Fusion de différents modes de capture pour la reconnaissance du visage appliquée aux e_transactions

Pierre Buyssens

Université de Caen
Basse-Normandie

Université de Caen Basse-Normandie

U.F.R. : Sciences

École doctorale SIMEM

THÈSE

Présentée par

Pierre Buyssens

Et soutenue

Le 4 Janvier 2011

En vue de l'obtention du

DOCTORAT de l'UNIVERSITÉ de CAEN
Spécialité : informatique et applications
(Arrêté du 07 août 2006)

Fusion de différents modes de capture pour la reconnaissance du visage appliquée aux e_transactions

Membres du jury

Mme	Bernadette	Dorizzi	Professeur	Télécom & Management SudParis	.....Rapporteur
M.	Christophe	Garcia	Professeur	Insa de Lyon	.....Rapporteur
M.	Olivier	Lézoray	Professeur	Université de Caen Basse-Normandie	..... Examinateur
Mme	Marinette	Revenu	Professeur	Ensicaen	.....Directrice de Thèse

Fusion de différents modes de capture pour la reconnaissance du visage appliquée aux e_transactions

Résumé :

Dans cette thèse, nous décrivons le problème de la reconnaissance automatique de visages en lumière visible et infrarouge (grandes longueurs d'ondes). Les principales méthodes de la littérature sont étudiées, et nous nous concentrons dans un premier temps sur une méthode fondée sur les réseaux de neurones à convolution. Appliquée aux deux modalités, la méthode de fusion par pondération des scores proposée permet d'augmenter sensiblement les taux de reconnaissance. Dans un deuxième temps, l'application de méthodes de préapprentissage du réseau via des approches parcimonieuses est étudiée pour la reconnaissance faciale. Enfin, nous nous penchons sur une approche de décomposition parcimonieuse de visages, couplée à une méthode de classification fondée sur une minimisation l_1. Cette dernière approche permet d'atteindre de bons taux d'identification sur la base de référence Notre–Dame.

Mots clés : Reconnaissance de visages, identification, modalité visible, modalité infrarouge, fusion de modalités, réseaux de neurones à convolution, décomposition parcimonieuse.

Abstract :

In this thesis, we describe the problem of automatic face recognition in visible and long–wave infrared lights. The state of the art methods are described, and we study, in a first time, a method based on convolutional neural networks. Applied to both modalities, the proposed fusion method, based on a weighted sum of scores, yields a substantial increasing of the recognition rates. In a second time, a pretraining of the network with sparse methods is studied for automatic facial recognition. Finally, we propose an approach based on a sparse decomposition of faces, coupled with a classification scheme involving in a l_1 minimization. This last approach gives good identification rates on the well known Notre–Dame database.

Keywords : Facial recognition, identification, visible modality, long–wave infrared modality, modality fusion, convolutional neural networks, sparse decomposition.

Discipline : Informatique et applications

Laboratoire : GREYC, 6 Boulevard du Maréchal Juin, 14050 Caen Cedex, FRANCE

Remerciements

Je voudrais en premier lieu remercier Mme Bernadette Dorizzi, professeur à Télécom & Management SudParis, et M. Christophe Garcia, professeur à l'INSA de Lyon, d'avoir accepté de rapporter ce travail malgré le peu de temps que je leur ai laissé.

Je tiens à remercier également M. Olivier Lézoray, professeur à l'université de Caen, d'avoir accepté de participer au jury en tant qu'examinateur.

Cette thèse s'est déroulée dans un premier temps au sein d'Orange Labs à Caen sous l'encadrement de M. Olivier Lepetit. Je tiens à le remercier pour avoir proposé ce sujet chez Orange Labs ainsi que pour sa gentillesse et son soutien. Je remercie également toute l'équipe au sein de laquelle j'évoluais chez Orange Labs pour son accueil.

Je tiens à remercier vivement ma directrice de thèse, Marinette Revenu, pour m'avoir guidé et encouragé tout au long de ces années de thèse. Ses conseils, sa disponibilité et sa patience m'ont permis de faire évoluer de manière significative mes travaux de recherche après mon départ de chez Orange Labs.

Je tiens à remercier toute l'équipe Image du Greyc, doctorants, post–doctorants et permanents pour leurs qualités humaines et/ou scientifiques, et qu'il serait risqué de nommer individuellement, de peur d'en oublier.

Je remercie par ailleurs ma famille et mes amis (dont certains font ou ont fait partie du Greyc). Les citer tous serait fastidieux et très probablement incomplet, aussi je ne m'y risque pas.

