres combinaisons pour obtenir une précision encore plus élevée. N'oublions pas que ces photos sont déjà traitées et qu'elles ont été prises dans des conditions de laboratoire précises. Si on prend des images dans les conditions du monde réel, on devra nous attendre à seulement 40 à 80% de réponses correctes.

3.3.6 La mise en œuvre de reconnaissance faciale en temps réel à partir d'une caméra :

Il est facile d'utiliser un flux de webcam comme entrée pour le système de reconnaissance de visage au lieu d'une liste de fichiers. Fondamentalement, il suffit de saisir les images d'une caméra au lieu de puiser dans un fichier, puis de maintenir l'exécution jusqu'au départ de l'utilisateur, au lieu de simplement exécuter jusqu'à ce que la liste des fichiers est terminée. OpenCV offre la fonction `cvCreateCameraCapture ()` (aussi connu comme `cvCaptureFromCAM ()`).

Maintenant, on a une façon de reconnaître les gens en temps réel à l'aide d'une caméra, mais pour qu'il saisisse de nouveaux visages, il faut arrêter le programme, enregistrez les

images de la caméra en tant que fichiers images, mettre à jour la liste des images d'apprentissage, utiliser la méthode `a` à partir de ligne de commande, puis exécutez à nouveau le programme en mode caméra en temps réel.

Voici donc le moyen le plus simple d'ajouter une nouvelle personne à la base de données de reconnaissance faciale à partir du flux de caméra sans arrêter le programme :

1. Recueillir un tas de photos de l'appareil photo (images faciales prétraitées), peut-être pendant que vous effectuez la reconnaissance de visage aussi.
2. Enregistrez les images de visage collectées sous forme de fichiers d'image sur le disque dur en utilisant `cvSaveImage()`.
3. Ajoutez le nom du fichier de chaque image de visage à la queue du fichier de la liste des images d'apprentissage.
4. Une fois qu'on est prêt des nouvelles images (tels que une fois que vous avez 20 visages, ou lorsque l'utilisateur dit qu'ils sont prêts), vous *recycler* la base de données de tous les fichiers images. Le fichier texte listant les fichiers d'image d'apprentissage contient de nouvelles images, et les images sont stockées sous forme de fichiers d'image sur l'ordinateur.
5. Mais avant de faire le recyclage, il est important de libérer les ressources qui ont été utilisées, et ré-initialiser les variables, de telle sorte qu'il se comporte comme si vous arrêtez le programme et redémarrer. Par exemple, que les images sont stockées sous forme de fichiers et ajoutées au fichier texte de la liste de la formation, on doit libérer les tableaux de visages propres, avant de faire l'équivalent de la formation hors ligne (qui implique le chargement de toutes les images à partir du fichier de liste de la formation, puis de trouver les visages propres et les rapports de la nouvelle base d'apprentissage en utilisant ACP).

3.3.7 Comment utiliser le système FaceRec de webcam en temps réel :

Si vous possédez une webcam branchée, alors vous devriez être en mesure de tester ce programme en compilant le code et l'exécuter. Appuyez sur la touche ESC de la fenêtre GUI lorsque vous souhaitez quitter le programme.

1. Dans le terminal, appuyez sur la touche `n` de votre clavier quand une personne est prête pour l'apprentissage. Cela va ajouter une nouvelle personne à la base de données. Tapez le nom de la personne (sans espace) et appuyez sur Entrée.
2. Il va commencer à stocker automatiquement toutes les visages frontales qu'il voit. (Les images faciales sont stockées dans des fichiers PGM dans le dossier `data`, et leurs noms sont ajoutés au fichier texte `train.txt`).
3. La personne en face de la caméra doit déplacer son visage un peu, de sorte qu'il y aura une certaine variation dans les images d'apprentissage.

