

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/303907498>

RECONNAISSANCE DE VISAGES HOMMES/FEMMES PAR LES EigenFaces et les SVMs

Technical Report · September 2007

DOI: 10.13140/RG.2.1.3973.3363

CITATIONS
0

READS
1,026

2 authors, including:



Nassim Abbas
University of Science and Technology Houari Boumediene

43 PUBLICATIONS 106 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



New Trends in Neutrosophic Theory and Applications [View project](#)

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'électronique et informatique

Département Télécommunication

Première année post-graduation

Spécialité :Eléctronique

Option : Traitement du Signal et des Images

Mini-projet

Thème

**RECONNAISSANCE DE VISAGES
HOMMES/FEMMES PAR LES EigenFaces et les
SVMs**

Proposé et dirigé par :

Pr. A. HOUACINE

Réalisé par :

Mr. NASSIM ABBAS

PROMOTION : 2006-2007

SOMMAIRE

INTRODUCTION GENERALE

1

CHAPITRE I RECONNAISSANCE DE VISAGES

INTRODUCTION	3
I. METHODES DE RECONNAISSANCE DE VISAGES	4
I.1. Méthodes globales de reconnaissance du visage	4
I.1.1. Corrélation	4
I.1.2. <i>EigenFaces</i> (EF) pervisée	5
I.1.3. DCT	5
I.1.4. Réseaux de neurones	6
I.1.5. Modèle surfacique du visage (3D)	6
I.2. Méthodes locales de reconnaissance du visage	7
I.2.1. <i>EigenObjects</i> (EO)	8
I.2.2. Hidden Markov Models (HMM)	8
I.2.3. Mesures et ratios	9
I.2.4. Couleurs	9
II. APPROCHES RETENUES	10
II.1. <i>EigenFaces</i>	10
II.1.1. Apprentissage	10
II.1.2. Reconnaissance	12
II.1.3. Ajout d'une personne	12
II.2. K-ppv et métriques de distance	13
III. CONCLUSION	14

CHAPITRE II MACHINES A VECTEURS DE SUPPORT (SVMs)

INTRODUCTION	15
I. THEORIE DE L'APPRENTISSAGE SUPERVISE DE Vapnik	15
I.1. Minimisation du risque empirique	17
I.2. Dimension de Vapnik et Chervonenkis (VC-dimension ou dVC)	18
I.3. Contrôle de la capacité de généralisation des processus d'apprentissage	19
I.4. Minimisation du risque structurel	20
II. MACHINES À VECTEURS DE SUPPORT (SVMs)	22
II.1. Cas de données linéairement séparables	23
II.2. Cas de données non-linéairement séparables	28
II.2.1. Marge souple (<i>Soft Margin</i>)	28
II.2.2. Fonctions noyaux	30

III. EXTENSION DES SVMs BINAIRES AU CAS MULTI CLASSES	34
III.1. Résolution du problème SVM multi-classes par une seule optimisation	34
III.2. Résolution du problème multi-classes par des SVMs binaires	35
III.2.1. Un contre tous (OAA pour <i>One Against All</i>)	35
III.2.2. Un contre un (OAO pour <i>One Against One</i>)	36
III.2.3. DAGSVM (Directed Acyclic Graph Support Vector Machines)	37
IV. CONCLUSION	38

CHAPITRE III

EigenFaces APPLIQUEES A LA RECONNAISSANCE DE VISAGES HOMME/FEMME

INTRODUCTION	39
I. MÉTHODOLOGIE SUIVIE	39
I.1. Méthode des EigenFaces	39
I.1.1. Phase d'apprentissage	39
I.1.2. Phase de reconnaissance	43
I.2. Méthode des SVMs	44
I.2.1. Phase d'apprentissage	44
I.2.2. Phase de reconnaissance	45
II. CONCLUSION	47

CHAPITRE IV

PRESENTATION DES RESULTATS, EVALUATION ET COMPARAISON

INTRODUCTION	48
I. DESCRIPTION DES DONNEES UTILISEES	48
I.1. Base d'apprentissage et base de tests	48
I.2. Prétraitement des données pour la reconnaissance	48
II. PRESENTATION DES RESULTATS ET EVALUATION	49
III. PRESENTASION DU LOGICIEL DEVELOPPE	51
III.1. Description générale	51
III.2. Menus	52
VI. CONCLUSION	54
CONCLUSION GENERALE	55
REFERENCES BEBLIOGRAPHIQUES	57

LISTE DES FIGURES

CHAPITRE I

Figure I.1 : Principales techniques de reconnaissance de visages

Figure I.2 : EigenObjects : Image moyenne ainsi que les 6 premiers vecteurs propres pour (a) l'oeil gauche et (b) le nez

CHAPITRE II

Figure II.1: Modèle général de l'apprentissage statistique selon Vapnik

Figure II.2 : L'effet du phénomène de sur-apprentissage

Figure II.3 : Consistance du principe de minimisation du risque empirique. Risque ($R(\alpha)$), risque minimal ($\inf(R(\alpha))$) et risque empirique ($R_{emp}(\alpha)$) en fonction de la capacité du modèle pour un ensemble d'apprentissage fixe

Figure II.4 : Comportement du risque empirique, de l'intervalle de confiance et du risque garanti en fonction de la VC-dimension

Figure II.5 : Principe des SVMs

Figure II.6 : Classification entre deux classes en utilisant les hyperplans : (a). les hyperplans arbitraires l , m et n ; (b). l'hyperplan de séparation optimale avec une grande marge identifiée par les hyperplans canoniques, traversant les vecteurs du support

Figure II.7 : Hyperplans séparateurs dans le cas de données non-linéairement séparables : H est un hyperplan quelconque, H_0 est l'hyperplan optimal, VS : sont les vecteurs support

Figure II.8 : Exemple montrant l'efficacité d'une transformation dans un espace de plus grande dimension pour faciliter la classification

Figure II.9 : Exemple de l'approche OAA qui utilise 3 SVMs binaires pour un pour un problème de classification de 3 classes C_1, C_2 et C_3 qui sont séparées respectivement par trois hyperplans H_1, H_2 et H_3

Figure II.10 : La décision DAG pour la recherche de la meilleure classe parmi 4 classes, la liste des états équivalents pour chaque nœud est indiquée à côté du nœud

CHAPITRE III

Figure III.1 : Organigramme de la reconnaissance de visages hommes/femmes par les EigenFaces

Figure III.2 : Un simple algorithme de la classification géométrique: soient deux classes de visages (représentées par ‘o’ et ‘+’). Calcul de leurs moyennes C_+ et C_- et affecte un visage de test I à une classe qui a la moyenne la plus proche. Cela peut être réalisé en regardant le produit scalaires entre $I - C$ (où $C = (C_+ + C_-)/2$ et $w = C_+ - C_-$, qui change le signe comme l'angle clos à travers $\pi/2$. Notons que la frontière de la décision correspondante est un hyperplan (la ligne pointillée) orthogonal à w

Figure III.3 : Organigramme de la reconnaissance de visages hommes/femmes par les SVMs

CHAPITRE IV

Figure IV.1 : Les vingt images des deux classes (chacune dans une ligne) de la base de données [www1]

Figure IV.2 : EigenFaces : Image moyenne ainsi que les 7 premiers visages propres. Ces images ont été générées à l'aide d'une banque d'images contenant 30 visages de femmes et 30 visages d'hommes [www1]

Figure IV.3 : ACP : Premières composantes principales obtenues à partir (a) de 300 visages femmes, et (b) de 450 visages hommes de la base d'apprentissage [www1]

Figure IV.4 : La comparaison des taux d'erreur contre le nombre des eigenfaces des algorithmes (a) des EigenFaces standard et (b) de SVM appliqués sur les images de la base de données de visages hommes/femmes[www1]

Figure IV. 5 : Fenêtre principale du logiciel

Figure IV. 6 : Menu ouvrir

Figure IV. 7 : Menu outils

Figure IV. 8 : Menu reconnaissance

Figure IV. 9 : Menu aide

LISTE DES TABLEAUX

CHAPITRE IV

Tableau IV.1 : Les résultats du taux d'erreur en moyenne de trois exécutions

INTRODUCTION GENERALE

Avec l'avènement de l'Internet, des échanges et d'acquisition d'informations, la reconnaissance des visages a pris l'ampleur et elle est devenue le premier intérêt des compagnies sécuritaires et des établissements gouvernementaux [1].

Reconnaître un visage d'un individu quels que soient son point de vue, ses dimensions, et son expression faciale ou encore l'angle sous lequel il est éclairé, est une faculté humaine prodigieuse ... Mais c'est peut-être aussi l'une des plus difficiles à reproduire. Pour les fournisseurs de vision industrielle, et plus particulièrement les éditeurs de logiciels de traitement d'images, c'est même un véritable défi. Car ce que "voit" une caméra est souvent très loin de la perception humaine.

Un système de reconnaissance de visages doit être capable de s'adapter des divers changements dans les images de visages. Cependant, « les variations entre les images du même visage dues aux illumination et direction de vue sont presque toujours importantes par rapport aux variations des images dues au changement de l'identité du visage » [2]. Ceci présente un grand défi en reconnaissance du visage. Ainsi, deux issues sont principales, la première est quels dispositifs utilisons nous pour représenter un visage. Une image d'un visage est sujette aux changements en point de vue, illumination, et en expression. Une représentation efficace doit être capable de s'adapter aux changements possibles. La deuxième issue est comment classifions nous une nouvelle image d'un visage en utilisant la représentation choisie.

La méthode des « Machines à Vecteurs du Support » **SVMs** (*Support Vector Machines*) a été introduite récemment par Vapnik et ses collaborateurs [3] comme une méthode très efficace dans le cadre général de la reconnaissance de formes. En effet, la méthode des SVMs permet la classification de données dans des espaces à grande dimension avec un nombre restreint d'exemples d'entraînement sans souffrir du phénomène de Hughes. Elle permet également de séparer des classes très proches avec un faible nombre d'exemples d'apprentissage si ceux-ci sont suffisamment représentatifs, et de minimiser le risque du mal classification non seulement des exemples de l'ensemble d'entraînement mais aussi les exemples de l'ensemble de test.

C'est dans ce cadre, au travers de la reconnaissance de visages hommes/femmes, que se positionne notre travail. Il consiste à la mise en œuvre de deux modèles : un premier modèle supervisé qui permet d'appliquer la méthode des visages propres (*EigenFaces*) en utilisant l'algorithme PPV, et un deuxième modèle supervisé basé sur la méthode des machines à

vecteurs du support (SVMs) en utilisant le noyau linéaire. Ainsi, l'algorithme d'apprentissage utilisé pour la mise en œuvre de ce dernier modèle est proposé par Schölkopf et Smola dans [41]. L'application du deuxième modèle aboutit à la reconnaissance d'un visage avec un meilleur taux d'erreur.

Ce mini-projet est structuré comme suit :

Dans le premier chapitre, nous présentons les techniques de la reconnaissance de visages les plus rencontrées dans la littérature, en nous attardant sur l'approche standard « *EigenFaces* » qui est utilisée pour la mise en œuvre du premier modèle, et ceci en utilisant l'algorithme du K plus proche voisin (K -ppv). Dans le deuxième chapitre, nous abordons la méthode des SVMs qui découle de la théorie d'apprentissage artificiel de Vapnik [4] et son application au problème de la reconnaissance de visages. Dans le troisième chapitre, nous exposons la méthodologie de reconnaissance de visages hommes/femmes que nous adoptons avec la présentation des deux modèles considérés. Le dernier chapitre est consacré à la présentation, l'interprétation, l'évaluation et la comparaison des différents résultats obtenus en utilisant une base de données des visages hommes/femmes formée d'un ensemble d'apprentissage et d'un ensemble de tests.

Nous présentons également dans ce chapitre le logiciel que nous avons développé sous environnement **Windows Xp** en utilisant le langage de programmation **IDL**.

Nous terminons ce mini projet par une conclusion générale et les perspectives éventuelles à ce travail.

CHAPITRE I

RECONNAISSANCE DE VISAGES

INTRODUCTION

Au cours de ces dernières années, l'accélération des progrès technologiques, notamment dans la fabrication des systèmes de vision industrielle, et plus particulièrement l'édition de logiciels de traitement d'images, a contribué à l'introduction de la reconnaissance des individus dans plusieurs champs d'applications, comme un outil efficace et performant d'aide à la prise de décisions. Parmi ces applications, on cite : l'authentification d'identité, le contrôle d'accès, et la surveillance, etc. [5] [2].

Pour toutes ces applications, l'exploitation des images de visages prises à partir des bases de données nécessite différents types de traitements numériques d'images. Parmi ces traitements figure la reconnaissance de visages.

Plusieurs techniques de reconnaissance d'individus ont été développées au cours des dernières années. La plupart d'entre elles ont le visage comme zone d'intérêt; une tâche qui est par ailleurs un problème de reconnaissance des formes assez complexe. En effet, contrairement à certaines problématiques comme la reconnaissance de caractères manuscrits, le nombre de classes à distinguer est très élevé et chacune d'elle ne possède qu'un nombre restreint d'exemples. Ces conditions particulières nuisent donc à certaines techniques d'apprentissage automatique qui nécessitent un grand nombre de données afin d'apprendre efficacement.

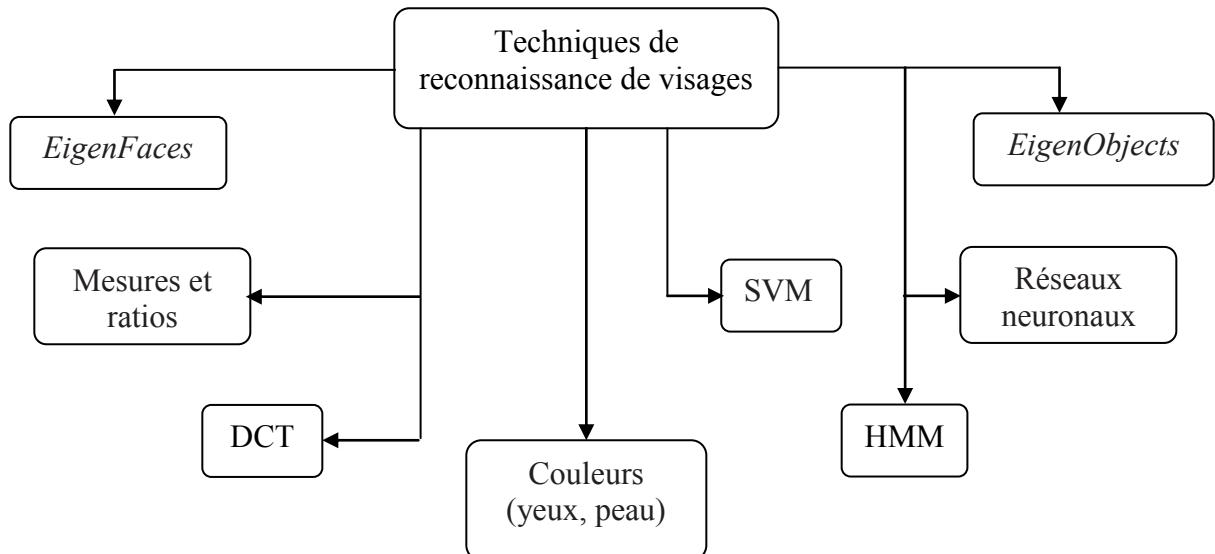


Figure I.1. Principales techniques de reconnaissance de visages [6].

Plusieurs techniques seront donc décrites en section I, comme entre autres les *EigenFaces* et les réseaux de neurones. Ensuite, en section II, on abordera les méthodes privilégiées dans le cadre de ce mini-projet.

I. METHODES DE RECONNAISSANCE DE VISAGES

La reconnaissance de visages peut être accomplie par une très grande variété de méthodes. La figure I.1 illustre un schéma représentant les techniques les plus rencontrées dans la littérature.

I.1. Méthodes globales de reconnaissance du visage

La première grande famille de méthodes de reconnaissance concerne celles qui utilisent le visage au complet comme source d'information et ce, sans segmentation de ses parties.

Dans la majorité des cas, les images sont représentées par un vecteur de pixels généré par la concaténation de toutes les colonnes de l'image. Ainsi, une image en niveaux de gris de dimensions de 150×150 pixels possédera une représentation vectorielle de 22500 éléments. Finalement, les couleurs ne sont habituellement pas utilisées par les méthodes globales de reconnaissance, ce qui simplifie un grand nombre d'opérations.

I.1.1. Corrélation

La technique de corrélation est basée sur une comparaison simple entre une image test et les visages d'apprentissage. Celui d'entre eux se trouvant à la plus faible distance du visage test sera sélectionné comme premier choix.

Plusieurs métriques peuvent être utilisées afin d'évaluer cette valeur comme par exemple les distances L_1 (city-block) et L_2 (euclidienne), la cross-corrélation, la distance de Mahalanobis, etc. Ce processus de décision est communément appelé algorithme du K plus proche voisin (K -ppv) et sera présenté avec davantage de détails à la sous-section II.2.

Malgré sa grande simplicité, cette méthode n'offre cependant pas d'avantages particulièrement intéressants. En effet, n'utilisant pas d'informations de plus haut niveau, la technique de corrélation offre peu de robustesse face aux expressions faciales, aux variations d'éclairage et aux changements physiques (par exemple : barbe).

I.1.2. *EigenFaces* (EF)

L'utilisation de méthodes statistiques appliquées à la modélisation et à la reconnaissance de visage est largement répandue. Kirby et Sirovich ont d'ailleurs [7] utilisé la transformée K-L (c-à-d : Karhunen-Loeve) afin de coder des visages et ainsi réduire la dimensionnalité de leur représentation.

En 1991, Turk et Pentland [8] introduisirent le concept d'*EigenFaces* à des fins de reconnaissance. Basée sur une analyse en composantes principales (ACP), la méthode des EF repose sur une utilisation des premiers vecteurs propres comme visages propres, d'où le terme *EigenFaces*.

La base formée par ces vecteurs génère alors un espace utilisé pour représenter les images des visages. Les personnes se voient donc attribuer un vecteur d'appartenance pour chacune de leur image.