Mes pensées vont également vers Choup', Arthur, Texane, Jean Coqteau et son harem (Oliver Poule et Swimming Poule), ainsi que ma saxo avec qui j'ai passé quotidiennement plusieurs heures.

Je tiens finalement à remercier Julie pour son soutien de tous les instants et qui partage ma vie, ainsi que mon fils Martin, qui grâce à son sommeil « difficile » m'a permis la rédaction nocturne d'une bonne partie de ce manuscrit.

Table des matières

Table des figures

Liste des algorithmes

Liste des tableaux

Première partie

Introduction à la biométrie et état de l'art des techniques de reconnaissance faciale

Chapitre 1

Introduction

1.1 Contexte

L'utilisation de caractères anthropométriques pour la reconnaissance de personnes est le moyen naturel chez l'homme pour reconnaître une personne. Cela passe par la voix, la démarche et bien sûr par le visage, le moyen le plus naturel pour identifier quelqu'un. L'identification grâce à l'empreinte du pouce servait déjà lors d'échanges commerciaux à Babylone dans l'Antiquité ainsi qu'en Chine au VII^{ème} siècle. C'est Alphone Bertillon, grand criminologiste français, qui, au milieu du XIX^{ème} siècle énonce l'idée d'utiliser des mesures physiologiques pour l'identification de suspects. Cependant, les procédures manuelles sont longues et coûteuses, et requierent un nombre important d'opérateurs.

Aujourd'hui, la puissance de calcul des ordinateurs peut être mise à contribution pour réaliser la même tâche de manière automatique. C'est notamment le besoin grandissant en sécurité qui a permis l'émergence puis le développement du domaine de la biométrie. Les applications de la biométrie, essentiellement sécuritaires, sont en effet nombreuses : accès à des locaux sensibles, sécurisation de transaction, accès à des ressources informatiques ... Les utilisations sont évidemment également policières : recherche d'individus, reconnaissance de fauteurs de troubles, mais aussi disculpation de suspects ...

Les moyens classiques de sécurisation sont le badge, la carte à puce ou encore le mot de passe. Ces moyens, bien qu'efficaces, souffrent de plusieurs défauts majeurs. En effet, un mot de passe peut être oublié ou volé, un badge falsifié, ou une carte à puces piratée. C'est précisément en complément de ces moyens classiques que la biométrie peut exister. En effet, une personne porte *sur elle* ses propres données biométriques. Ces caractéristiques ne peuvent donc être perdues, et sont difficilement falsifiables. L'utilisation de la biométrie peut également se faire en complément des

méthodes classiques, par exemple lorsque la donnée biométrique d'une personne est contenue dans une carte à puce. C'est le cas notamment du passeport biométrique qui depuis juin 2009 présente, outre la photo du détenteur, la numérisation de deux empreintes digitales du possesseur du passeport.

La thèse s'est déroulée durant trois ans chez Orange Labs à Caen, puis durant une année au sein du GREYC [1]. L'objectif de la thèse est l'étude de modalités biométriques sans contact pour une utilisation comme brique de base supplémentaire pour les transactions électroniques sécurisées. Une telle brique de sécurisation peut en effet intervenir en plus des moyens classiques de sécurisation tels les mots de passe ou les badges. Cette thèse a été proposée dans la lignée de précédents travaux sur la biométrie sans contact effectués chez Orange Labs, notamment la reconnaissance de personnes par la main (thèse effectuée par Julien Doublet [87]). Il s'agit ici d'étudier différentes modalités pour la reconnaissance faciale, ainsi que la pertinence de leur fusion. Les modalités initialement considérées sont la modalité visible déjà étudiée dans le cadre de travaux effectués à Orange Labs Rennes, la modalité infrarouge (thermique) pour laquelle Orange Labs Caen s'est équipé d'une caméra d'acquisition utilisée dans le contexte de la main, ainsi que la modalité 3D qui a vu une thèse commencer à Orange Labs Lannion portant sur la reconstruction de visages 3D à partir de plusieurs captures 2D. L'arrêt du projet de reconstruction de visages 3D, et plus généralement l'arrêt de l'étude de la biométrie chez Orange Labs nous a conduit à ne plus considérer que les modalités visible et infrarouge, et à privilégier les aspects méthodologiques au détriment des contraintes liées aux e_transactions.