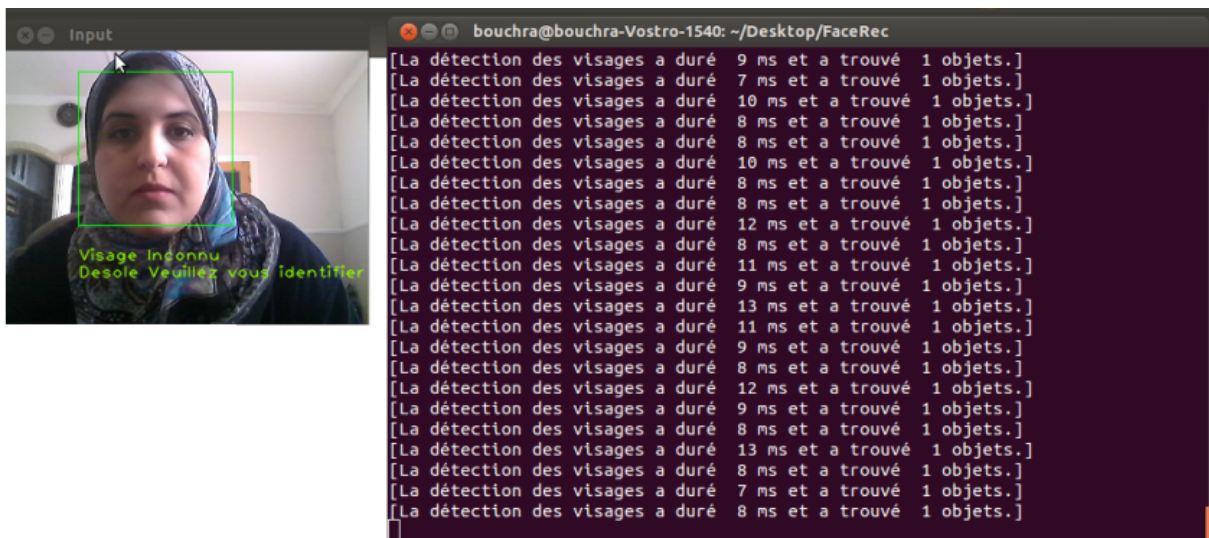


FIGURE 3.5 – Capture d'écran de l'application en fonctionnement.

4. Puis, quand on a assez détecté visages pour cette personne, idéalement plus de 30 pour chaque personne, appuyer sur la touche `t` dans le terminal pour commencer l'apprentissage sur les images qui viennent d'être collectées. Il fera ensuite une pause d'environ 05-30 secondes (selon le nombre de visages et les gens dans la base de données). Le fichier de base de données est appelée `facedata.xml`.
5. Cette application devrait afficher le nom de la personne dans la console à chaque fois qu'il les reconnaît.
6. On répète cette opération depuis l'étape 1 lorsqu'on souhaite ajouter une nouvelle personne, même après l'arrêt du programme.
7. Si On appuie sur la touche Echap dans le terminal (ou GUI) puis il s'arrête en toute sécurité. On peut ensuite exécuter le programme plus tard et il devrait déjà être formé pour reconnaître quelqu'un qui a été ajouté.