Cela étant dit, la reconnaissance est réalisée en comparant les coefficients de projection d'un visage test avec ceux appartenant aux visages d'entraînement. La méthode est relativement rapide en phase de reconnaissance et peut également bénéficier de plusieurs optimisations algorithmiques [8]. Cette méthode sera présentée avec davantage de détails à la sous-section II.1.

I.1.3. DCT

L'utilisation de la transformée de cosinus discrète (*Discrete Cosine Transform* ou DCT) [12] à des fins de reconnaissance de visage est assez récente [13]. Similaire aux *EigenFaces* d'un point de vue mathématique, elle est par contre beaucoup plus rapide, tant en phase d'apprentissage qu'en phase de reconnaissance.

Cela étant dit, chaque image de visage est représentée par un vecteur composé des premiers coefficients de la transformée. Lorsqu'un visage est présenté au module de reconnaissance, sa transformée est calculée et un certain nombre de coefficients est retenu pour comparaison avec ceux de la banque de données. Cette dernière étape est réalisée à l'aide de la distance L_1 ou avec d'autres métriques pertinentes.

I.1.4. Réseaux de neurones

Cette technique envisage l'utilisation des réseaux de neurones comme engin d'apprentissage et de reconnaissance. Pour débuter, une image brute (ou prétraitée) de dimensions fixes constitue habituellement la source d'entrée des réseaux. Les dimensions doivent être établies au préalable car le nombre de neurones sur la couche d'entrée en dépend.

Cela étant dit, plus les dimensions de l'image sont élevées, plus la complexité et le temps d'apprentissage augmentent. En effet, pour une image de dimensions 150×150 pixels, 22500 neurones seront requis sur la couche d'entrée, ce qui est énorme. L'apprentissage efficace (c-à-d : la convergence) d'un tel réseau est également douteux.

Le nombre de sorties du réseau dépend par ailleurs directement du nombre de classes visages à discriminer. Il est donc évident qu'un apprentissage incrémental (avec de nouvelles classes et non de nouveaux exemples) sera difficile et requerra des ajustements directs à l'architecture du réseau.

Certains auteurs ont par ailleurs utilisé des variantes de la technique de base en modifiant les données d'entrée. Les coefficients de projections d'images dans un espace des visages (*EigenFaces*) peuvent par exemple être utilisés comme source d'information [14]. Cette méthode peut évidemment être étendue aux coefficients de DCT, de Fourier, etc.

Les informations peuvent également être fusionnées ensemble avant d'être acheminées au réseau : c'est le cas par exemple de la concaténation d'une image prétraitée (dimensions 30×40 pixels) et de l'échantillonnage d'histogrammes de couleurs RGB [15].

I.1.5. Modèle surfacique du visage (3D)

Cette méthode de reconnaissance repose sur l'utilisation d'un modèle tridimensionnel du visage. Pour que cette technique soit réellement efficace, une vue rapprochée du visage est nécessaire pour chacune des caméras impliquées dans l'acquisition : l'utilisation d'une caméra combinée avec un laser de faible puissance est également envisageable pour réaliser une acquisition tridimensionnelle.

Dans certains cas, il est possible de réaliser de la stéréo dense, c'est-à-dire d'extraire un grand nombre de points dans une zone relativement restreinte. Celle-ci garantie alors de

meilleures précisions sur les mesures ainsi qu'une résolution accrue. Une fois l'appariement des points réalisé, le modèle peut être normalisé et stocké dans la base de données.

Lorsqu'un individu se présente devant les caméras, la même procédure s'applique, mais suivie d'une étape de comparaison. En effet, le modèle à reconnaître doit être comparé à tous les modèles de la base de données, ce qui représente un travail colossal. Une réduction de calculs est donc impérative pour minimiser la complexité de ce problème d'optimisation.

Ceci peut être réalisé en alignant les centres des yeux des deux modèles. Il ne reste alors qu'à mesurer l'erreur entre les deux surfaces. Certains auteurs [16] ont d'ailleurs proposé l'utilisation d'une distance d'Hausdorff modifiée pour réaliser ce calcul.

Il est finalement envisageable de prélever certaines mesures sur le modèle du visage, comme la distance réelle entre les composantes du visage (par exemple : distance entre les yeux) ou leurs dimensions. Ainsi, ces informations pourraient être utilisées ensuite pour la reconnaissance de visages.

I.2. Méthodes locales de reconnaissance du visage

Le principal désavantage des méthodes globales réside au niveau de détails utilisé. En effet, lorsqu'une technique s'attarde aux variations dans toute une image, elle tentera de limiter l'impact des changements locaux et concentrera le maximum d'énergie pour représenter adéquatement l'ensemble de l'image (par exemple : *EigenFaces*). Par contre, il arrive parfois que des personnes possèdent une physionomie faciale très semblable, mais que certains petits détails diffèrent grandement. Ce serait le cas par exemple d'une personne possédant un nez imposant.

En utilisant une méthode locale, davantage d'énergie sera accordée aux fins détails locaux, évitant ainsi le bruit causé par les cheveux, les chapeaux, la barbe, etc. De plus, certaines parties du visage sont relativement invariantes pour une même personne malgré ses expressions faciales; c'est le cas notamment des yeux et du nez. Ceci demeure vrai tant que ces caractéristiques du visage ne sont pas en occultation. Les prochains paragraphes porteront donc sur les principales techniques de reconnaissance locales.

I.2.1. *EigenObjects* (EO)

Basés sur les mêmes principes théoriques que la méthode des *EigenFaces* abordée à la section I.1.2, les *EigenObjects* visent cette fois certaines parties bien précises du visage. La personne peut par exemple être reconnue uniquement grâce à ses yeux.

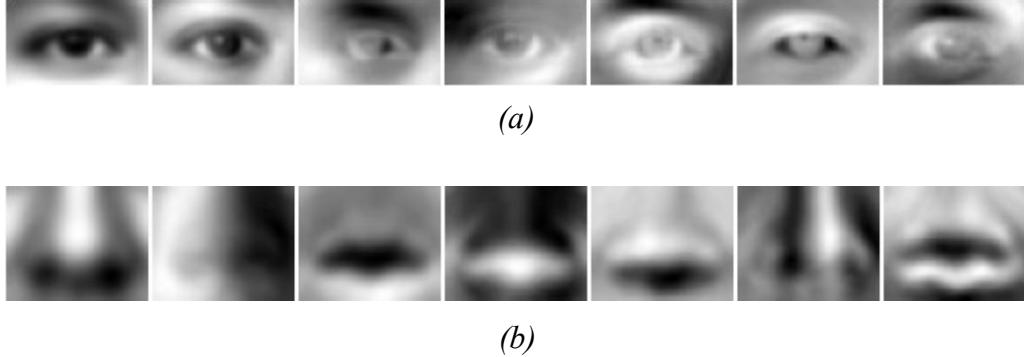


Figure I.2. *EigenObjects* : Image moyenne ainsi que les 6 premiers vecteurs propres pour (a) l’œil gauche et (b) le nez [6].

Pour réaliser l’apprentissage, un module de ce type doit tout d’abord procéder à une ACP des yeux contenus dans la banque de visages. L’espace des yeux (*eye space*) ainsi construit pourra alors servir au processus de reconnaissance qui est identique à celui utilisé pour les *EigenFaces*. Des résultats intéressants ont notamment été obtenus par Moghaddam et Pentland [17] sur une portion de la banque d’images FERET [10].

La figure I.2 (a) illustre un exemple d’œil gauche moyen ainsi que les 6 premiers *eigeneyes* associés à la banque FERET, alors que la figure I.2 (b) illustre les mêmes informations pour le nez.

I.2.2. Hidden Markov Models (HMM)

Les modèles de Markov cachés (HMM) sont utilisés depuis plusieurs années pour la détection et la reconnaissance du visage [18]. Différentes variantes ont également été proposées mais celle des (*Embedded HMM*) génère des résultats supérieurs [19] aux méthodes HMM de base.

Les *Embedded HMM* sont caractérisés par l’utilisation d’un HMM 1D de base, modélisant l’apparence du visage de haut en bas. Ensuite, chacun des états de ce modèle général contient un

autre HMM 1D, dénommé *embedded* (c-à-d : incorporé). Ceux-ci modélisent cette fois l'apparence du visage de la gauche vers la droite.

Reposant sur certains coefficients de la transformée en cosinus discrète (DCT) comme source d'observations, les *Embedded HMM* constituent un algorithme de reconnaissance très performant. Or, les temps d'exécution des phases d'apprentissage et de test sont relativement élevés, nuisant donc à son utilisation en temps réel sur d'immenses banques d'images.

I.2.3. Mesures et ratios

Lorsque la localisation des différentes parties (yeux, nez, oreilles, bouche, menton, début des cheveux, etc.) du visage est complétée, certaines mesures en pixels peuvent être prélevées à des fins de reconnaissance [20, 21]. Ces différentes valeurs peuvent être regroupées en deux catégories importantes, soient les dimensions des parties du visage et leurs distances relatives. Les mesures prélevées peuvent par exemple être les particularités suivantes :

- Dimensions de la tête, du nez, de la bouche, etc. ;
- Épaisseurs des sourcils, de la bouche, etc. ;
- Forme du menton (représenté par des distances relatives au centre de la bouche) [21] ;
- Positions relatives (positions relatives par référence aux coordonnées des yeux) du nez, des sourcils, de la bouche, etc.

Afin que cette technique soit efficace, l'image doit être préalablement normalisée sans altérer son ratio original.

Cela étant dit, la pose du visage doit être semblable à celle observée lors de l'apprentissage. Cette limitation étant difficilement respectée en pratique, l'utilisation de cette approche ne peut être efficace que dans un environnement contrôlé (par exemple : une personne devant se présenter à une station d'identification).

I.2.4. Couleurs

Une des caractéristiques les plus discriminantes entre les personnes repose sur la couleur. En effet, on peut identifier rapidement une personne de notre entourage selon la couleur de ses cheveux (par exemple : rouge ou blanc). Il est donc possible d'intégrer plusieurs mesures de couleurs pour la reconnaissance, notamment pour les yeux, les cheveux et la peau [22].

Cela étant dit, les mesures dépendent énormément de l'éclairage et elles sont par ailleurs assez bien contournables (c-à-d : verres de contact, bronzage, perruque, etc.) par un imposteur. Cette méthode devrait donc être utilisée conjointement avec d'autres techniques pour améliorer sa robustesse.

Pour terminer, les caméras utilisées lors des acquisitions influencent également le niveau de performance de cette méthode. En effet, des couleurs différentes peuvent être obtenues avec une même caméra, dépendant des paramètres sélectionnés. Cette conclusion s'applique également à des caméras de marques différentes.

II. APPROCHES RETENUES

Une seule approche a été sélectionnée, soit les *EigenFace* [23]. La sous-section suivante présentera cette méthode en détails.

II.1. *EigenFaces*

Il est pratiquement impossible de concevoir un système de reconnaissance du visage sans envisager la très populaire technique des *EigenFaces*, introduite en 1991 par Turk et Pentland [23].

Cette méthode est tout d'abord basée sur une analyse en composantes principales (ACP) appliquée sur l'ensemble de visages d'une banque d'entraînement. Elle consiste essentiellement à effectuer une réduction de dimensionnalité en codant les visages dans une nouvelle base formée par les premiers vecteurs propres (c-à-d : *EigenFaces*) provenant du calcul de l'ACP.

Les *EigenFaces* associés aux plus fortes valeurs propres représentent donc, dans l'espace des images, les directions dans lesquelles les variations sont les plus marquées. C'est ainsi que les premiers visages propres représentent habituellement les différences d'éclairage ainsi que les personnes portant des lunettes ou une barbe.

II.1.1. Apprentissage

La phase d'apprentissage (ou de modélisation) des *EigenFaces* se déroule comme suit :

1. Un visage moyen ψ est calculé à partir des N images d'entraînement I_i de dimensions

$L \times H$:

$$\Psi = \frac{1}{N} \sum_{i=1}^N I_i \quad (\text{I.1})$$

2. Ce visage moyen est soustrait des images d'apprentissage (on élimine donc les ressemblances pour se concentrer sur les différences), ce qui génère les valeurs de différences Φ_i associées à chacune des images :

$$\Phi_i = I_i - \Psi \quad (\text{I.2})$$

3. La matrice de covariance C est construite (approche inter-pixels) :

$$C = \frac{1}{N} \sum_{i=1}^N \Phi_i \Phi_i^T = A A^T \quad (\text{I.3})$$

où chacune des colonnes de A représente un vecteur de différences, soit

$$A = [\Phi_1, \Phi_2, \dots, \Phi_N] ;$$

4. Étant donné les dimensions élevées de $C(LH \times LH)$, une approche inter-images est privilégiée (Il serait par ailleurs presque irréalisable d'effectuer l'inversion d'une matrice 22500×22500 correspondant à des images de dimensions 150×150 pixels). Le calcul se limite [23] donc à une matrice $L = A A^T$ dépendant du nombre d'images dans la banque d'apprentissage (c-à-d : dimensions $N \times N$) ;
5. Calcul des valeurs et vecteurs propres de la matrice L ;
6. Le visage propre u_i associée à la $i^{\text{ème}}$ valeur propre est formé en utilisant les vecteurs propres v_i de la matrice L :

$$u_i = \sum_{k=1}^N v_{ik} \Phi_k \quad (\text{I.4})$$

7. Les M premiers vecteurs propres (EF) (c-à-d : associés aux plus fortes valeurs propres) sont conservés. Ils définissent ainsi l'espace de visages (*face space*) ;
8. Les images originales sont projetées dans l'espace des visages pour former une suite de coefficients d'appartenance, ce qui donne pour une image I_i :

$$\omega_k = u_k^T \Phi_k \quad (\text{I.5})$$

où $k = 1, \dots, M$;

9. Ces coefficients forment alors un vecteur représentant l'image I_i :

$$\Omega_i = [\omega_1, \omega_2, \dots, \omega_M] \quad (\text{I.6})$$

Une fois l'apprentissage complété, les différentes représentations d'un individu

peuvent être regroupées afin de former une classe. Ceci peut être réalisé notamment en calculant une moyenne des différents vecteurs i correspondants à la personne [23].

Il est également possible de considérer les représentations individuellement et ainsi les utiliser directement avec l'algorithme du K -ppv. C'est d'ailleurs cette approche qui a été retenue dans l'implémentation du système.

II.1.2. Reconnaissance

Lorsqu'un visage est présenté au système, la procédure d'identification consiste à :

1. Projeter l'image d'entrée I dans l'espace des visages, ce qui engendre un coefficient d'appartenance ω_k à un *EigenFace* u_k en utilisant l'équation 1.5 ;
2. Les coefficients d'appartenance forment alors un vecteur de représentation de taille M ;
3. Ce dernier est comparé avec ceux obtenus lors de la phase d'apprentissage à l'aide de l'algorithme K -ppv (voir sous-section II.2) en appliquant une métrique de distance particulière.

II.1.3. Ajout d'une personne

Lorsqu'une nouvelle personne (le même processus est nécessaire lorsque de nouvelles images sont ajoutées à des personnes déjà présentes dans la banque) est ajoutée à la base de données, la méthode classique consiste à refaire l'apprentissage complet (c-à-d : ACP pour déterminer les nouveaux visages propres). Il existe cependant deux alternatives à cette solution.

Premièrement, lorsque la banque d'apprentissage est relativement grande et que les visages qu'on y retrouve sont représentatifs, il est possible d'utiliser directement les EF existants afin de calculer les coefficients de projection des nouvelles images.

Il serait par contre intéressant à long terme (et après plusieurs ajouts de personnes) de réaliser un réapprentissage complet afin d'obtenir des visages propres plus représentatifs de la base de données.

La deuxième méthode est relativement récente et repose sur une fusion d'espaces de visages [24]. Il est en effet possible de fusionner deux *face space* sans toutefois nuire au processus de reconnaissance (par exemple : altération des visages propres). Donc en pratique, un espace temporaire est généré à partir des nouvelles images pour être ensuite fusionné avec l'espace principal.

Cette opération est également avantageuse en terme de temps de calcul [24] comparativement à un re-calculation complet. Par ailleurs, comme l'espace des visages est modifié, il

est primordial de re-projeter tous les visages d'entraînement afin de reconstruire les représentations.

II.2. K-ppv et métriques de distance

Pour un bon nombre de techniques, l'algorithme du K plus proche voisin (*K-ppv*) est utilisé lors de la phase de reconnaissance. C'est le cas notamment des techniques *EigenFaces*, *EigenObjects* ainsi que DCT abordées précédemment. Le *K-ppv* se base tout simplement sur une liste ordonnée des voisins les plus près d'une image test.

Pour ce faire, le vecteur test est comparé avec chacun des prototypes constituant la banque d'apprentissage. Pour chacune de ces comparaisons, une distance est calculée et agit comme taux de ressemblance avec un certain prototype. Cette distance peut être mesurée par différentes métriques, dont voici une liste partielle :

- **City-block (L_1) :** La distance L_1 consiste à calculer la somme des différences absolues entre les éléments des vecteurs, soit la fonction suivante :

$$L_1 = |U - V| = \sum_{i=1}^N |U_i - V_i| \quad (\text{I.7})$$

où U et V représentent les vecteurs à comparer et N la taille des vecteurs.

- **Euclidienne (L_2) :** La distance L_2 ou distance euclidienne entre deux vecteurs consiste à calculer la racine de la somme des différences au carré, soit :

$$L_2 = \|U - V\| = \sqrt{\sum_{i=1}^N (U_i - V_i)^2} \quad (\text{I.8})$$

où U et V représentent les vecteurs à comparer et N la taille des vecteurs.

- **Angle :** L'angle négatif entre les deux vecteurs [25] équivaut à l'équation :

$$\text{Angle} = -\frac{\langle U, V \rangle}{\|U\| \|V\|} = -\frac{\sum_{i=1}^N U_i V_i}{\sqrt{\sum_{i=1}^N U_i^2 \sum_{i=1}^N V_i^2}} \quad (\text{I.9})$$

où U et V représentent les vecteurs à comparer et N la taille des vecteurs.