1.1.1 Définition de la biométrie

La biométrie désigne dans un sens très large l'étude quantitative d'une population à l'aide des mathématiques, et plus précisément dans le cas qui nous intéresse, la reconnaissance et l'identification d'individus à l'aide d'une ou de plusieurs caractéristiques physiologiques. Cependant, toutes les caractéristiques d'un être humain ne peuvent pas être utilisées comme moyens d'identification. En effet, pour qu'un système biométrique puisse fonctionner en environnement réel, les caractéristiques physiologiques doivent satisfaire les conditions suivantes [75] :

- être universelles : la caractéristique doit être possédée par chaque individu,
- uniques : la caractéristique doit permettre la différenciation d'un individu par rapport à un autre,
- permanentes : la caractéristique doit être invariante dans le temps,
- enregistrables : la caractéristique doit pouvoir être acquise.

1. Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen

D'autres conditions sont également souvent ajoutées lors de la création d'un système biométrique :

- performante : la caractéristique doit permettre au système de reconnaître efficacement un individu, en minimisant autant que possible les fausses alarmes,
- acceptable : la caractéristique doit être acceptée par les utilisateurs, ce point dépend de la façon dont est perçue la caractéristique par la population,
- infalsifiable : la caractéristique doit être difficilement falsifiable afin d'éviter une utilisation frauduleuse du système.

Les différentes caractéristiques par lesquelles il est possible d'identifier un individu sont appelées modalités biométriques (Figure 1.1). Les plus couramment utilisées (ou étudiées) sont :

- L'empreinte digitale, une des premières (sinon la première) modalités utilisées. Cette modalité a donné lieu à de très nombreux travaux, et est une des premières applications ayant donné naissance à des produits finis. Notons qu'elle requiert la pose d'un doigt sur un capteur, ce qui peut être mal accepté dans certaines cultures où les questions d'hygiène sont importantes.

- La géométrie de la main comprenant la longueur des doigts ou la largeur de la main. Cette modalité requiert souvent un guide où la main vient se glisser, rendant ainsi la détection/segmentation bien plus aisée. Notons que des méthodes sans contact ont également été développées.

- L'empreinte de la paume de la main où les lignes de la main sont souvent utilisées comme caractéristiques. Cette modalité vient souvent en complément de la géométrie de la main, l'utilisateur n'ayant en effet pas de manipulation supplémentaire à effectuer.

- La voix souvent bien perçue car universelle et sans contact. Cette modalité permet en outre des reconnaissances distantes via un téléphone par exemple.

- L'iris, zone circulaire entourant la pupille. Supposée unique et invariable dans le temps, cette modalité nécessite cependant la coopération de l'utilisateur, la capture pour être utilisable doit en effet être effectuée à une certaine distance maximale de l'objectif.

- La rétine (ou fond de l'œil) est encore plus contraignante que l'iris, mais offre encore davantage de fiabilité.

- L'ADN (ou Acide DésoxyriboNucléique) supposé unique à chaque individu.

Son utilisation reste difficile et son exploitation longue. De plus, l'ADN contient bien plus d'informations que l'identité de l'individu. Son utilisation reste limitée aux enquêtes policières (et c'est tant mieux !).

– Le visage en tant que moyen le plus naturel de reconnaître une personne. Modalité sans contact, une capture peut être réalisée à l'insu des personnes. De nombreuses recherches sont effectuées dans ce domaine, mais les problèmes sont nombreux.

– La géométrie de l'oreille permet l'identification de personnes de profil, et peut venir en complément d'une technologie basée sur le visage.

– La démarche, supposée unique (ou presque) à chaque individu. Les déformations des jambes et bras au niveau des articulations permet la détection du comportement ainsi que la reconnaissance de personnes.

– La signature en tant que biométrie comportementale. Cette modalité peut cependant être variable au cours du temps.

– La dynamique de frappe au clavier où les intervalles de temps entre les pressions successives de deux touches d'un clavier ainsi que la durée des pressions peuvent être utilisées pour une authentification. Cette modalité présente l'avantage d'être à faible coût, et peut venir en complément d'une vérification par mot de passe.

Nombre de ces modalités peuvent de plus donner naissance à d'autres modalités lorsque le capteur est modifié. Par exemple, lors de la capture d'un visage à l'aide d'une caméra infrarouge, il s'agit toujours du visage mais on parle alors de modalité infrarouge du visage.