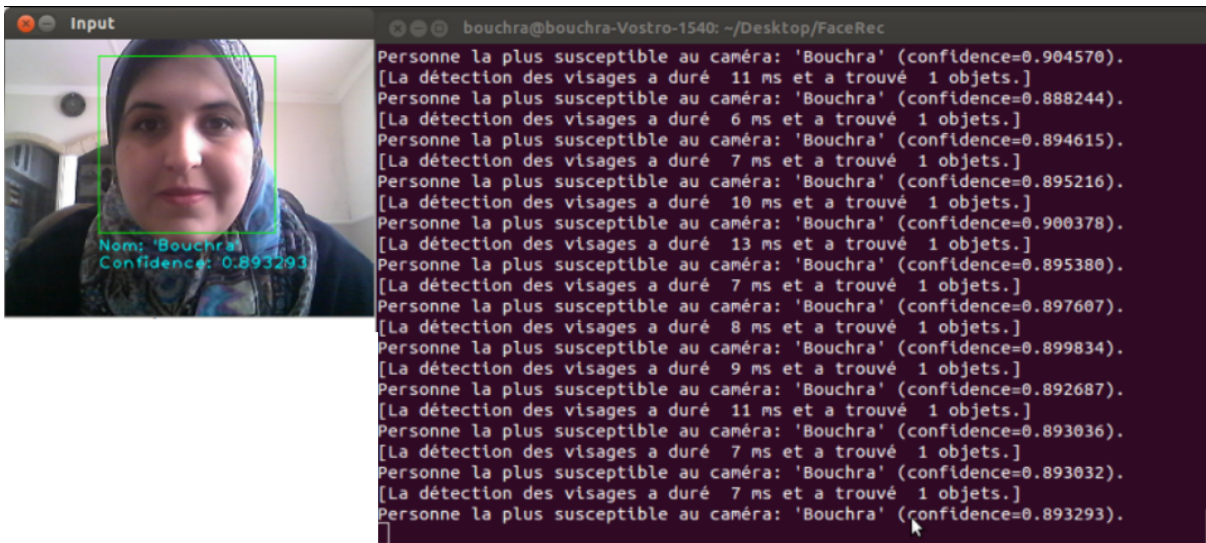


FIGURE 3.6 – Reconnaissance de visage.

Nous pouvons généralement améliorer la précision de la reconnaissance faciale en utilisant plus d’images d’entrée, au moins 50 par personne, en prenant plus de photos de chaque personne, en particulier sous des angles différents et dans des conditions d’éclairage différentes. Si nous ne pouvons pas prendre plus de photos, il existe plusieurs techniques simples que nous pouvons utiliser pour obtenir plus d’images d’apprentissage, en générant de nouvelles images à partir de celles déjà existantes :

- Nous pouvons créer des copies miroir de nos images faciales, de sorte que nous aurons deux fois plus d’images d’apprentissage ;
- Nous pouvons traduire ou redimensionner ou faire pivoter nos images faciales légèrement pour produire de nombreuses images alternatives.
- Nous pouvons également ajouter du bruit pour avoir plus d’images d’apprentissage qui améliorent la tolérance au bruit.

Il est important d’avoir des variations dans les conditions de prise en photo pour chaque personne, de sorte que le classificateur sera en mesure de reconnaître la personne dans différentes conditions d’éclairage et de positions, au lieu de chercher des conditions particulières. Mais il est également important de s’assurer qu’un ensemble d’images pour une personne n’est pas trop varié, comme si nous avons fait pivoter des images de 90 degrés. Cela rendrait le classificateur trop générique et également donner de très mauvais résultats, donc si nous pensons que nous aurons un ensemble d’images avec trop de variance (telles que la rotation de plus de 20 degrés), alors nous pouvons créer des ensembles distincts d’images d’apprentissage pour chaque personne. Par exemple, nous pourrions former un classificateur à reconnaître `PersonA_Face` et un autre pour `PersonA_Gauche` et d’autres `PersonB_Face`, `PersonB_Gauche`, *etc.*

Ensuite, chaque classificateur peut avoir beaucoup de variance, mais pas trop, et il suffit d’associer alors les différents classificateurs pour chaque personne.

Nous pouvons souvent obtenir de très mauvais résultats si nous n'utilisons pas le bon prétraitement sur nos images. Comme je l'ai mentionné plus tôt, égalisation d'histogramme est une méthode de pré-traitement d'image très simple qui peut rendre les choses pires dans certaines situations, de sorte que nous aurons probablement à combiner plusieurs méthodes différentes jusqu'à ce que nous obtenons des résultats décents.

Il est également important de noter, c'est qu'au cœur de l'algorithme, en faisant essentiellement l'équivalent de la soustraction de l'image de test avec une image d'apprentissage pour voir comment ils sont similaires. Cela fonctionne assez bien si un homme a exécuté, mais l'ordinateur pense seulement en termes de pixels et de chiffres. Donc, si vous venez de passer une image de quelques pixels de large, ou utilisez une image qui se trouve à quelques pixels plus gros ou a un peu plus de pixels du front, etc, alors il pense qu'ils sont complètement des images différentes. Cela est également vrai si le contexte est différent, car le code ne connaît pas la différence entre fond et de premier plan (visage), ce qui explique pourquoi il est important de rogner autant de fond que vous pouvez, par exemple en utilisant seulement une faible section à l'intérieur de la face qui ne comporte pas de fond du tout.