Une fois toutes les distances mesurées, une liste ordonnée croissante est générée afin de départager les candidats. Habituellement, K se voit assigner une valeur de 1, ce qui signifie que

le prototype de la banque d'apprentissage le plus proche est sélectionné. Si K est supérieur à 1, le prototype qui possède la majorité de votes parmi les K premiers voisins sera choisi.

III. CONCLUSION

Dans ce chapitre, nous avons présenté un bref aperçu sur les méthodes de reconnaissance supervisée les plus rencontrées dans la littérature. En effet, les méthodes de reconnaissance supervisée peuvent être développées selon une approche statistique, géométrique, biométrique, ou discriminante.

Dans ce mini-projet nous nous intéressons à la méthode des SVMs qui est une méthode discriminante récente et très prometteuse pour la reconnaissance de visages.

Le chapitre suivant sera consacré à la méthode des machines à vecteurs du support qui découle de la théorie de l'apprentissage statistique de Vapnik. Le formalisme mathématique de cette méthode pour la reconnaissance de visage sera alors présenté.

CHAPITRE II

MACHINES A VECTEURS DE SUPPORT (SVMs)

INTRODUCTION

Les machines à vecteurs de support désignées par SVMs (*Support Vector Machines*) sont une méthode et un terme inventé principalement par Vapnik en 1992 dans son livre « *The nature of statistical learning theory* » [4].

L'aptitude à généraliser des résultats obtenus à partir d'un échantillon limité constitue l'enjeu essentiel de l'apprentissage artificiel (*Machine Learning*) sur lequel est fondé la méthode des SVMs. Il ne plus à démontrer que la seule minimisation du risque empirique (erreur d'apprentissage) ne garantit pas une faible erreur un corpus de test (erreur réelle).

Du point de vue conceptuel, la notion de risque structurel introduite par Vapnik donne une borne de l'erreur de test en fonction de l'erreur d'apprentissage et de la complexité du modèle. En pratique, les SVMs offrent un moyen opérationnel pour minimiser le risque structurel, ce qui explique l'engouement qu'elles ont suscité dans la communauté scientifique.

L'apprentissage artificiel s'inscrit comme un axe de recherche dans le domaine de l'Intelligence Artificielle. Il en apparaît comme une composante importante car, l'intelligence de la machine dépend autant de ses connaissances que de ces capacités de raisonnement.

L'apprentissage artificiel est une activité circulaire, partant de ce qui est aisément accessible (des exemples, des points, une base de données), cherche à construire automatiquement des connaissances, règles ou modèles. La finalité de ces connaissances est de permettre la prise automatique de décisions et/ou la compréhension du phénomène illustré par les exemples. Rappelons enfin, que l'apprentissage à partir d'exemples est appelé aussi apprentissage supervisé.

I. THEORIE DE L'APPRENTISSAGE SUPERVISE DE Vapnik

D'après Vapnik [4], le modèle général de l'apprentissage statistique est constitué des trois composantes de la figure II.1 :

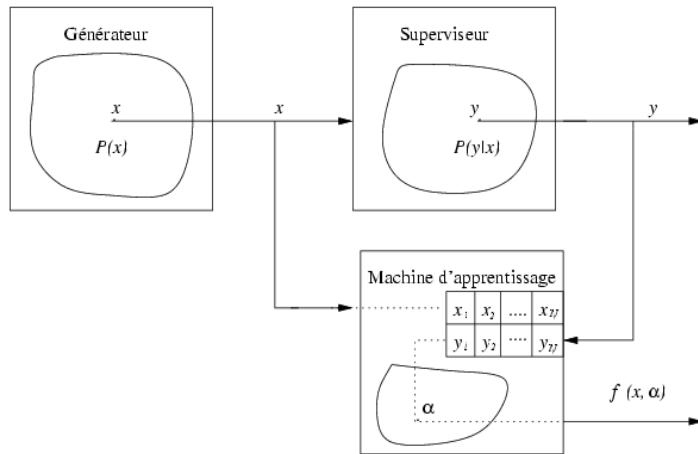


Figure II.1. Modèle général de l'apprentissage statistique selon Vapnik.

1. **Un générateur** de vecteurs aléatoires $x \in R^n$, tiré indépendamment les uns des autres. Ils sont identiquement distribués suivant une distribution de probabilité $P(x)$, fixe mais inconnue.
2. **Un superviseur** qui retourne une valeur de sortie désirée y pour chaque vecteur d'entrée x , suivant une distribution conditionnelle de probabilité $P(y/x)$, qui est aussi fixe mais inconnue.
3. **Une machine (ou un modèle) d'apprentissage (M)** capable de réaliser les fonctions $f(x, \alpha)$, $\alpha \in \Lambda$ où Λ est un ensemble de paramètres.

Le problème d'apprentissage statistique à partir d'exemples étiquetés, appelé *apprentissage supervisé*, consiste à rechercher la fonction f qui approche au mieux les réponses du superviseur, dans l'ensemble des fonctions $\Omega = \{f(x, \alpha), \alpha \in \Lambda\}$ réalisables par le modèle choisi M . La seule connaissance sur le problème dont on dispose, *a priori*, pour faire le choix de cette fonction est l'ensemble :

$$D = \{(x_1, y_1), \dots, (x_N, y_N)\}$$

appelé *ensemble d'apprentissage*, composé de N observations indépendantes et identiquement distribuées suivant $P(x, y)$.

La *fonction perte* formalise la notion de la qualité d'apprentissage. Notons $L(y, f(x, \alpha))$ la fonction coût qui correspond à la réponse $f(x, \alpha)$ associée à l'entrée x et à la réponse y donnée par le superviseur. Cette fonction peut prendre des formes différentes, selon le type du problème auquel on est confronté. Par exemple, pour la reconnaissance de formes, la fonction coût s'écrit sous la forme suivante :

$$L(y, f(x, \alpha)) = \begin{cases} 0 & \text{si } y = f(x, \alpha) \\ 1 & \text{si } y \neq f(x, \alpha) \end{cases}$$

L'espérance mathématique de la fonction coût définit le *risque fonctionnel* $R(\alpha)$:

$$R(\alpha) = \int L(y, f(x, \alpha)) dP(x, y)$$

Le but de l'apprentissage est donc de trouver la fonction $f(x, \alpha_0)$ qui minimise le risque fonctionnel $R(\alpha)$, parmi la famille des fonctions $f(x, \alpha)$ réalisables par notre machine M , dans le cas où $P(x, y)$ n'est pas connue et où toute l'information dont on dispose est l'ensemble d'apprentissage D .

I.1. Minimisation du risque empirique

On ne peut pas minimiser directement le risque fonctionnel $R(\alpha)$ à partir de l'ensemble d'apprentissage. Le principe inductif de la Minimisation de Risque Empirique (**MRE**) consiste à remplacer le risque $R(\alpha)$ par le risque empirique [26] :

$$R_{emp}(\alpha) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i, \alpha)) \quad (\text{II.1})$$

qui est l'espérance empirique de la fonction coût sur l'ensemble d'apprentissage. Ainsi nous approchons la fonction $f(x, \alpha_0)$ qui minimise le risque fonctionnel $R(\alpha)$ par la fonction $\hat{f}(x, \hat{\alpha})$ qui minimise le risque empirique $R_{emp}(\alpha)$.

Le problème de l'apprentissage à partir d'exemples (principe d'induction) est qu'il ne permet pas pour un modèle M donné, le contrôle de la généralisation qui est la première qualité qu'on attend d'un modèle. De façon intuitive, un modèle d'apprentissage a une capacité de généralisation lorsqu'il produit, en moyenne sur tout les exemples, des réponses proches de la valeur désirée, ce qui revient à dire que son risque est faible. Un faible risque empirique, par contre, n'est pas une garantie d'une bonne généralisation. Ce problème est bien connu sous le nom de sur-apprentissage (*overfitting*). La figure II.2 empruntée également à Vapnik [4] représente une interprétation du phénomène du sur-apprentissage.

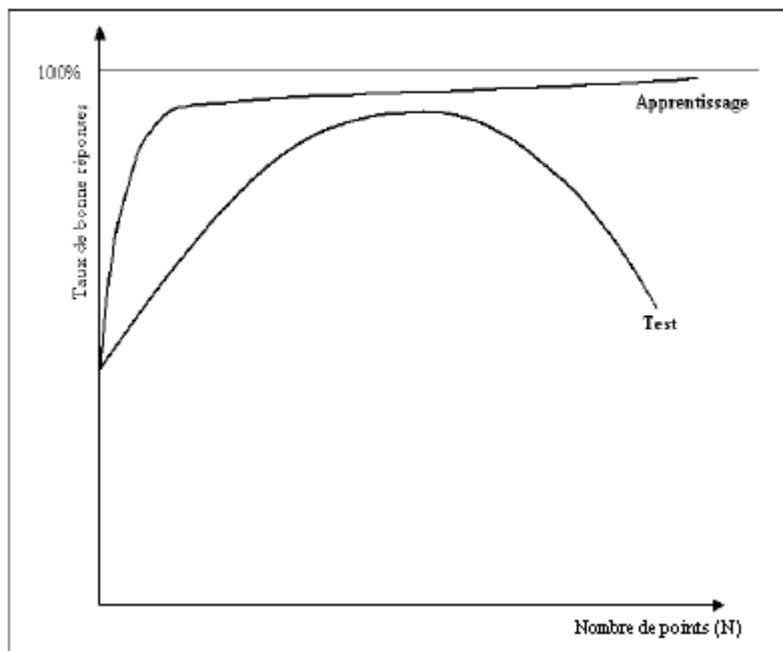


Figure II.2. L'effet du phénomène de sur-apprentissage.

Pour résoudre ce problème, Vapnik et Chervonenkis ont mis en place une nouvelle approche qu'ils ont nommée « *Minimisation du Risque Structurel* » (**MRS**) basé sur un autre concept nommé « dimension de Vapnik et Chervonenkis ».

I.2. Dimension de Vapnik et Chervonenkis (VC-dimension ou dVC)

Vers la fin des années soixante, Vapnik et Chervonenkis ont introduit un nouveau concept nommé VC-dimension qui présente une sorte de mesure de capacité de calcul d'une famille de fonctions Ω . Ce concept a été proposé dans un premier temps pour les fonctions indicatrices avant d'être étendue aux fonctions à valeurs réelles.

Soit un ensemble de fonctions indicatrices Ω de l'ensemble D . On dit que Ω sépare $A \subseteq D$ si $\forall B \subseteq A, \exists f_B \in \Omega$ tel qu'on ait : $f_B(A / B) = 0$ et $f_B(B) = 1$.

Ainsi la dimension de Vapnik peut être définie comme le cardinal maximal des sous ensembles A séparables par Ω .

Définition

La dimension de Vapnik-Chervonenkis ou VC-dimension d'une famille de fonctions Ω est le nombre maximum d'exemples d'un ensemble de données D qui peuvent être séparables par la famille de fonctions Ω [4].

Notons que la dimension d'une famille peut être infinie.

Après avoir donné la définition de VC-dimension, nous pouvons énoncer le théorème suivant :

Théorème

Pour que le principe de MRE soit consistant, indépendamment de la distribution de probabilité des exemples, il suffit que la famille de fonctions que le système d'apprentissage est capable d'implémenter ait une VC-dimension finie.

Ce théorème donne seulement une condition suffisante concernant la consistance de l'approche MRE, et il fournit également des bornes sur la vitesse de convergence du processus d'apprentissage. Ainsi, Vapnik et Chervonenkis ont démontré que pour $N > h$ où h est la VC-dimension de la famille Ω et N est la taille de l'ensemble d'exemples, l'inégalité suivante sera vérifiée avec une probabilité au moins égale à $1 - \eta$ (avec η une constante telle que $0 < \eta < 1$) :

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h \left(\log\left(\frac{2N}{h}\right) + 1 \right) - \log\left(\frac{\eta}{4}\right)}{N}} \quad (\text{II.2})$$

I.3. Contrôle de la capacité de généralisation des processus d'apprentissage

À partir de l'inégalité (II.2), le second membre appelé le risque garanti est composé de deux termes, le risque empirique et une quantité qui dépend du contenu N/h , appelée «intervalle de confiance», puisqu'elle représente la différence entre le risque empirique $R_{emp}(\alpha)$ et le risque $R(\alpha)$.

Si le rapport N/h est suffisamment grand, la minimisation du risque empirique suffit pour contrôler la capacité de généralisation du modèle. En fait une faible valeur du risque empirique permet de garantir une faible valeur du risque fonctionnel $R(\alpha)$. En revanche, lorsque la taille N de l'échantillon est petite, le rapport N/h est petit et l'intervalle de confiance peut prendre une valeur importante. Ceci implique que la minimisation du risque empirique n'est suffisante pour garantir une valeur minimale du risque fonctionnel. Ainsi pour minimiser le second membre de l'inégalité (II.2), il faut minimiser simultanément les deux termes qui le composent; pour cela, il faut pouvoir contrôler le VC-dimension h du système.

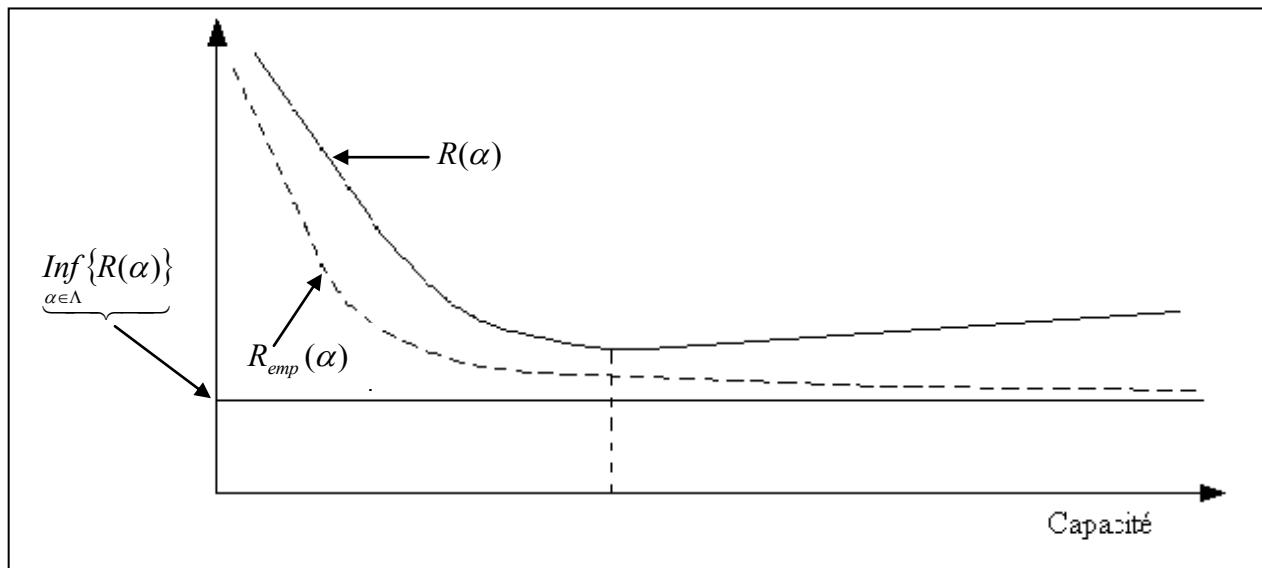


Figure II.3. Consistance du principe de minimisation du risque empirique. Risque ($R(\alpha)$), risque minimal ($\inf_{\alpha \in \Lambda} \{R(\alpha)\}$) et risque empirique ($R_{emp}(\alpha)$) en fonction de la capacité du modèle pour un ensemble d'apprentissage fixe [27].

Dans la pratique, le rapport entre le nombre d'exemples et la *capacité* de la machine d'apprentissage, qui caractérise sa complexité, est de grande importance. De façon similaire nous pouvons dire que plus cette capacité est grande :

- i. plus le risque fonctionnel et le risque empirique (respectivement $R(\alpha)$ et $R_{emp}(\alpha)$) sont faibles.
- ii. mais plus le nombre de données d'apprentissage doit être grand pour assurer que $R(\alpha)$ converge vers $\inf_{\alpha \in \Lambda} \{R(\alpha)\}$.

Quand la taille de l'ensemble d'apprentissage est fixe, il existe une capacité optimale, au-dessus de laquelle le risque augmente même si le risque empirique continue à décroître (figure II.3).

I.4. Minimisation du risque structurel

Afin de contrôler la VC-dimension h , Vapnik propose d'appliquer un nouveau principe inductif qui est le principe de Minimisation du risque structurel (**MRS**). Ce principe est basé sur la minimisation conjointe des deux causes d'erreur : le risque empirique et l'intervalle de confiance.

Définition

Soit $F = \{f(x, \alpha), \alpha \in \Lambda\}$ une famille de fonctions.

Une structure sur Ω est une suite de sous-ensembles emboîtés $F_i = \{f^i(x, \alpha), \alpha \in \Lambda_i\}$:

$$F_1 \subset F_2 \subset \dots \subset F_i \subset \dots$$

dont les VC-dimensions h_i sont finies. Ces VC-dimensions forment alors une suite croissante.

$$h_1 \leq h_2 \leq \dots \leq h_i \leq \dots$$

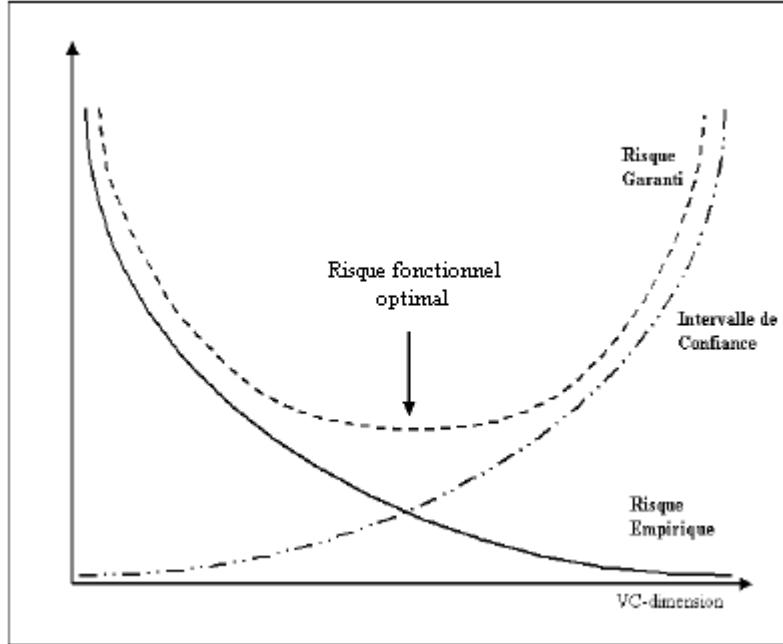


Figure II.4. Comportement du risque empirique, de l'intervalle de confiance et du risque garanti en fonction de la VC-dimension [4].