Notons également que d'autres modalités biométriques ont émergé depuis peu : ainsi il a été montré que des articulations des doigts peuvent être utilisées pour la reconnaissance d'un individu [167]. Cette modalité peut judicieusement être couplée à la géométrie de la main, l'empreinte de la paume ou encore l'empreinte digitale au sein d'un même capteur, ce qui ne réduit pas l'acceptabilité de cette modalité.

L'analyse des ondes émises par le cerveau humain est également une modalité étudiée. La collectabilité de cette modalité est cependant limitée.

Enfin, la modalité concernant la signature peut être étendue par l'analyse de la pression ou de l'inclinaison d'un stylo avec lequel une personne signe un document [140].

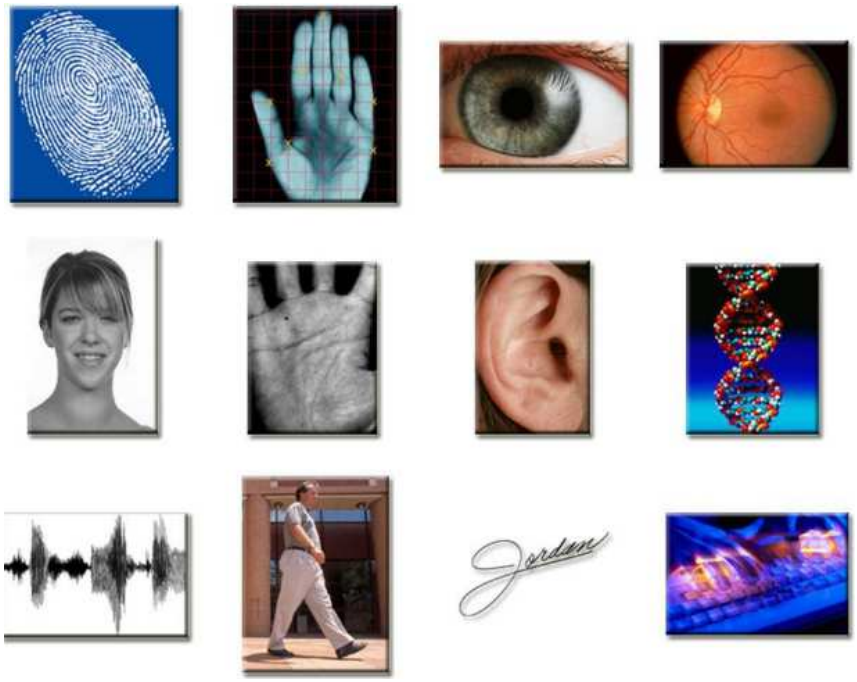


FIGURE 1.1 – Quelques modalités biométriques.

Toutes ces modalités présentent des avantages et des inconvénients selon les critères établis plus haut. Le tableau 1.1 (extrait de [150]) compare les principales modalités biométriques selon plusieurs critères.

1.1.2 Enrôlement/Authentification/Identification

Les systèmes biométriques fonctionnent selon trois modes de fonctionnement :

- l'*enrôlement*,
- l'*authentification* (ou *vérification*),
- l'*identification*.

L'enrôlement (voir la figure 1.2) est la première étape de tout système biométrique : il s'agit de l'*enregistrement* d'un utilisateur dans le système. L'individu souhaitant se faire enrôler clame son identité et le système capture une ou plusieurs caractéristiques physiologiques de la personne. Ses caractéristiques sont ensuite insérées dans une base de données biométriques permettant de relier un vecteur de caractéristiques à une identité.

Critère / Modalité	U	N	P	F	E	A	I
ADN	+	+	+	-	+	-	-
Démarche	=	-	-	+	-	+	=
Frappe au clavier	-	-	-	=	-	=	=
Empreinte digitale	=	+	+	=	+	=	+
Paume de la main	=	+	+	=	+	=	+
Géométrie de la main	=	=	=	+	=	=	=
Iris	+	+	+	=	+	-	+
Rétine	+	+	=	-	+	-	+
Signature	-	-	-	+	-	+	+
Thermographie de la main	=	=	=	=	=	=	+
Thermographie du visage	=	=	=	-	=	=	+
Visage	+	-	=	+	-	+	-
Voix	=	-	-	=	-	+	-

TABLE 1.1 – Comparaison des systèmes biométriques selon différents critères : (U) Universalité, (N) Unicité, (P) Permanence, (F) Facilité d'enregistrement, (E) Performance, (A) Acceptabilité, (I) Infalsifiabilité. Notation : (+) fort, (=) moyen, (-) faible.