C'est pourquoi la reconnaissance du visage est relativement facile à faire en temps réel si nous nous entraînons sur quelqu'un et puis essayer immédiatement de les reconnaître après, puisque ce sera la même caméra, et le fond sera le même, leurs expressions seront presque identiques, l'éclairage sera le même, et la direction que vous les regardez à partir sera le même. Ainsi, vous obtiendrez souvent de bons résultats de la reconnaissance à ce moment-là. Mais une fois que vous essayez de les reconnaître d'une autre direction ou dans une pièce différente ou à l'extérieur ou sur un autre moment de la journée, il donnera souvent de mauvais résultats.

4

Conclusion

L'objectif de ce mémoire est de concevoir et d'implémenter une application de reconnaissance faciale capable, en temps réel, de reconnaître les visages. Vu la quantité de logiciels potentiels (sécurité, réseaux sociaux,...) pouvant se baser sur cette application, celle-ci doit répondre à des exigences de rapidité et de robustesse des résultats.

En ce sens, la première partie de l'application qui consiste à localiser les visages emploie l'algorithme de Viola et Jones, largement reconnu comme méthode fonctionnant en temps réel et fournissant des résultats robustes et fiables.

La seconde partie de l'application s'occupe de la reconnaissance des visages localisés. Notre travail consiste à mettre au point un algorithme efficace 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.

Il faut savoir que eigenfaces est considérée comme la méthode la plus simple et la plus précise de reconnaissance de visage, mais beaucoup d'autres méthodes ou combinaisons de méthodes multiples (beaucoup plus compliqué) peuvent être utilisées.

En effet, capturer une image d'un visage, en particulier à travers une caméra 2D, est simple et non invasif. C'est donc une modalité biométrique facilement tolérée par les utilisateurs, mais les performances de la reconnaissance faciale sont toujours bien en deça de ce que l'on pourrait espérer pour de telles applications. L'augmentation des taux de reconnaissance, la diminution des reconnaissances à tort et l'accélération des temps de réponse lors de la recherche dans les grandes bases de données biométrique sont les défis auxquels doivent faire face les algorithmes actuellement développés.

Ce projet nous a permis de découvrir plus profondément plusieurs aspects du développement d'une application complexe. Il nous a fallu d'abord nous renseigner sur le côté algorithmique de la reconnaissance de visage, et plus généralement de la vision par ordinateur, qui est un domaine en pleine expansion. Cette recherche nous a donc mené à la réalisation d'un moteur de reconnaissance faciale *brut*. Il nous a fallu résoudre plusieurs problèmes algorithmiques ayant plus ou moins de rapport avec les mathématiques, discipline importante dans le traitement d'images en général.

Si nous avions plus de temps, nous aurions aimé développer un peu plus l'interface

graphique afin de compléter son rôle. Il aurait été également intéressant d'affiner le moteur de reconnaissance, en trouvant un moyen de prendre des images correctement cadrées et en déterminant les seuils de façon rigoureuse. Si d'autres projets se font sur ce thème, il serait sans doute intéressant de leur fournir l'interface graphique afin de tester une utilisation par des utilisateurs externes.

Bibliographie

- [1] A. J. Goldstein, L. D. Harmon and A. B. Lesk, Identification of Humman Faces, Proc.IEEE, May 1971, vol.**59**, No. 5, 748-760.
- [2] L. Sirovich and M. Kirby, A Low -Dimensional Procedure for the Characterization of Human Faces, J. Optical Soc. Am. A, 1987, vol.4, No. 3, 559-524.
- [3] M. A. Turk and A. P. Pentland, Face Recognition using Eigenfaces, Proc. IEEE, 1991, 586-591.
- [4] National Science and Technology Concil (NSTC). Comittee on Technology. Face Recongnition. 7 Aout 2006, 10p.
- [5] Martin Willich, 2nd End-User Group Meeting on 3D Face Recognition, February 2008, Berlin.
- [6] S. Rogerson, Smart CCTV. IMIS, February 2002, vol.**12**, No. 1.
- [7] Shan Li and David Saino, August 21, 2011. Advertises strait using facial recognition to tailor pitches. Los Angeles Times.
<<http://www.articles.latimes.com/2011/aug/21/business/la-fi-facial-recognition-20110821/2>>.
- [8] Fabien Soyez, Janvier 2013. Reconnaissance faciale : plus vraiment de la science-fiction. CNET France.
<<http://www.cnetfrance.fr/news/reconnaissance-faciale-plus-vraiment-de-la-science-fiction-39786494.htm>>.
- [9] Robin Yapp, april 2011. Brazilian police to use robocop-style glasses at world cup. The Telegraph.
<<http://www.telegraph.co.uk/news/worldnews/southamerica/brazil/8446088>>.
- [10] Boudjellal. S. Détection et identification de personne par méthode biométrique. Mémoire de Magister : Electronique-Téledétection. Tizi-ouzou : Université Mouloud Mammeri, 95p.
- [11] Ming-Hsuan Yang, David J. Kriegman et Narendra Ahuja. Detecting faces images : A survey. Dans IEEE Transaction on Pattern Analysis and Machine Intelligence, 2002, vol.**24**(1), 34-58.