En pratique, lorsque la VC-dimension augmente, le risque empirique décroît tandis que l'intervalle de confiance croît (un classifieur à large capacité a un faible risque empirique c'est-à-dire qu'il apprend bien mais généralise mal).

L'objectif du principe de la minimisation du risque structurel consiste à chercher un compromis entre la qualité de l'approximation sur l'échantillon considéré et la complexité de la fonction qui réalise l'approximation. Ceci est illustré par la figure II.4 qui montre le comportement du risque empirique, l'intervalle de confiance et le risque garanti en fonction de la VC-dimension. On remarque que le minimum du risque garanti est atteint pour une valeur optimale de h [28].

Cependant, l'implémentation de ce principe n'est pas facile. Le calcul de la VC-dimension de F_i peut être difficile et, même s'il est possible, minimiser le risque garanti n'est pas toujours immédiat. Dans plusieurs de cas, on doit minimiser au préalable le risque empirique pour chaque sous-ensemble F_i , puis choisir le sous-ensemble qui minimise le risque garanti.

La méthode des SVMs réalise la minimisation simultanée du terme de confiance N/h en fonction de la VC-dimension et du risque empirique d'apprentissage.

II. MACHINES A VECTEURS DE SUPPORT (SVMs)

Pour la reconnaissance des formes (cas de deux classes d'échantillons), les SVM recherche une surface de décision optimale, déterminée par certains points de l'ensemble d'apprentissage appelés *vecteurs support*, en projetant les données d'entrée non-linéairement séparables dans un espace de plus grande dimension appelé *espace de caractéristique* (figure II.5). Cette surface, qui est dans l'espace des caractéristiques, peut être considérée comme un hyperplan optimal de décision. Elle est obtenue par la résolution d'un problème de programmation quadratique dépendant des paramètres de régularisation.

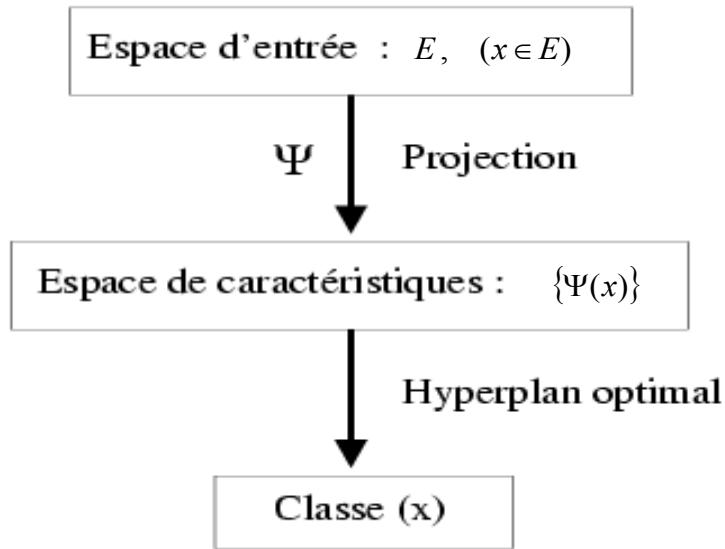


Figure II.5. Principe des SVMs [27].

On considère un ensemble D de N données d'apprentissage séparables en deux classes distinctes (C_1, C_2), tel que :

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} = \{(x_i, y_i)\}_{i=1}^N, \text{ où } y_i \in \{-1, +1\} \text{ et } x_i \in R^n \text{ les échantillons.}$$

(II.3)

$$C_1 = \{x_i, \text{ avec } y_i = +1\}, \text{ et } C_2 = \{x_i, \text{ avec } y_i = -1\}$$

Chaque exemple x_i est caractérisé par un ensemble de n variables descriptives $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ et y_i définit la classe d'un exemple donné x_i .

L'objectif des SVMs est de trouver un hyperplan permettant de séparer les données d'apprentissage de sorte que tous les points d'une même classe soient d'un même côté de l'hyperplan. Deux cas se présentent : cas des données linéairement séparables, et cas des données non linéairement séparables.

II.1. Cas de données linéairement séparables

Pour un problème de classification de données linéairement séparables, l'objectif est de séparer les deux classes de données par une fonction qui est déterminée à partir des échantillons disponibles. Considérons les exemples de la figure II.6 (a), où il existe plusieurs classificateurs linéaires possibles qui peuvent séparer les données, mais il existe qu'un seul (voir la figure II.6 (b)) qui maximise la marge (représente la plus petite distance entre les différentes données des deux classes et l'hyperplan). Ce classifieur linéaire est connu sous le nom Hyperplan Optimale (HO). Intuitivement, Nous escomptons à bien généraliser cette limite par rapport aux autres limites possibles comme ci montré en figure II.6 (a).

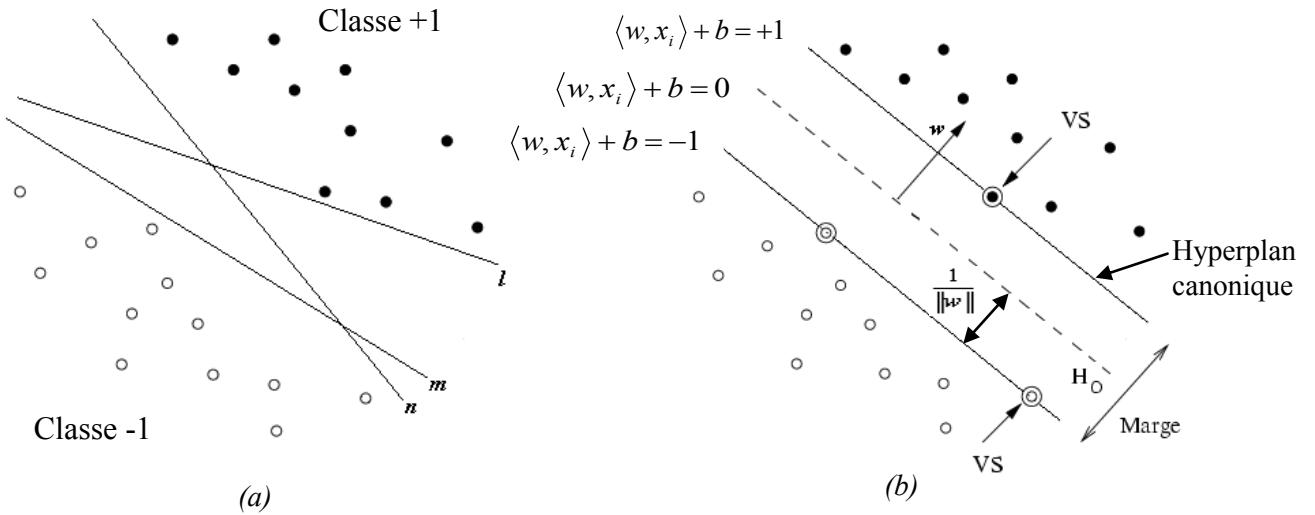


Figure II.6. Classification entre deux classes en utilisant les hyperplans : (a). les hyperplans arbitraires l , m et n ; (b). l'hyperplan de séparation optimale avec une grande marge identifiée par les hyperplans canoniques, traversant les vecteurs du support [2].

Soit $H : \langle w, b \rangle + b$ un hyperplan quelconque de la figure III.6. (a) et satisfait les conditions suivantes :

$$\begin{cases} \langle w, x_i \rangle + b \geq +1 & \text{si } y_i = +1 \\ \langle w, x_i \rangle + b \leq -1 & \text{si } y_i = -1 \end{cases}$$

Ce qui est équivalent à :

$$y_i \cdot [\langle w, x_i \rangle + b] \geq 1 \quad \text{pour } i = 1, \dots, N \quad (\text{II.4})$$

La distance $d(w, b; x)$ entre un point x et l'hyperplan H est donné par :

$$d(w, b; x) = \frac{|\langle w, x \rangle + b|}{\|w\|} \quad (\text{II.5})$$

L'hyperplan optimal est donné par la maximisation de la marge $M(w, b)$, sous les contraintes de l'équation (II.4). La marge $M(w, b)$ s'écrit sous la forme suivante :

$$\begin{aligned} M(w, b) &= \min_{x_i | y_i = +1} d(w, b; x_i) + \min_{x_i | y_i = -1} d(w, b; x_i) \\ &= \min_{x_i | y_i = +1} \left\{ \frac{|\langle w, x_i \rangle + b|}{\|w\|} \right\} + \min_{x_i | y_i = -1} \left\{ \frac{|\langle w, x_i \rangle + b|}{\|w\|} \right\} \\ &= \frac{1}{\|w\|} \left[\min_{x_i | y_i = +1} |\langle w, x_i \rangle + b| + \min_{x_i | y_i = -1} |\langle w, x_i \rangle + b| \right] \\ &= \frac{1}{\|w\|} [1 + 1] \\ &= \frac{2}{\|w\|} \end{aligned}$$

Plus la marge est grande, plus l'erreur attendue est faible. Maximiser la marge $M(w, b)$, revient à minimiser le carré de la norme ($\|w\|^2 = \langle w, w \rangle$) du vecteur w sous les contraintes de l'équation (II.4).

Ainsi, l'hyperplan optimal qui vérifie les conditions de l'équations (II.4) est celui qui minimise la fonction définie par :

$$\Phi(w) = \frac{1}{2} \|w\|^2 \quad (\text{II.6})$$

La solution du problème d'optimisation de (II.6) sous les contraintes de (II.4) est donnée par le point-selle $\bar{w}, \bar{b}, \bar{\alpha}$ du Lagrangien :

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i \{y_i [\langle w, x_i \rangle + b] - 1\} \quad (\text{II.7})$$

où $\alpha = (\alpha_1, \dots, \alpha_N)$.

Le Lagrangien doit être minimal par rapport à w et b et maximal par rapport à $\alpha \geq 0$.

➤ $L(w, b, \alpha)$ est minimal par rapport à b :

$$\frac{\partial L(w, b, \alpha)}{\partial b} = 0 \iff \sum_{i=1}^N \alpha_i y_i = 0 \quad (\text{II.8})$$

➤ $L(w, b, \alpha)$ est minimal par rapport à w :

$$\frac{\partial L(w, b, \alpha)}{\partial w} = 0 \iff w = \sum_{i=1}^N \alpha_i x_i y_i \quad (\text{II.9})$$

Ces contraintes d'égalité permettent d'exprimer la forme duale Lagrangien :

$$F(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \quad (\text{II.10})$$

➤ $L(w, b, \alpha)$ est maximal par rapport à $\alpha \geq 0$:

Pour trouver le point-selle, il suffit alors de maximiser (et non plus de minimiser) $F(\alpha)$ dans le quadrant positif $\alpha_i \geq 0$, pour tout $i \in \{1, \dots, N\}$ sous la contrainte :

$$\sum_{i=1}^N \bar{\alpha}_i y_i = 0 \quad (\text{II.11})$$

Ce point-selle vérifie les conditions :

$$\bar{\alpha}_i [y_i (\bar{w}, \bar{x}_i) + \bar{b}] = 0, \quad \text{pour tout } i \in \{1, \dots, N\} \quad (\text{II.12})$$

Définition

Les vecteurs support **VS** sont les vecteurs \bar{x}_i pour lesquels l'égalité $y_i (\bar{w}, \bar{x}_i) + \bar{b} = 1$ est vérifiée. Concrètement, ce sont les vecteurs les plus proches de l'hyperplan optimal. Pour tous les autres exemples (échantillons) qui vérifient l'équation (II.12), nous avons donc : $\bar{\alpha}_i = 0$. Ce qui est équivalent à :

$$VS = \{\bar{x}_i \mid \bar{\alpha}_i > 0\} \quad \text{pour } i = 1, \dots, N$$

Ainsi, on peut facilement calculer \bar{w} :

$$\bar{w} = \sum_{x_i \in VS} \bar{\alpha}_i y_i \bar{x}_i \quad (\text{II.13})$$

La résolution de l'équation (II.12) nous donne les solutions suivantes :

$$\bar{\alpha}_i = 0 \quad \text{ou} \quad y_i [\bar{w}, \bar{x}_i] + \bar{b} = 1 \quad \text{avec } i = 1, \dots, N \quad (\text{II.14})$$

Pour calculer le biais \bar{b} nous avons utilisé l'équation :

$$y_i [\langle \bar{w}, \bar{x}_i \rangle + \bar{b}] = 1 \quad \text{avec} \quad \bar{\alpha}_i > 0$$

Pour un vecteur support de la classe +1 :

$$\langle \bar{w}, \bar{v}_{S_{(classe+1)}} \rangle + \bar{b} = 1$$

Pour un vecteur support de la classe -1 :

$$\langle \bar{w}, \bar{v}_{S_{(classe-1)}} \rangle + \bar{b} = -1$$

D'après ces deux équations nous obtenons la valeur du biais \bar{b} :

$$\bar{b} = -\frac{1}{2} [\langle \bar{w}, \bar{v}_{S_{(classe+1)}} \rangle + \langle \bar{w}, \bar{v}_{S_{(classe-1)}} \rangle] \quad (\text{II.15})$$

Ainsi, la fonction de classification $classe(x)$ s'écrit sous la forme suivante :

$$\begin{aligned} classe(x) &= sign(\langle \bar{w}, \bar{x} \rangle + \bar{b}) \\ classe(x) &= sign[\left\langle \sum_{x_i \in VS} \bar{\alpha}_i \bar{y}_i \bar{x}_i, \bar{x} \right\rangle + \bar{b}] \\ classe(x) &= sign[\sum_{x_i \in VS} \bar{\alpha}_i \bar{y}_i \langle \bar{x}_i, \bar{x} \rangle + \bar{b}] \end{aligned} \quad (\text{II.16})$$

Si la fonction $classe(x)$ est négative alors x appartient à la classe -1 sinon x appartient à la classe +1.

II.2. Cas de données non-linéairement séparables

Dans la plupart des problèmes réels, les données, comme indiqué par la figure II.7 ne sont pas toujours linéairement séparables. Il est nécessaire de contourner ce problème. Les SVMs ont été généralisées grâce à deux outils : la marge souple et les fonctions noyaux.

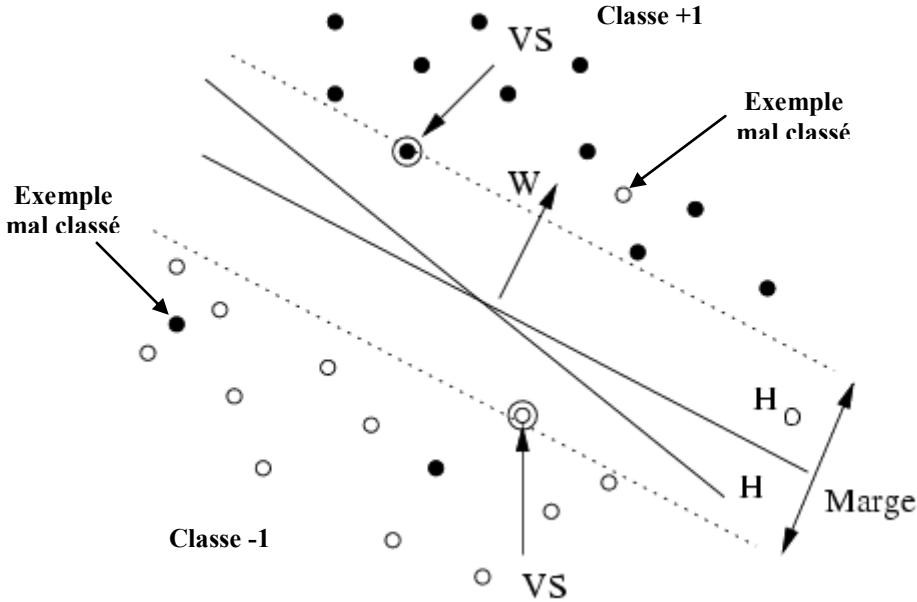


Figure II.7. Hyperplans séparateurs dans le cas de données non-linéairement séparables : H est un hyperplan quelconque, H_0 est l'hyperplan optimal, VS : sont les vecteurs support [27].

II.2.1. Marge souple (Soft Margin)

Le principe de la marge souple est d'autoriser des erreurs de classification. Le nouveau problème de séparation optimale est reformulé comme suit : l'hyperplan optimal séparant les deux classes est celui qui sépare les données avec le minimum d'erreurs, et satisfait donc les conditions suivantes :

1. La distance entre les vecteurs bien classés et l'hyperplan doit être maximale.
2. La distance entre les vecteurs mal classés et l'hyperplan doit être minimale.

Pour généraliser l'hyperplan optimal au cas de données non-linéairement séparables, des variables de pénalité (variables d'écart) ξ_i sont introduites dans [29]. Par conséquent, les contraintes (II.4) sont modifiées comme suit :

$$\begin{cases} \langle w, x_i \rangle + b \geq 1 - \xi_i & \text{si } y_i = +1 \\ \langle w, x_i \rangle + b \leq \xi_i - 1 & \text{si } y_i = -1 \end{cases}$$

Ce qui est équivalent à :

$$y_i \cdot [\langle w, x_i \rangle + b] \geq 1 - \xi_i \quad \text{pour } i = 1, \dots, N \quad (\text{II.17})$$

Ainsi, au lieu de chercher le vecteur de poids \bar{w} qui minimise le carré de la norme ($\langle w, w \rangle$) comme dans le cas de données linéairement séparables, on cherche à minimiser la fonction suivante :

$$\Phi(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (\text{II.18})$$

(où $\xi = (\xi_1, \dots, \xi_N)$ et C ($C \geq 0$) est une constante choisie par l'utilisateur) sous les contraintes de (II.17).

Le paramètre C est interprété comme une tolérance au bruit de classification. Pour des grandes valeurs de C , seules de très faibles valeurs de ξ sont autorisées et, par conséquent, le nombre de points mal classés sera très faible. Cependant, si C est petit, ξ peut devenir assez grand et on autorise alors bien plus d'erreurs de classification.

La solution $\bar{\alpha}$ s'obtient de la même manière, par la maximisation de Lagrangien dual.

$$\max_{\alpha} L(w, b, \alpha)$$

Sous des contraintes, on peut différencier le cas d'une classification de données linéairement séparables :

$$\sum_{i=1}^N \bar{\alpha}_i y_i = 0, \quad 0 \leq \bar{\alpha}_i \leq C \quad \text{pour } i = 1, \dots, N$$

Ainsi, nous obtenons l'équation de l'hyperplan optimal au sens des nouvelles contraintes tenant compte des ξ_i . Les vecteurs support sont toujours les vecteurs échantillons les plus proches de cet hyperplan.

Le calcul de la normale \bar{w} , du biais \bar{b} et de la fonction de classification $classe(x)$ reste exactement le même que pour le cas des données linéairement séparables.

II.2.2. Fonctions noyaux

Nous venons de voir que l'idée fondamentale des SVM est de rendre possible la projection des vecteurs d'entrée dans un espace de représentation interne de plus grande dimension afin que l'hyperplan optimal construit sur cet espace soit général, indépendamment de la dimension de ce dernier.

Soit Ψ une fonction non linéaire qui transforme l'espace d'entrée E à un espace interne H de plus grande dimension.

$$\begin{aligned}\Psi : E &\rightarrow H \\ x &\mapsto h\end{aligned}$$

où $x \in E$ et $h \in H$.

Ainsi, l'ensemble d'exemples D' sera transformé sous la forme suivante :

$$D' = \{(\Psi(x_1), y_1), (\Psi(x_2), y_2), \dots, (\Psi(x_N), y_N)\} = \{(\Psi(x_i), y_i)\}_{i=1}^N$$

où $y_i \in \{-1, +1\}$, $x_i \in R^n$, et $\Psi(x_i) \in R^p$ (les entrées transformées par la projection Ψ).

Notons que la dimension p de l'espace des caractéristiques (espace de représentation interne) est supérieure ou égale à la dimension n de l'espace de départ (espace d'entrée).

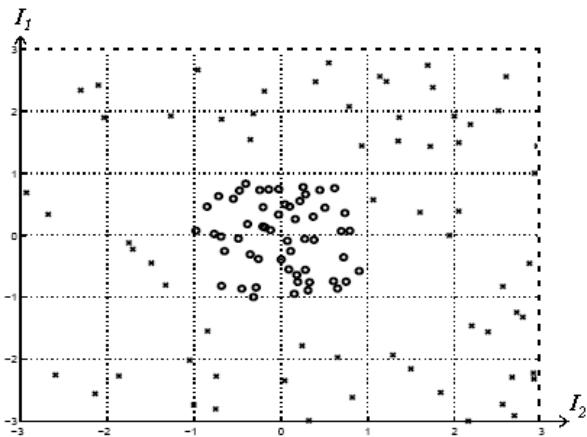
Exemple

Pour avoir une idée plus claire sur les SVM, nous donnons un exemple inspiré du travail de Schölkopf [30] qui met en pratique le principe des SVM. Dans cet exemple, les données non-

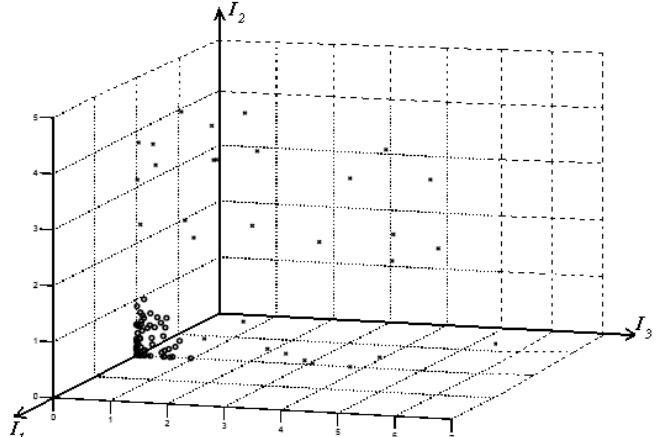
linéairement séparables dans R^2 deviennent linéairement séparables dans R^3 grâce à la transformation Ψ définie par :

$$\begin{aligned}\psi : R^2 &\rightarrow R^3 \\ (x_{i1}, x_{i2}) &\mapsto (x_{i1}^2, \sqrt{2} x_{i1} x_{i2}, x_{i2}^2)\end{aligned}$$

La figure II.8 représente une simulation de cette transformation qui est réalisée avec Matlab. Les données de la figure II.8. (a) ont été tirées aléatoirement dans R^2 et la figure II.8. (b) représente l'image de ces données dans R^3 suivant la transformation ψ .



a : exemples tirés aléatoirement dans R^2 appartenant à deux classes non-linéairement



b : l'image des exemples de la figure II.8.(a) dans R^3 en utilisant la transformation Ψ .

Figure II.8. Exemple montrant l'efficacité d'une transformation dans un espace de plus grande dimension pour faciliter la classification [31].

Pour construire l'hyperplan optimal dans l'espace des représentations internes sans considérer explicitement cet espace, il suffit de savoir calculer les produits scalaires entre les vecteurs support et les vecteurs de l'espace de représentation interne.

Ainsi, il suffit de prendre toutes les équations énoncées dans les paragraphes précédents, pour la recherche de l'hyperplan optimal, et de remplacer les produits scalaires des vecteurs d'entrée $\langle\langle x_i, x_j \rangle\rangle$ par leurs images dans l'espace de représentation interne $\langle\langle \Psi(x_i), \Psi(x_j) \rangle\rangle$ ou tout simplement par des fonctions noyaux $K(x_i, x_j)$:

$$K(x_i, x_j) = \langle \Psi(x_i), \Psi(x_j) \rangle$$

En résumé, pour tout problème de classification supervisée, la solution est donnée par la maximisation de la forme quadratique suivante (Lagrangien dual) :

$$F(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (\text{II.19})$$

sous les contraintes :

$$\sum_{i=1}^N \alpha_i y_i = 0 ; 0 \leq \alpha_i \leq C \quad \text{pour tout } i = 1, \dots, N$$

Après avoir calculé les $\bar{\alpha}_i$, on peut définir la fonction de classification par la formule suivante :

$$\text{classe}(x) = \text{sign} \left[\sum_{x_i \in VS} \bar{\alpha}_i y_i K(x_i, x) + \bar{b} \right] \quad (\text{II.20})$$

Si la fonction $\text{classe}(x)$ est négative alors x appartient à la classe -1 sinon x appartient à la classe +1.

Les VS sont les vecteurs x_i associés aux $\alpha_i > 0$. Géométriquement, ce sont les vecteurs les plus proches de l'hyperplan optimal qui séparent les deux classes dans l'espace des représentations internes.

Dans ce qui suit, nous présentons les noyaux les plus utilisés dans la classification par la technique des SVMs :

1. Le noyau linéaire (produit scalaire) :

$$K(x, x_i) = \langle x, x_i \rangle$$

2. Le noyau polynomial :

$$K(x, x_i) = (\langle x, x_i \rangle + 1)^d$$

où d est le degré du polynôme à déterminer par l'utilisateur.

3. Le noyau RBF (Radial Basis Function) :

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right)$$

où σ est un paramètre de régulation qui est à déterminer de façon empirique. Ce noyau est aussi désigné par noyau Gaussien.

Reprenons l'exemple qu'on a évoqué précédemment. Donc on a la transformation Ψ telle que :

$$\begin{aligned} \psi : \quad & R^2 \rightarrow R^3 \\ & (x_{i1}, x_{i2}) \mapsto (x_{i1}^2, \sqrt{2} x_{i1} x_{i2}, x_{i2}^2) \end{aligned}$$

D'où :

$$\begin{aligned} K(x_i, x_j) &= \langle \Psi(x_i), \Psi(x_j) \rangle \\ &= (x_{i1}^2, \sqrt{2} x_{i1} x_{i2}, x_{i2}^2) \begin{pmatrix} x_{j1}^2 \\ \sqrt{2} x_{j1} x_{j2} \\ x_{j2}^2 \end{pmatrix} \\ &= (x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{i2} x_{j1} x_{j2} + x_{i2}^2 x_{j2}^2) \\ &= (x_{i1} x_{i2} + x_{j1} x_{j2})^2 \\ &= (x_{i1}, x_{i2}) \begin{pmatrix} x_{j1} \\ x_{j2} \end{pmatrix} = \langle x_i, x_j \rangle^2 \end{aligned}$$

Comme on peut le constater le noyau correspondant à la transformation Ψ de notre exemple proposé au début de cette section n'est autre qu'un noyau polynomial de degré 2.

III. EXTENSION DES SVMs BINAIRES AU CAS MULTI CLASSES

Initialement les SVMs ont été conçues principalement pour la classification binaire [29]. La question de leur extension au problème de la classification multi-classes reste un thème de recherche. Actuellement, il y a deux types d'approches pour les SVMs multi-classes : une par considération de toutes les classes dans une seule formule d'optimisation, l'autre par une décomposition en combinant plusieurs classifieurs binaires [32] [33] [34].

Le problème de multi-classes consiste à construire une fonction de décision, étant donnée la base D de N données d'apprentissage séparables en s classes (C_1, C_2, \dots, C_s) , tel que :

$$D = \{(x_i, y_i), x_i \in R^n; i=1, \dots, N, y_i \in \{1, \dots, s\}\} \quad (\text{II.21})$$

III.1. Résolution du problème SVM multi-classes par une seule optimisation [3], [35]

Dans cette approche, il ne s'agit plus de trouver une seule séparatrice entre deux classes mais il faudra que l'on soit en mesure de classer les exemples en plusieurs classes, ce qui revient à construire s règles de deux classes, où la $j^{\text{ème}}$ fonction $\langle w_j, x \rangle + b_j$ sépare les vecteurs d'apprentissage qui correspondent à la classe j de tous les vecteurs des autres classes. Nous avons alors s fonctions de décision mais toutes s'obtiennent par la résolution d'un seul problème. Ainsi, la généralisation du SVM binaire [4] au cas multi-classes revient à résoudre le problème de programmation quadratique suivant :

$$\left\{ \begin{array}{l} \min_{w_j, b_j, \xi} \left(\frac{1}{2} \sum_{j=1}^s \|w_j\|^2 + C \sum_{i=1}^N \sum_{j \neq y_i} \xi_i^j \right) \\ \text{sous les contraintes :} \\ \langle w_{y_i}, x_i \rangle + b_{y_i} - \left(\langle w_j, x_i \rangle + b_j \right) \geq 1 - \xi_i^j \quad \text{avec} \quad \xi_i^j \geq 0, i=1, \dots, N \text{ et } j \in \{1, \dots, s\} \end{array} \right. \quad (\text{II.22})$$

Ainsi, la fonction de classification $\text{classe}(x)$ s'écrit sous la forme suivante :

$$\text{classe}(x) = \arg \max_{1 \leq j \leq s} \text{classe}_j(x) \quad \text{où} \quad \text{classe}_j(x) = \langle w_j, x \rangle + b_j, j \in \{1, \dots, s\} \quad (\text{II.23})$$

Comme précédemment, il suffit de remplacer le produit scalaire $\langle w_j, x \rangle$ par son image dans l'espace de représentation interne dans (II.23) en utilisant la fonction de projection Ψ ou par la fonction noyau K :

$$\langle w_j, x \rangle \Rightarrow \langle \Psi(w_j), \Psi(x) \rangle = K(w_j, x)$$

Donc, un nouvel individu x aura comme classe, la classe y_i telle que :

$$classe_{y_i}(x) \geq classe_j(x), \quad \forall j \neq y_i, \text{ et } j \in \{1, \dots, s\} \quad (\text{II.24})$$

III.2. Résolution du problème multi-classes par des SVMs binaires

Le deuxième type de SVM multi-classes est basé sur la combinaison de plusieurs classifieurs. Nous trouvons trois méthodes : un-contre-tous, un-contre-un, et la méthode DAGSVM (Directed Acyclic Graph Support Vector Machines).

III.2.1. Un contre tous (OAA pour *One Against All*)

La première méthode qui a été implémentée est l'approche un-contre-tous [36]. Cette méthode construit s modèles SVM (un SVM pour chaque classe). Le $i^{\text{ème}}$ SVM est appris avec tous les exemples. La $i^{\text{ème}}$ classe est indexée avec des étiquettes positives (+1) et toutes les autres avec des étiquettes négatives (-1). Ce $i^{\text{ème}}$ classifieur construit un hyperplan entre la $i^{\text{ème}}$ classe et les $s - 1$ autres classes.

La partition linéaire de l'espace, dans le cas de trois classes par exemple, est montrée en figure II.9. On remarque les régions de conflit (deux classes sont choisies simultanément) ou d'ambiguïté (aucune classe n'est choisie). Deux approches permettent de résoudre ce problème. La première considère une classe supplémentaire dite de rejet (ou de non décision) groupant les régions de conflit ou d'ambiguïté. Selon l'application, des critères sont imposés sur la définition des frontières de cette nouvelle classe. La seconde approche est dite un contre un.

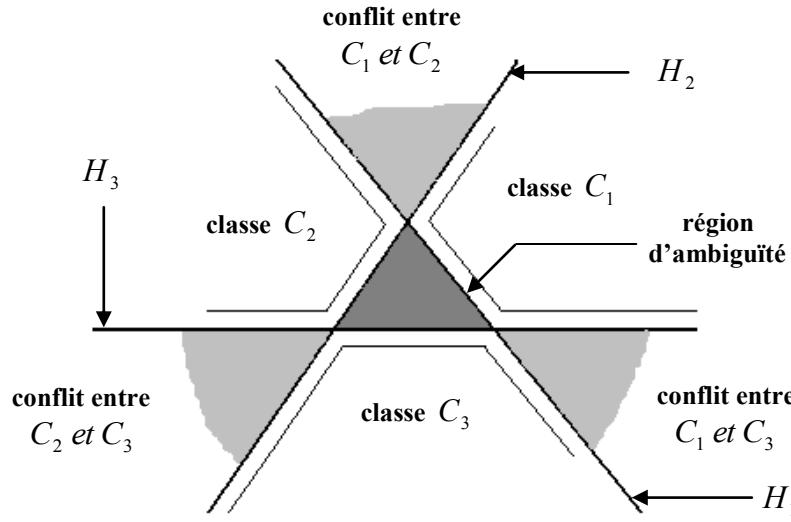


Figure II.9. Exemple de l'approche OAA qui utilise 3 SVMs binaires pour un pour un problème de classification de 3 classes C_1, C_2 et C_3 qui sont séparées respectivement par trois hyperplans H_1, H_2 et H_3 .

III.2.2. Un contre un (OAO pour *One Against One*)

La deuxième méthode, appelée un-contre-un, a été introduite par [37]. La première utilisation de cette stratégie avec les SVM a été réalisée par [38] et [39]. Cette méthode construit $s(s - 1)/2$ classifieurs où chacun est appris sur les données des deux classes. Pendant la phase du test et après la construction de tous les classifieurs, on utilise la stratégie de vote proposée par [39]. Si le signe de la fonction de décision des deux classes i et j indique que x appartient à la $i^{\text{ème}}$ classe, et le vote pour la $i^{\text{ème}}$ est incrémenté par +1. Alors que, le vote pour la $j^{\text{ème}}$ classe est décrémenté de 1. À la fin, on prédit que x appartient à la classe qui a le vote le plus grand. Par conséquent, cette stratégie présente l'avantage de dessiner moins de zones d'ambiguïté, mais elle a l'inconvénient de demander le calcul de $s(s - 1)/2$ jeux de paramètres au lieu de s .

Les problèmes d'ambiguïté peuvent être résolus par un système ultérieur de décision sur l'ensemble des classes, prenant en compte des paramètres supplémentaires, par exemple la distance aux hyperplans appris et/ou les probabilités *a priori* des classes. On peut aussi regrouper provisoirement les classes par des méthodes de classification hiérarchiques, et réaliser un processus de décision arborescent sur cette base. Ceci est en rapport avec les méthodes de classification hiérarchiques, et réaliser un processus de décision arborescent sur cette base. Ceci est en rapport avec les arbres de décision.

III.2.3. DAGSVM (Directed Acyclic Graph Support Vector Machines)

La troisième méthode, DAGSVM (Directed Acyclic Graph Support Vector Machines) a été proposée par [40]. Sa phase d'apprentissage est la même que celle de la méthode un-contre-un, elle utilise la résolution proposée par les $s(s - 1)/2$ classificateurs binaires. Cependant, dans la phase du test, elle utilise un graphe acyclique dirigé binaire avec une seule racine. Ce graphe possède $s(s - 1)/2$ nœuds internes et s sorties. Chaque nœud est un SVM binaire pour des $i^{\text{ème}}$ et $j^{\text{ème}}$ classes. Pour un exemple x , en commençant par le nœud racine, la fonction de décision binaire est évaluée. Ensuite, elle se déplace à droite ou à gauche selon sa valeur de sortie. Enfin, on obtient un chemin de parcours de la racine jusqu'au nœud de sortie qui indique la classe recherchée. La figure II.10 donne un exemple de décision à quatre classes.