- [12] Commercial Security Devices[en ligne](page consultée le 01/10/2013).
<<http://www.commercialsecuritydevises.com/fr/détection-de-visage.html>>
- [13] Cheng-Chin, Wen-Kai Tai, Mau-Tsuen Yang, Yi-Ting Huang, and chi-Janng Huang. A novel method for detecting lips, eyes and faces in real time. *Real-Time Imaging*, **9**(4) : 277-287, 2003.
- [14] Wenlong Zheng and Suchendra M. Bhandarkar. Face detection and haking using a boosted adaptative particle filter. *Journal of visual communication and Imog Representation*, **20**(1) : 9-27, 2009.
- [15] H. A. Rowley, S. Baluja, and T. Kanade. Neural Network-based face detection. *IEEE Transachen on Pattern Analysis and Machine Intelligence*, **20**(1) : 23-38, 1998.
- [16] H. Schneidermand and T. Kanade. Probabilistic modeling of local appearance and spacial relation ships for object recognition. *Computer Vision and Pattern Recognition, IEEE Computer Society Conférence on 0* :45, 1998.
- [17] Paul Viola and Michael Jones. Robust real-time object detection. In *Second international work shop on statistical and computation atheories of vision*, Vancouver, Canada, July 13 2001.
- [18] M. Jones et P. Viola, Fast Multi-View Face Detection, *IEEE CVPR*, 2003.
- [19] A. L. C. Barzak, To Ward and Efficient Implementation of a Rotation Invariant Detection using Haar-like Features. *Proceeding of the 2009 IEEE/RSJ international conference on Intelligent robotsand systems. Nouvelle-Zélande, Dunedin. 2005*, P31-36.
- [20] M. Kolsch et M. Turk, Analysis of Rotational Robustness of Hand Detection with a Viola-Jones Detector, *ICPR*, Vol.**3**, 2004.
- [21] M.Van Wambeke, Reconnaissance et suivi de visages et implémentation en robotique temps-réel. Master Ingénieur Civil en Génie Biomédical. Belgique. Université!catholique de Lovain. 2010, 93p.
- [22] M. Behrman, M. Moscovitch, G. Winocur. Facing the issue : New research shows that the brain process faces and objects in separate brain systems. *Journal of Cognitive Neuroscience*, 1997.
- [23] X.Jiang, E. Rosen, T. Zeffiro, J. Vanmeter, V. Blanz, and M. Riesentruber. Evaluation of a shape-based model of human face discrimination using fmri and behavioral techniqueq. *Neuron*, **50**(1) : 159-72, April 2006.
- [24] Paul Viola and Michael Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57 : 137-154, 2004.