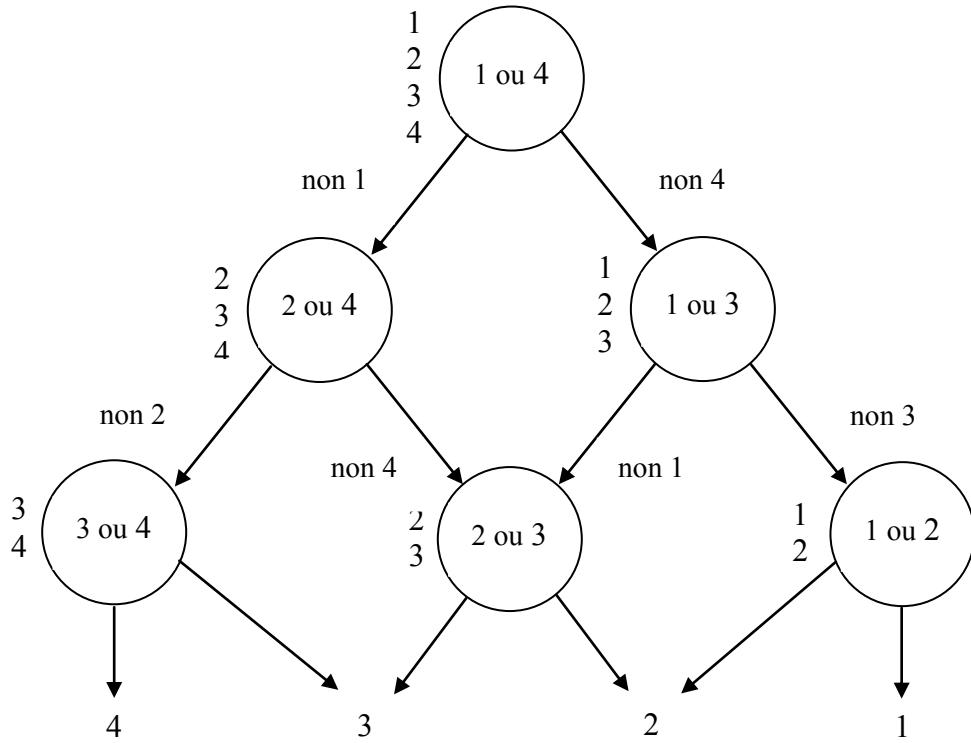


Figure II.10. La décision DAG pour la recherche de la meilleure classe parmi 4 classes, la liste d'états équivalents pour chaque nœud est indiquée à côté du nœud [27].

Cette théorie a été mise en œuvre pour l'amélioration de la qualité d'identification des individus, à travers une reconnaissance de visages. Concernant notre travail, nous nous intéressons à présenter un modèle de reconnaissance appliqué au cas de deux classes de visages

hommes/femmes. Ce modèle a été proposé par Schölkopf et Smola dans [41], dont son principe est basé sur la construction d'un hyperplan séparateur pour l'établissement d'un classifieur binaire de décision.

IV. CONCLUSION

La reconnaissance de visages dans une base de données est un processus de vérification des visages, il s'agit donc d'une classification de ces visages. Parmi les méthodes de classification, nous avons pensé à proposer l'utilisation des SVMs pour la reconnaissance de visages.

La méthode des SVMs est l'une des rares méthodes d'apprentissage qui soient entièrement issues de considérations théoriques (minimisation du risque empirique et minimisation du risque structurel). En effet, l'approfondissement de leur justification théorique a conduit à un règlement automatique du compromis entre la complexité de la classe des hypothèses, la distribution des données d'apprentissage et la nécessaire fidélité aux données. Pour ces raisons d'ordre conceptuel, on assiste actuellement à une grande utilisation dans plusieurs domaines de la méthode des SVMs.

De notre part, et dans le cadre de notre travail, cette théorie et la méthode des *EigenFaces* ont été mises en œuvre pour la construction d'un classifieur homme/femme, en utilisant deux modèles de reconnaissance appliqués au cas de deux classes. Ces deux modèles seront présentés dans le chapitre suivant de ce mini-projet.

CHAPITRE III

**EigenFaces ET SVMs APPLIQUEES A LA
RECONNAISSANCE DE VISAGES
HOMMES/FEMMES**

INTRODUCTION

Les *EigenFaces* sont bien adaptés aux problèmes de reconnaissance supervisée de visages hommes/femmes. Dans ces problèmes, on cherche à classifier les visages hommes et les visages femmes d'une base de données selon les valeurs de leurs coefficients de projection dans l'espace des visages propres (*face space*).

Le but de notre étude est de construire un classifieur de type (*SVMs*) capable d'améliorer les performances de la reconnaissance qu'on peut avoir avec un classifieur de type (*EigenFaces*) et donc, de faire la distinction entre le visage d'un homme et celui d'une femme avec un taux d'erreur minimal. Cependant, la reconnaissance dans le cas de plusieurs classes est aussi envisageable.

Comme n'importe quelle méthode de reconnaissance supervisée, les *EigenFaces* et les *SVMs* se déroulent en deux phases distinctes :

1. Une phase d'apprentissage dans laquelle on utilise la base de données d'entraînement. Ces données sont sélectionnées au préalable à partir d'une connaissance *a priori* sur le nombre de classes et le nombre d'échantillons correspondant à chaque classe.
2. Une phase de reconnaissance de tous les visages de la base de tests en utilisant :
 - **Cas des EigenFaces** : les vecteurs représentants de chaque classe établis durant la phase d'apprentissage ainsi le vecteur représentant le visage du test.
 - **Cas des SVMs** : le paramètre *offset* (le biais).

I. MÉTHODOLOGIE SUIVIE

I.1. Méthode des *EigenFaces*

La démarche méthodologique que nous avons suivie pour la construction d'un classifieur homme/femme de type (*EigenFaces*) est schématisée par l'organigramme de la figure III.1. Cet organigramme est constitué des différentes étapes suivantes :

I.1.1. Phase d'apprentissage

Les étapes de la phase d'apprentissage des *EigenFaces* sont données comme suit :

1) Calcul de visage moyen

Le visage moyen ψ est calculé à partir des N images d'entraînement I_i (vecteurs lignes de dimensions $L \times H$) en utilisant la formule suivante :

$$\Psi = \frac{1}{N} \sum_{i=1}^N I_i \quad (\text{III.1})$$

2) Calcul des valeurs de différences

Pour éliminer les ressemblances qui existent entre les visages de la base d'apprentissage, on calcule les valeurs de différences Φ_i associées à chacune des images :

$$\Phi_i = I_i - \Psi \quad (\text{III.2})$$

3) Calcul de la matrice de covariance

Pour des considérations pratiques (l'espace mémoire utilisé), le calcul de la matrice de covariance est effectué en utilisant une approche inter-images :

$$L = A A^T \quad (\text{III.3})$$

où chacune des lignes de A représente un vecteur de différences (vecteur ligne), soit

$$A = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_N \end{bmatrix}.$$

Notons que la matrice de covariance L dépendante du nombre d'images dans la banque d'apprentissage (c-à-d : dimension $N \times N$).

4) Calcul des valeurs et vecteurs propres de la matrice de covariance

Une fois que la matrice de covariance L est construite, les valeurs propres $(\lambda_i)_{i=1,\overline{N}}$ et la matrice des vecteurs propres $(v_{ik})_{i,k=1,\overline{N}}$ sont déterminées.

5) Construction de la base de l'espace des visages (*face space*)

Les M premiers visages propres (EF) (c-à-d : associés aux plus fortes valeurs propres) sont calculés en utilisant la formule suivante :

$$\begin{cases} u_i = \sum_{k=1}^N v_{ik} \Phi_k \\ \text{tel que : } i = \overline{1, M} \end{cases} \quad (\text{III.4})$$

Alors, les vecteurs $\{u_i, i = \overline{1, M}\}$ définissent ainsi la base de l'espace des visages.

6) Calcul des coefficients d'appartenance et formation des représentants

Les images originales de la base d'apprentissage sont projetées dans l'espace des visages pour former une suite de coefficients d'appartenance, ce qui donne pour l'image I_i :

$$\begin{cases} \omega_k = u_k I_i^T \\ \text{tel que : } k = \overline{1, M} \end{cases} \quad (\text{III.5})$$

Ces M coefficients forment alors un vecteur représentant l'image I_i :

$$\Omega_i = [\omega_1, \omega_2, \dots, \omega_M] \quad (\text{III.6})$$

7) Détermination de la représentation de chaque classe

Une fois l'apprentissage complété, les différentes représentations de visages femmes (respectivement : de visages hommes) peuvent être regroupées afin de former une classe. Ceci peut être réalisé notamment en calculant une moyenne des différents vecteurs I_i correspondants aux visages femmes (respectivement : visages hommes).

Le vecteur représentant la classe des visages femmes est calculé en utilisant la formule suivante :

$$\Omega_f = \frac{1}{N_f} \sum_{i=1}^{N_f} \Omega_i = [\omega_{f1}, \omega_{f2}, \dots, \omega_{fM}] \quad (\text{III.7})$$

tel que : N_f représente le nombre de visages femmes de la base d'apprentissage.

Le vecteur représentant la classe des visages hommes est calculé en utilisant la formule suivante :

$$\Omega_h = \frac{1}{N_h} \sum_{i=N_f+1}^N \Omega_i = [\omega_{h1}, \omega_{h2}, \dots, \omega_{hM}] \quad (\text{III.8})$$

Avec : N_h est le nombre de visages hommes de la base d'entraînement.

Il est également possible de considérer les représentations séparément et ainsi les utiliser directement avec l'algorithme du K -ppv. C'est d'ailleurs cette approche qui a été retenue dans l'implémentation du système.

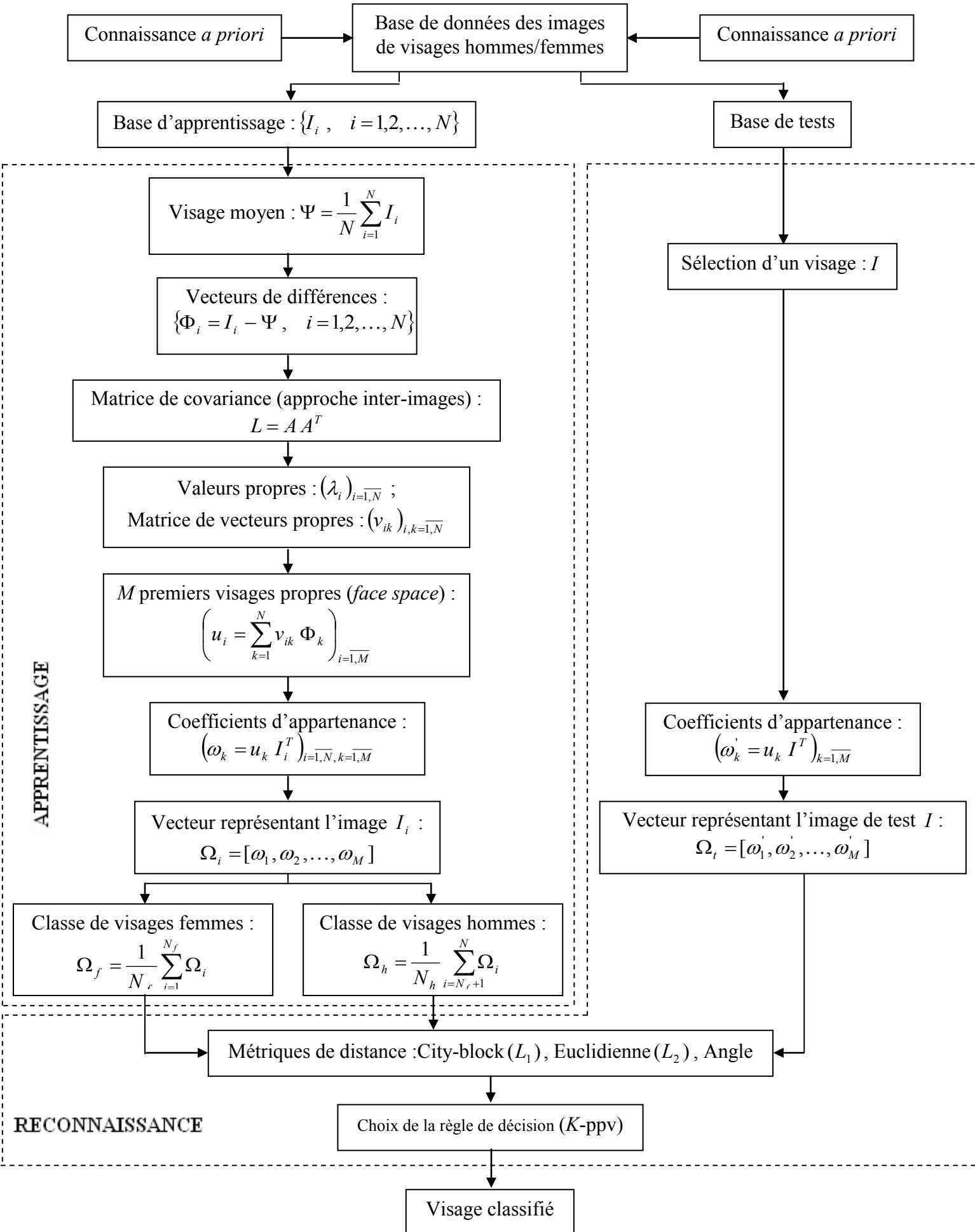


Figure III.1 : Organigramme de la reconnaissance de visages hommes/femmes par les EigenFaces

I.1.2. Phase de reconnaissance

Lorsqu'un visage d'une femme ou d'un homme est présenté au système, la procédure d'identification consiste à :

- 1) Projeter l'image d'entrée I dans l'espace des visages, ce qui engendre un coefficient d'appartenance ω_k^i à un *EigenFace* u_k en utilisant l'équation suivante :

$$\omega_k^i = u_k^T I \quad (\text{III.9})$$

- 2) Les coefficients d'appartenance forment alors un vecteur de représentation Ω_t de taille M :

$$\Omega_t = [\omega_1^i, \omega_2^i, \dots, \omega_M^i] \quad (\text{III.10})$$

- 3) Le vecteur Ω_t est comparé avec ceux obtenus lors de la phase d'apprentissage (c-à-d : Ω_f et Ω_h) à l'aide de l'algorithme K -ppv en appliquant une métrique de distance particulière.

3.1) Calcul des métriques de distance

City-block (L_1) : La distance L_{1f} (respectivement : L_{1h}) entre le vecteur du test et le prototype de visages femmes (respectivement : visages hommes) est calculée en utilisant l'équation III.11 (respectivement : l'équation III.12) :

$$L_{1f} = |\Omega_f - \Omega_t| = \sum_{i=1}^M |\omega_{fi} - \omega_i^i| \quad (\text{III.11})$$

$$L_{1h} = |\Omega_h - \Omega_t| = \sum_{i=1}^M |\omega_{hi} - \omega_i^i| \quad (\text{III.12})$$

Euclidienne (L_2) : La distance L_{2f} (respectivement : L_{2h}) entre le vecteur du test et le prototype de visages femmes (respectivement : visages hommes) est calculée en utilisant l'équation III.13 (respectivement : l'équation III.14) :

$$L_{2f} = \|\Omega_f - \Omega_t\| = \sqrt{\sum_{i=1}^M (\omega_{fi} - \omega_i^i)^2} \quad (\text{III.13})$$

$$L_{2h} = \|\Omega_h - \Omega_t\| = \sqrt{\sum_{i=1}^M (\omega_{hi} - \omega_t^i)^2} \quad (\text{III.14})$$

Angle : L'angle négatif Θ_f (respectivement : Θ_h) entre le vecteur du test et le prototype de visages femmes (respectivement : visages hommes) équivaut à l'équation III.15 (respectivement : à l'équation III.16) :

$$\Theta_f = -\frac{\langle \Omega_f, \Omega_t \rangle}{\|\Omega_f\| \|\Omega_t\|} = -\frac{\sum_{i=1}^M \omega_{fi} \omega_t^i}{\sqrt{\sum_{i=1}^M (\omega_{fi})^2 \times \sum_{i=1}^M (\omega_t^i)^2}} \quad (\text{III.15})$$

$$\Theta_h = -\frac{\langle \Omega_h, \Omega_t \rangle}{\|\Omega_h\| \|\Omega_t\|} = -\frac{\sum_{i=1}^M \omega_{hi} \omega_t^i}{\sqrt{\sum_{i=1}^M (\omega_{hi})^2 \times \sum_{i=1}^M (\omega_t^i)^2}} \quad (\text{III.16})$$

3.2) Prise de décision et résultat final de la reconnaissance

La décision de l'attribution d'un visage à la classe des visages femmes (respectivement : la classe des visages hommes) est prise selon l'une des règles suivantes :

- Minimisation de la distance *City-block*;
- Minimisation de la distance *Euclidienne*.

I.2. Méthode des SVMs [41]

La démarche méthodologique que nous avons suivie pour la construction d'un classifieur homme/femme de type (*SVMs*) est schématisée par l'organigramme de la figure III.3. Cet organigramme est constitué des deux phases suivantes :

I.2.1. Phase d'apprentissage

Dans cette section, nous allons décrire un algorithme d'apprentissage de reconnaissance de visages hommes/femmes par la méthode des SVMs, c'est l'un des algorithmes les plus simples possibles. Nous supposons que nos données sont considérées dans l'espace de produit scalaire H . Nous pouvons mesurer des distances dans cet espace. L'idée de base de l'algorithme est d'affecter un visage de la base du test à la classe dont la moyenne est plus proche.

Donc, Nous commençons en calculant les moyennes des deux classes dans l'espace des caractéristiques;

$$C_+ = \frac{1}{N_f} \sum_{\{i|y_i=+1\}} I_i, \quad (\text{III.17})$$

$$C_- = \frac{1}{N_h} \sum_{\{i|y_i=-1\}} I_i, \quad (\text{III.18})$$

où N_h et N_f sont les nombres d'exemples avec des étiquettes positives et négatives, respectivement. Nous supposons que les deux classes sont non vides. Donc N_h et $N_f > 0$. Nous affectons un nouveau visage I à la classe avec une moyenne plus proche (Figure III.1). Cette construction géométrique peut être formulée en termes de produit scalaire $\langle \cdot, \cdot \rangle$. Le centre de segment droit $[C_+, C_-]$ est défini par le point $C = (C_+ + C_-)/2$.