-
- [25] Paul Viola et Michael Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, IEEE CVPR, 2001
- [26] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces : Recognition using class specific linear projection. IEEE Trans. PAMI, 1997.
- [27] A. M. Martinez and A. C. Kak. Pca versus lda. IEEE Trans. PAMI, **23**(2) : 228-233, 2001.
- [28] Michael David Kelly. Visual identification of people by computer. PhD thesis, Stanford, CA, USA, 1971.
- [29] KANADE, T. 1973. Computer recognition of human faces. Birkhauser, Basel, Switzerland, and Stuttgart, Germany.
- [30] Brunelli, R., and Poggio, T., Face Recognition : Features versus Template, IEEE Trans. Pattern Anal. Machine Intell., vol.**15**, pp.1042-1052, 1993.
- [31] Manjunath, B. S.,Chellapa, R., Andmalsburg, C. V. D. 1992. A feature based approach to face recognition. In Proceedings, IEEE Conference on Computer Vision and Pattern Recognition. 373-378.
- [32] Martinez, A. 2002. Recognizing imprecisely localized, partially occluded and expression variant faces from a single sample per class. IEEE Trans. Patt. Anal. Mach. Intell. 24, 748-763.
- [33] Moghaddam, A. Pentland, Probabilistic Visual Learning for Object Representation, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.**19**, No. 7, July 1997, pp. 696-710
- [34] Wiskott, L., Fellous, J.-M., Kruger, N., and von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. IEEE Trans. on Pattern Analysis and Machine Intelligence, **19**(7) :775-779
- [35] D. H. Hubel and T. N. Wiesel, Functional architecture of macaque monkey visual cortex, Proc. Royal Soc. B (London), vol. **198**, pp.1-59, 1978.
- [36] Takeo Kanade, Akihiko Yamada. Multi-Subregion Based Probabilistic Approach To- ward Pose-Invariant Face Recognition. In In IEEE International Symposium on Computational Intelligence in Robotics and Automation, Vol. **2** (2003), pp. 954-959
- [37] Face Recognition Home Page. <<http://www.face-rec.org/databases/>>.
- [38] K. Messer, J. Matas, J. Kittler, J. Luettn, G. Maitre, ' XM2VTSDB : The Extended M2VTS Database, Proc. AVBPA'99 Washington DC, 1999.

- [39] P. J. Philips, H. Moon, S. A. Rizvi, P. J. Rauss, The FERET evaluation methodology for face-recognition algorithms, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.**22**, no.10, pp.1090-1100, 2000.
- [40] Le site web de la base de données Color FERET, <<http://www.nist.gov/humanid/colorferet>>.
- [41] A.M. Martinez, R. Benavente, The AR face database, Technical Report 24, Computer Vision Center, Espagne, 1998.
- [42] CVL Face Database, <<http://www.lrv.fri.uni-lj.si/facedb.html>>.
- [43] G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *ICCV*, pages 1-8, 2007.
- [44] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *ACCV*, 2009.
- [45] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, Vol. **3**, No. 1, pp. 71–86, 1991.
- [46] 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*.
- [47] OpenCV 2.4.6.0 documentation. Face Recognition with OpenCV. <<http://www.docs.opencv.org>>

Résumé

La reconnaissance faciale est actuellement un domaine en plein essor. Elle rentre petit à petit dans nos vies au travers de nos téléphones mobiles ou de nos ordinateurs portables. Malgré l'amélioration du taux de détection elle reste actuellement l'objet de nombreuses études. L'objectif de mon projet sera de mettre en oeuvre un système d'identification de personnes, cette identification s'appuie sur des séquences vidéo. Ces séquences sont analysées et on extrait l'information visuelle liée au visage, on construit alors notre base de données et on présente la technique permettant de reconnaître une personne parmi l'ensemble. Le système de reconnaissance faciale est basé sur le principe de Eigenfaces et nous utilisons la bibliothèque OpenCV et son implémentation de l'algorithme de Viola et Jones pour la détection de visage.

Mots-clés: Reconnaissance faciale, Biométrie, Eigenfaces, Viola et Jones, Prétraitement, Base de données, OpenCV.

Abstract

Face recognition is now a burgeoning field. She goes slowly into our lives through our mobile phones or our laptops. Despite the improvement in the detection rate remains currently the subject of many studies. The aim of my project is to implement a personal identification system, this identification is based on video sequences. These sequences are analyzed and visual information related to the face is extracted, then we built our database and the technique to recognize a person from the set are presented. The facial recognition system is based on the principle of Eigenfaces and we use the OpenCV library and her implementassions the Viola and Jones algorithm for face detection.

Keywords: Face recognition, Biometrics, Eingenfaces, Viola and Jones, Preprocessing, Database, OpenCV.

Rapport-Gratuit.com