Nous calculons la classe de I par le test :

Si le vecteur $I - C$ connectant C à I joint un angle plus petit que $\pi/2$ avec le vecteur $w = C_+ - C_-$ connectant les moyennes des classes. Cela mène à :

$$\begin{aligned} y &= \text{sgn} \langle (I - C), w \rangle \\ &= \text{sgn} \langle (I - (C_+ + C_-)/2), (C_+ - C_-) \rangle \\ &= \text{sgn}(\langle I, C_+ \rangle - \langle I, C_- \rangle + b). \end{aligned} \quad (\text{III.19})$$

Ici, nous avons défini l'offset :

$$b = \frac{1}{2} \left(\|C_-\|^2 - \|C_+\|^2 \right) \quad (\text{III.20})$$

avec la norme $\|I\| = \sqrt{\langle I, I \rangle}$. Si les moyennes des classes ont la même distance à l'origine, alors b s'annule.

I.2.2. Phase de reconnaissance

Notons que (III.19) induit une frontière de la décision qui a la forme d'un hyperplan (Figure III.2); c'est un ensemble de points qui satisfait une contrainte exprimée comme une équation linéaire.

Il est utile de récrire (III.19) en fonction des entrées I_i , en utilisant le noyau k pour calculer les produits scalaires.

Par conséquent nous avons besoin d'exprimer les vecteurs C_i et w en fonction des vecteurs I_1, \dots, I_N .

À cette fin, remplaçant (III.17) et (III.18) dans (III.19) pour obtenir la *fonction de décision*.

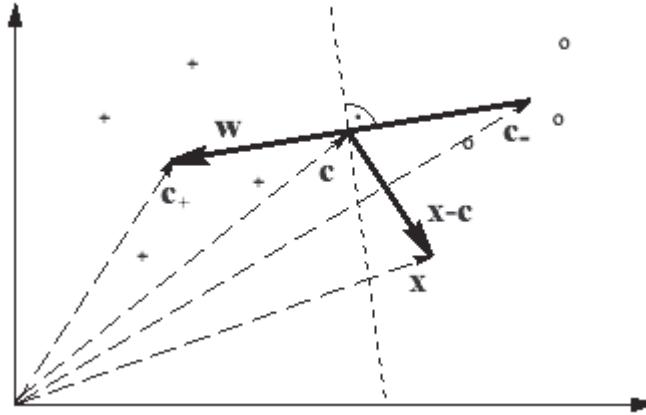


Figure III.2 : Un simple algorithme de la classification géométrique: soient deux classes de visages (représentées par ‘o’ et ‘+’). Calcule de leurs moyennes C_+ et C_- et affecte un visage de test I à une classe qui a la moyenne la plus proche. Cela peut être réalisé en regardant le produit scalaires entre $I - C$ (où $C = (C_+ + C_-)/2$ et $w = C_+ - C_-$, qui change le signe comme l’angle clos à travers $\pi/2$. Notons que la frontière de la décision correspondante est un hyperplan (la ligne pointillée) orthogonal à w .

$$\begin{aligned} y &= \text{sgn} \left(\frac{1}{N_h} \sum_{\{i|y_i=+1\}} \langle \Psi(I), \Psi(I_i) \rangle - \frac{1}{N_f} \sum_{\{i|y_i=-1\}} \langle \Psi(I), \Psi(I_i) \rangle + b \right) \\ &= \text{sgn} \left(\frac{1}{N_h} \sum_{\{i|y_i=+1\}} k(I, I_i) - \frac{1}{N_f} \sum_{\{i|y_i=-1\}} k(I, I_i) + b \right) \end{aligned} \quad (\text{III.21})$$

De même l'offset devient :

$$b = \frac{1}{2} \left(\frac{1}{N_f^2} \sum_{\{(i,j)|y_i=y_j=-1\}} k(I_i, I_j) - \frac{1}{N_h^2} \sum_{\{(i,j)|y_i=y_j=+1\}} k(I_i, I_j) \right) \quad (\text{III.22})$$

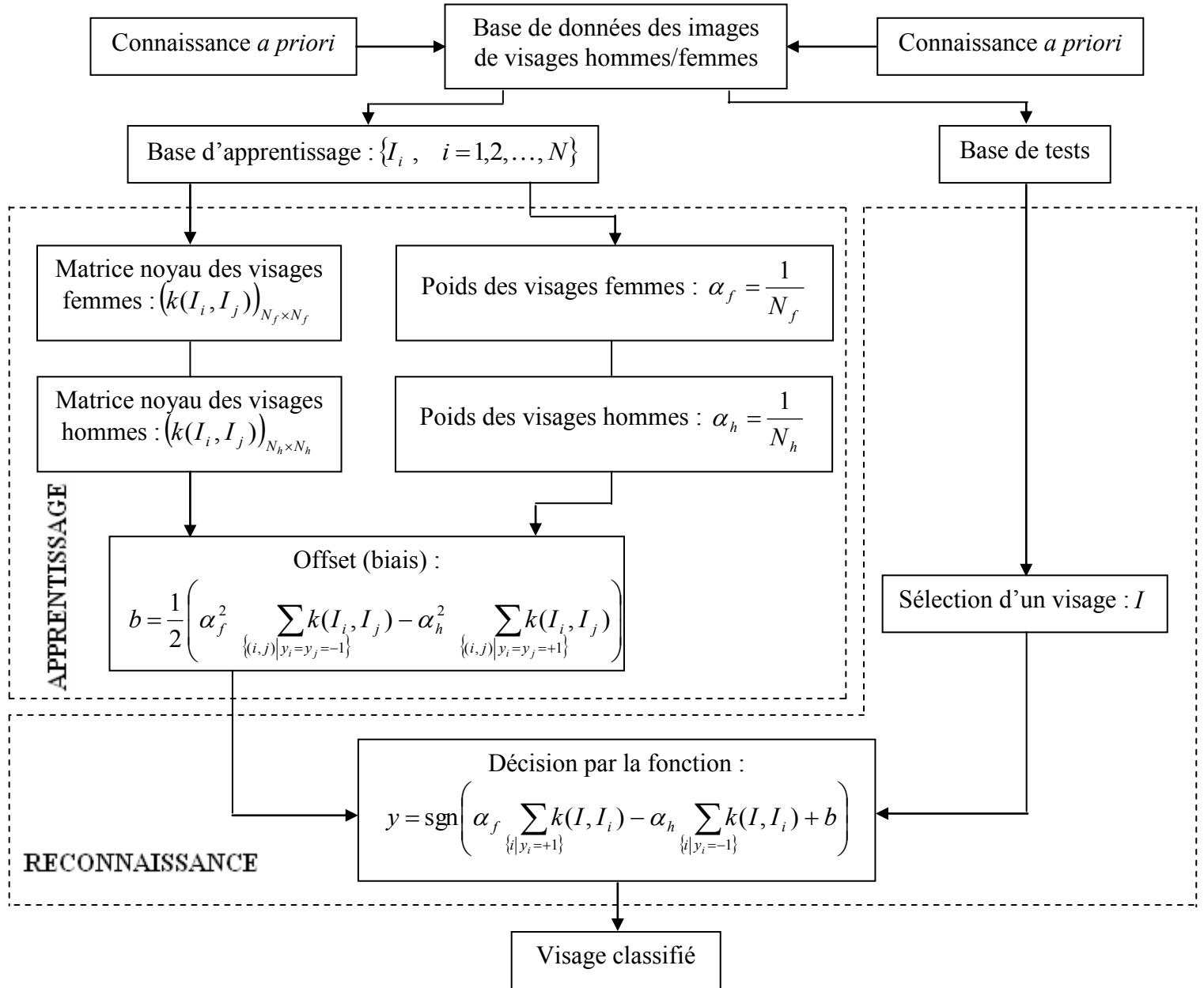


Figure III.3 : Organigramme de la reconnaissance de visages hommes/femmes par les SVMs

II. CONCLUSION

Dans ce chapitre, nous avons présenté la méthodologie de la reconnaissance de visages hommes/femmes à laquelle nous nous sommes intéressés. Comme nous avons aussi présenté les deux modèles que nous avons étudiés dans le cadre de la théorie des EigenFaces et la théorie des SVMs et que nous avons établis et implémentés pour reconnaissance de visages d'hommes ou de femmes.

Le chapitre suivant sera consacré à la présentation, la discussion et l'évaluation des différents résultats obtenus. Une comparaison sera également établie entre le résultat fourni par le premier modèle et celui fourni par le deuxième modèle.

CHAPITRE IV

**PRESENTATION DES RESULTATS, EVALUATION ET
COMPARAISON**

INTRODUCTION

La première partie de ce chapitre est consacrée d'abord, à la description des données utilisées, ces données ont été téléchargées à partir du site Internet [[www](#)]. Ensuite, et pour la mise en œuvre de la méthode des SVMs que nous avons étudiée et que nous avons décrite dans le chapitre précédent, il est nécessaire de transformer les images en utilisant une analyse en composantes principales (ACP) pour réduire la dimension des données. Cette étape ainsi que les résultats qui en découlent seront présentés et évalués. Enfin, les différents résultats de la reconnaissance de visages hommes/femmes selon les deux modèles de reconnaissance étudiés seront exposés, commentés et comparés.

La deuxième partie de ce chapitre est consacrée à la présentation générale de l'interface développée en utilisant le langage de programmation IDL.

I. DESCRIPTION DES DONNEES UTILISEES

Les données mises à notre disposition sont 1054 images de visages hommes/femmes en noir et blanc, codés sur 150*150 pixels.

I.1. Base d'apprentissage et base de tests

Un ensemble de données complexe est utilisé comme un ensemble d'apprentissage pour calculer les eignfaces, et pour construire les fonctions de discrimination par les SVMs. Il est composé de 750 images : 300 images de visages femmes et 450 images de visages hommes. Les images des deux classes de visages sont choisies aléatoirement à partir de la base de données [[www](#)]. Les 304 images restantes sont utilisées comme un ensemble du test.

I.2. Prétraitement des données pour la reconnaissance

Pour la mise en œuvre des deux modèles de reconnaissance, les différentes étapes de prétraitement suivantes ont déjà été effectuées :

1. une rotation, pour aligner les yeux sur une ligne horizontale,
2. une mise à l'échelle pour avoir des images 150*150 pixels,
3. une application d'un cache elliptique pour supprimer la partie de l'image qui entoure le visage,
4. une harmonisation de la courbe de niveau, pour qu'il y ait autant de pixel en blanc, en gris et en noir.

Toutes ces étapes ont été réalisées pour la simplification de tâche, afin que les visages aient une représentation qui soit la même, autant que possible.

II. PRESENTATION DES RESULTATS ET EVALUATION

L'expérience est présentée pour évaluer et comparer l'algorithme basé sur les *SVMs* en utilisant le noyau linéaire avec l'approche des *EigenFaces*.

Cette expérience est effectuée sur les images de la base de données téléchargée à partir du site Internet [[www](#)], qui est constituée de 614 visages d'hommes différents et de 440 visages de femmes différents. Chaque personne possède une image, prise une seule fois. Dans la figure IV.1, nous présentons les images de vingt individus de la base de données [[www](#)]. Il existe des variations au niveau des expressions faciales telles que : des yeux fermés / ouverts, souriants / non souriants, et aussi des détails faciaux tels que : avec ou sans lunettes. Toutes ces images sont prises contre un arrière plan homogène et sombre avec une position frontale et tout droit de la partie supérieure, et avec une tolérance de quelques mouvements faciaux. Il existe aussi quelques variations d'échelle.



Figure IV.1 : Les vingt images des deux classes (chacune dans une ligne) de la base de données [[www](#)].

La figure IV.2 illustre l'image moyenne ainsi que les sept premiers visages propres associées à l'ensemble d'apprentissage constitué de 30 visages femmes et de 30 visages hommes.



Figure IV.2 : EigenFaces : Image moyenne ainsi que les 7 premiers visages propres. Ces images ont été générées à l'aide d'une banque d'images contenant 30 visages de femmes et 30 visages d'hommes [[www](#)].

Pour effectuer l'apprentissage des *SVMs* par l'algorithme que nous avons proposé dans le chapitre précédent, il est nécessaire de transformer les images en utilisant une analyse en composantes principales (ACP) pour réduire la dimension des données. Ainsi, pour sélectionner le nombre de composantes qui seront conservées, nous devons analyser la proportion de variance conservée en fonction du nombre de composantes. En général, on

considère qu'en conservant 95% de la variance on en perd pas trop d'information. Bien entendu, nous pouvons considérer un seuil nettement inférieur à 95%. Il est utile de comparer les images initiales et les images réduites par l'ACP puis reconstruites par la projection inverse. Nous pouvons ainsi observer la quantité d'information perdue lors des manipulations par ACP.

La figure IV.3 illustre les trois premières composantes principales obtenues par l'application de l'ACP sur 20 images de visages femmes et 20 images de visages hommes sélectionnées à partir de l'ensemble d'apprentissage.

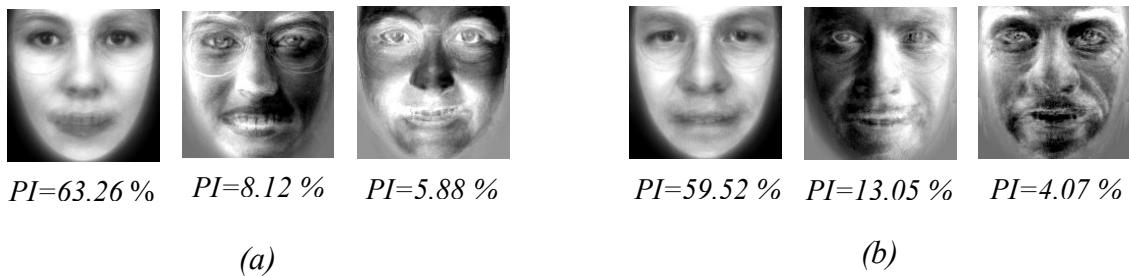


Figure IV.3 : ACP : Les trois premières composantes principales obtenues à partir (a) de 20 visages femmes, et (b) de 20 visages hommes de la base d'apprentissage [www].

L'indice *PI* désigne la portion d'information de chaque composante obtenue par application de l'ACP.

Dans nos expériences de reconnaissance de visages hommes/femmes que nous avons effectué sur la base de données téléchargée via le site Internet [www], nous avons sélectionné aléatoirement 750 échantillons (1 pour chaque individus) comme un ensemble d'apprentissage, à partir desquels nous calculons les eignfaces et nous construisons les machines à vecteurs du support (SVMs). Les 304 échantillons restants sont utilisés comme un ensemble de tests. Nous répétons de telles procédures trois fois, c-à-d, trois exécutions, leurs résultats sont stockés en 3 groupes de données. Pour chaque groupe, nous calculons les taux d'erreur contre le nombre des eignfaces (de 2-39) dont les résultats de la moyenne de trois exécutions sont donnés dans le tableau IV.1. La figure IV.4 montre les résultats de la moyenne de trois exécutions. Pour la comparaison, nous présentons les résultats des SVM et des EigenFaces dans les figures IV.4.a et IV.4.b. Il est évident que les taux de SVM sont plus inférieurs par rapport à ceux des EigenFaces. Le taux moyen d'erreur minimum des SVM en moyen est de 20,72 %, tandis que celui des EigenFaces est de 24,34 %. Si nous choisissons les meilleurs résultats parmi les trois groupes, le taux d'erreur le plus faible de la SVM peut atteindre 18,42 %.

Nombre de visages femmes	Nombre de visages hommes	Nombre de visages propres	Taux d'erreur par les Eigenfaces		Taux d'erreur par SVM (Noyau linéaire)
			Norme city-block	Norme euclidienne	
10	10	2	0.52	0.48	0.40
30	30	7	0.33	0.37	0.26
60	60	12	0.29	0.34	0.24
90	90	16	0.29	0.35	0.25
120	120	19	0.27	0.30	0.20
150	150	23	0.29	0.33	0.20
210	210	30	0.26	0.33	0.23
300	300	39	0.24	0.33	0.23

Tableau IV.1 : Les résultats du taux d'erreur en moyenne de trois exécutions.

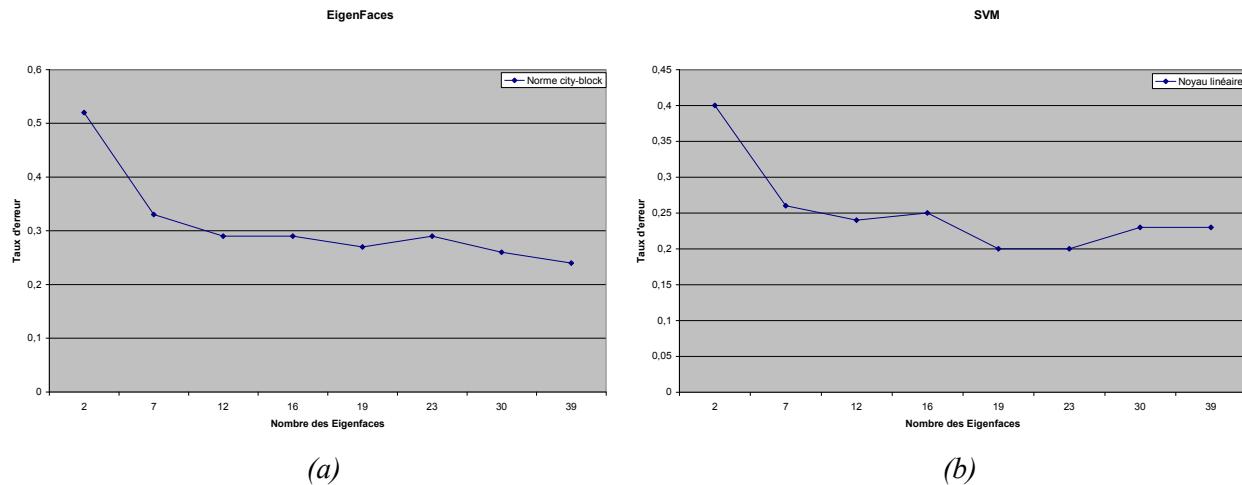


Figure IV.4 : La comparaison des taux d'erreur contre le nombre des eigenfaces des algorithmes (a) des EigenFaces standard et (b) de SVM appliqués sur les images de la base de données de visages hommes/femmes [www].

Les différents résultats que nous venons de présenter et d'évaluer sont obtenus à partir d'un logiciel que nous avons développé et dont nous présentons un bref aperçu dans ce qui suit.

III. PRESENTATION DU LOGICIEL DEVELOPPE

III.1. Description générale

Notre logiciel a été développé avec le langage IDL (Interactive Data Language) version 6.2 de l'entreprise Australienne RSI (Research System Inc), sous environnement Windows XP. Il permet d'effectuer plusieurs tâches telles que :

- La visualisation d'images à partir des fichiers d'extensions différentes (.img, .raw, .jpg, .bsq, .hdr) ;

- L'évaluation statistique des visages classifiés ;
 - La reconnaissance par les EigenFaces et les SVMs des visages hommes et femmes ;
- Notre interface se présente dans une fenêtre principale (figure IV. ?) qui englobe toutes les commandes d'exécution de notre application.

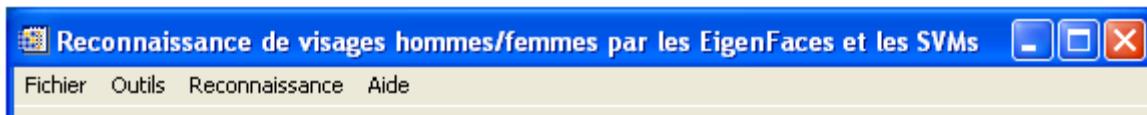


Figure IV. 5 : Fenêtre principale du logiciel

III.2. Menus

1. Menu fichier

Ce menu permet de réaliser les fonctions suivantes :

- **Ouvrir** : permet d'ouvrir des images (visage par visage, multi-sélection, format BSQ).
- **Fermer** : fermer toutes les images ouvertes.
- **Quitter** : quitter le logiciel.
- **Quitter IDL** : quitter le logiciel et l'IDL.

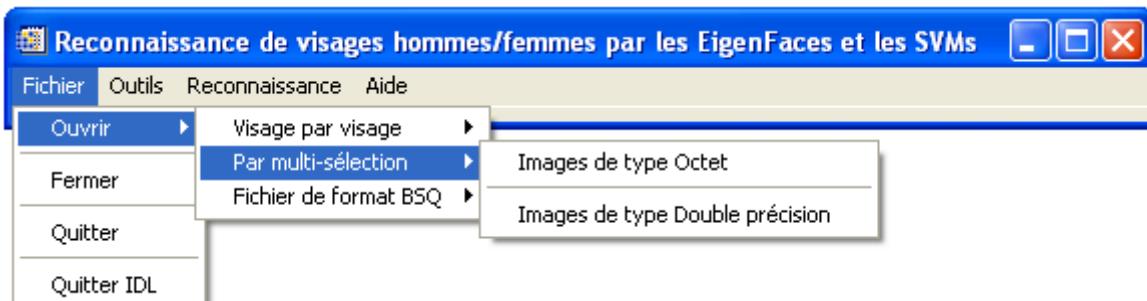


Figure IV. 6 : Menu ouvrir

2. Menu outils

Ce menu permet de réaliser les fonctions suivantes :

- Calculer le taux d'erreur obtenu par l'application de la méthode des EigenFaces et la méthode des SVMs (cas du noyau linéaire).
- Former une base d'apprentissage soit par création direct d'un format BSQ ou bien par réduction de données en appliquant l'ACP.

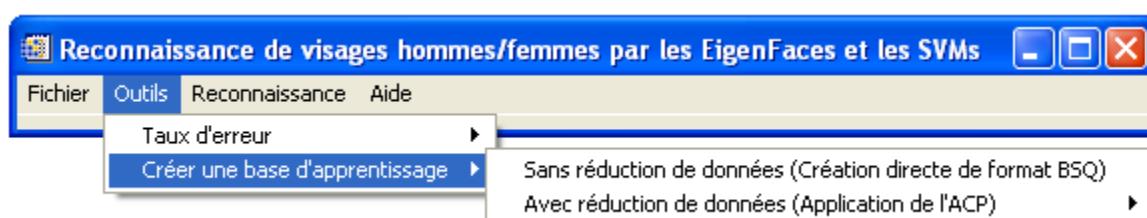


Figure IV. 7 : Menu outils

3. Menu reconnaissance

Ce menu comporte les commandes qui nous permet d'effectuer la reconnaissance de visages hommes/femmes soit par la méthode des visages propres (EigenFaces), soit par la méthode des machines à vecteurs de support (SVMs) en utilisant le noyau linéaire.

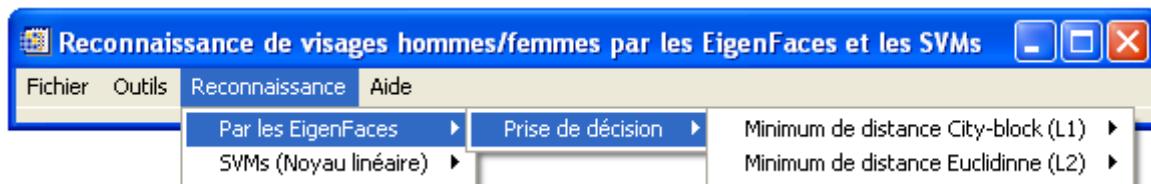


Figure IV. 8 : Menu reconnaissance

4. Menu aide

Ce menu donne la main sur les deux commandes :

- L'aide sur l'IDL
- L'aide à propos de

L'exécution de la deuxième commande permet l'affichage d'une boîte de message contenant des informations sur le logiciel.

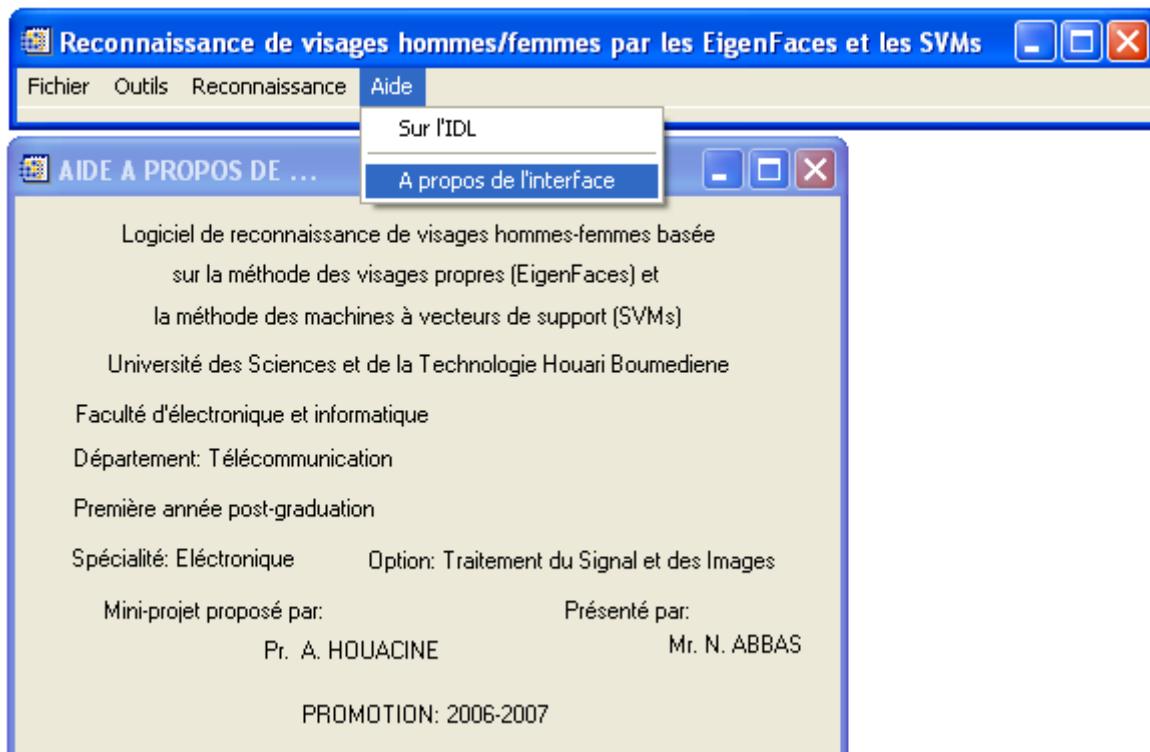


Figure IV. 9 : Menu aide

VI. CONCLUSION

Nous avons présenté les expériences de reconnaissance de visages hommes/femmes en utilisant le classifieur linéaire SVMs. En comparant cette approche avec la technique des EigenFaces, en constatant que la méthode des SVMs peut être utilisée d'une manière efficace dans le cadre de la reconnaissance de visages hommes/femmes. Les résultats expérimentaux montrent que les SVMs est considérée comme un algorithme très efficace par rapport à l'approche des EigenFaces dans le cadre de la reconnaissance de visages hommes/femmes.

CONCLUSION GENERALE

L'objectif de ce travail se positionne dans le cadre de la reconnaissance de visages hommes/femmes. En effet, notre but est la mise en œuvre de deux modèles : un premier modèle supervisé qui permet d'appliquer la méthode des visages propres (*EigenFaces*) en utilisant l'algorithme PPV, et un deuxième modèle supervisé basé sur la méthode des machines à vecteurs du support (SVMs) en utilisant le noyau linéaire. Ainsi, l'algorithme d'apprentissage utilisé pour la mise en œuvre de ce dernier modèle est proposé par Schölkopf et Smola dans [41]. L'application du deuxième modèle aboutit à la reconnaissance d'un visage avec un meilleur taux d'erreur.

Ainsi, dans un premier temps, nous avons présenté les techniques de la reconnaissance de visages les plus rencontrées dans la littérature, et nous avons attardé sur l'approche standard « *EigenFaces* » que nous avons utilisé dans la mise en œuvre du premier modèle, et ceci en utilisant l'algorithme du K plus proche voisin (K -ppv). Ensuite nous nous sommes intéressé au concept de la méthode des SVMs qui découle de la théorie d'apprentissage artificiel de Vapnik [4] et son application au problème de la reconnaissance de visages. Dans la deuxième partie, nous avons présenté la méthodologie de reconnaissance de visages hommes/femmes que nous avons adopté avec la présentation des deux modèles considérés. La dernière partie de ce travail est consacrée à la présentation, l'interprétation, l'évaluation et la comparaison des différents résultats obtenus en utilisant une base de données des visages hommes/femmes formée d'un ensemble d'apprentissage et d'un ensemble de tests.

Nous avons aussi présenté le logiciel que nous avons développé sous environnement **Windows Xp** en utilisant le langage de programmation **IDL**.

A partir de cette étude nous constatons que la méthode des SVMs peut être utilisée d'une manière efficace par rapport à celle des EigenFaces dans le cadre de la reconnaissance de visages hommes/femmes. En outre, les SVMs permettent de faire une classification non linéaire et linéaire, mais ils ne permettent pas de faire une classification autre chose que binaire (sauf si nous utilisons plusieurs SVMs).

Comme perspectives, nous proposons d'élargir le champs d'applicabilité de cette approche aux cas de :

- Reconnaissance d'objets ;
- Reconnaissance d'écriture manuscrite ;
- Catégorisation de texte ;

- Identification de locateur ;
- Bio-informatique.

REFERENCES BIBLIOGRAPHIQUES

- [1] **Yessaadi Sabrina, M. T. Laskri.**, «Un modèle basé templates Matching/Réseau de neurones pour la reconnaissance des visages humains». Groupe de recherche en intelligence artificielle, Département d'informatique, Université d'Annaba. livesab2002@gmail.com.
- [2] **Guodong Guo, Stan Z. Li, and Kapluk Chan.**, «Face Recognition by Support Vector Machines». School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 {egdguo, eszli, eklchan}@ntu.edu.sg.
- [3] **V. N. Vapnik.** Statistical learning theory. John Wiley & Sons, New York, 1998.
- [4] **Vapnik, V., 1995.** The nature of statistical learning theory. Springer-Verlag, New-York, USA, 1995.
- [5] **Mithridate Mahmoudi.**, «Logiciels de traitement d'images». Imasys, "Le Quadral", 23 bis rue Édouard Nieuport 92150 Suresnes, MESURES 753 - MARS 2003.
- [6] **ALEXANDRE LEMIEUX.**, Système d'identification de personnes par vision numérique. Mémoire pour l'obtention du grade maître ès sciences (M.Sc.), département de génie électrique et de génie informatique, faculté des sciences et de génie, Université Laval Québec, Décembre 2003.
- [7] **M. Kirby et L. Sirovich.**, Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(1):103–108, 1990.
- [8] **Matthew Turk et Alex Pentland.**, Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.
- [9] **M. A. Turk and A. P. Pentland.**, Eigenfaces for recognition. *J. Cognitive Neurosci.*, 3(1):71-86, 1991.
- [10] **P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi et Patrick J. Rauss.**, The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.
- [11] **Diego A. Socolinsky et Andrea Selinger.**, A comparative analysis of face recognition performance with visible and thermal infrared imagery. Dans 16th International Conference on Pattern Recognition (ICPR), volume 4, pages 217–222, August 2002.
- [12] **K. Rao et P. Yip.**, *Discrete Cosine Transform : Algorithms, Advantages, Applications*. Academic Press, 1990.
- [13] **Ziad M. Hafed et Martin D. Levine.**, Face recognition using discrete cosine transform. *International Journal of Computer Vision*, 43(3):167–188, July – August 2001.
- [14] **Matthew Turk et Alex Pentland.**, Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

- [15] **Sebastien Marcel et Samy Bengio.**, Improving face verification using skin color information. Dans *16th International Conference on Pattern Recognition (ICPR)*, pages 378–381, Québec, Canada, August 2002.
- [16] **Bernard Achermann et H. Bunke.**, Classifying range images of human faces with hausdorff distance. Dans *International Conference on Pattern Recognition (ICPR)*, pages 813–817, 2000.
- [17] **A. Pentland, B. Moghaddam et T. Starner.**, View-based and modular eigenspaces for face recognition. Dans *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA, June 1994.
- [18] **Ferdinando Samaria.**, *Face Recognition Using Hidden Markov Models*. Thèse de doctorat, Engineering Department, Cambridge University, Trumpington Street, Cambridge CB2 1PZ, UK, October 1994.
- [19] **Ara V. Nefian et Monson H. Hayes III.**, An embedded hmm based approach for face detection and recognition. Dans *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume VI, pages 3553–3556, March 1999.
- [20] **R. Brunelli et T. Poggio.**, Face recognition : features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(10):1042–1052, October 1993.
- [21] **Roberto Brunelli et Tomaso Poggio.**, Face recognition through geometrical features. Dans *European Conference on Computer Vision (ECCV)*, pages 792–800, 1992.
- [22] **Sebastien Marcel et Samy Bengio.**, Improving face verification using skin color information. Dans *16th International Conference on Pattern Recognition (ICPR)*, pages 378–381, Qu’ebec, Canada, August 2002.
- [23] **Matthew Turk et Alex Pentland.**, Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.
- [24] **A. Franco, A. Lumini et D. Maio.**, Eigenspace merging for model updating. Dans *16th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 156–159, Québec, Canada, August 2002.
- [25] **W. Yambor, B. Draper et R. Beveridge.**, Analyzing PCA-based Face Recognition Algorithms : Eigenvector Selection and Distance Measures. Dans H. Christensen et J. Phillips, éditeurs : *Empirical Evaluation Methods in Computer Vision*. World Scientific Press, Singapore, 2002.
- [26] **E. Osuna, R. Freund, et F. Girosi.** Support vector machines : Training and applications. Rapport technique AIM-1602, Center for Biological and computational Learning, Massachusetts Institute on Technology, Cambridge, 1997.
- [27] **Yassine BEN AYED.**, Detection de mots clés dans un flux de parole. Thèse de Doctorat en électronique, spécialité Signal et Images, Ecole Nationale Supérieure des Télécommunications, Laboratoire Lorrain de Recherche en Informatique et ses Applications, 2003.
- [28] **K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, et B. Schölkopf.** An introduction to kernel based learning algorithms. *Transactions on Neural Networks*, 12, 2001.
- [29] **C. Cortes and V. Vapnik.** Support vector networks. *Machine Learning*, 20:273-297, 1995.
- [30] **B. Scholkopf.**, " Support vector learning " Thesis, R. Oldenbourg verlag, Munich, 1997.

[31] **Jamal Kharrobi.**, Etude de Techniques de Classement " Machines à Vecteurs Supports " pour la Vérification Automatique du Locuteur. Thèse de Doctorat en électronique, spécialité Signal et Images, Ecole Nationale Supérieure des Télécommunications, 2002.

[32] **J. Weston et C. Watknis.** Multi-class support vector machines. Rapport technique, CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998.

[33] **Y. Guermeur, A. Elisseeff, et H. Paugam-Moisy.** Estimating the sample complexity of a multi-class discriminant model. In *proceedings of the Industrial Conference on Artificiel Neural Networks*, pages 310-315, 1999.

[34] **C. Hsu et C. Lin.** A comparaison of methods for muli-class support vector machines. Rapport technique, Department of Computer Science and Information Engineering, National Taiwan University, 2001.

[35] **J. Weston et C. Watknis.** Support vector machines for multiclass pattern recognition. In *Proceedings of the Seventh European Symposium On Artificial Neural Networks*, 1999.

[36] **L. Bottou, C. Cortes, J. Drucker, I. Guyon, Y. LeCunn, U. Muller, E. Sackinger, P. Simard, et V. Vapnik.** Comparaison of Classifier methods : a case study in hundwriting digit recognition. In *Proceedings of the International Conference on Pattern Recognition*, pages 77-87, 1994.

[37] **S. Knerr, L. Personnaz, et G. Dreyfus.** Single-layer learning revisited : a stepwise procedure for building and training a neural network. *Neurocomputing : algorithms, Architectures and Applications*, 68, 1990.

[38] **U. Krebel.** *Advances in kernel methods support vector learning*. MIT Press, 1998.

[39] **J. Friedman.** Another approch to polychotomous classification. Rapport technique. Department of Statistics Stanford University, 1996.

[40] **J. C. Platt, N. Cristianini, et J. Shawe-Taylor.** Large margin dags for multiclass classification. In *Advances in Neural Information Processing Systems*, MIT Press, 12:547-553, 2000.

[41] **Bernhard Schölkopf, Alexander J. Smola.** Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press, Cambridge, Massachusetts, London, England 2002.

Site Internet consulté

[www] <http://www.dice.ucl.ac.be/~verleyse/lectures/elec2870/elec2870.htm>