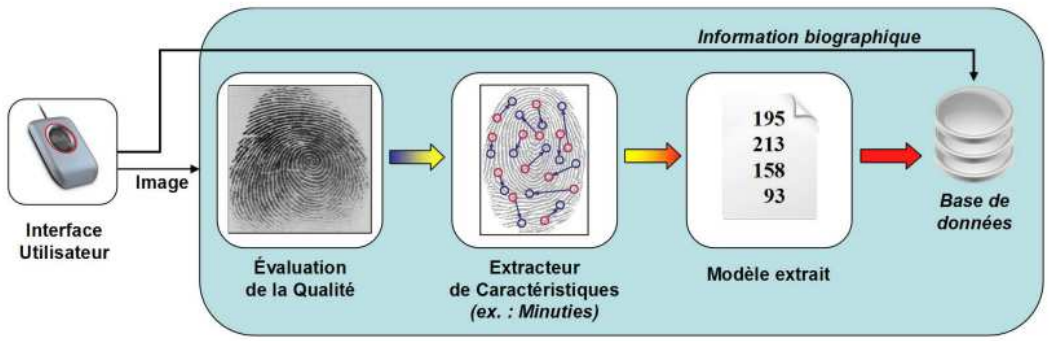


FIGURE 1.2 – Enrôlement d'une personne dans un système biométrique (ici l'empreinte digitale).

L'authentification (voir la figure 1.3) est un type d'application pour lequel le système doit répondre à la question :

– *Suis–je bien la personne que je prétends être ?*

Le cas d'usage est une personne clamant son identité au système et celui–ci doit alors vérifier si la personne est bien la personne qu'elle prétend être.

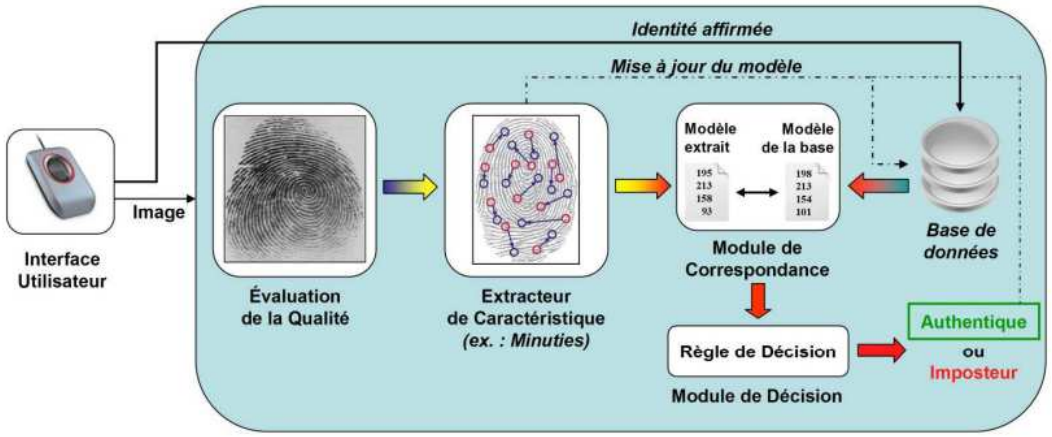


FIGURE 1.3 – Principe de l'authentification d'un individu dans un système biométrique.

Parmi les applications liées à l'authentification, citons l'accès à des données sécurisées, des ressources informatiques ou encore des transactions sécurisées.

Le problème de l'authentification peut être formalisé par :
Soit le vecteur d'entrée X_p définissant les caractéristiques biométriques extraites par le système lorsqu'une personne p se présente devant celui–ci, et I_c l'identité clamée par cette personne. Le système doit déterminer la valeur booléenne de $f(E_p, I_c)$ permettant de déclarer l'individu comme étant le bon utilisateur ou un imposteur. La fonction f peut ainsi être définie :

$$f(X_p, X_{I_c}) = \begin{cases} 1 & \text{si } S(X_p, X_{I_d}) \geq s \\ 0 & \text{sinon} \end{cases}$$

où X_{I_d} est le vecteur de caractéristiques correspondant à l'identité clamée I_d, S est la fonction de similarité définissant la *correspondance* entre les deux vecteurs, et s est le seuil à partir duquel les deux vecteurs sont déclarés correspondre (et donc les identités). Notons que le seuil est supérieur à 0 étant donnée la variabilité qui peut intervenir entre deux captures d'une même modalité pour une même personne.

L'identification (voir la figure 1.4) est un type d'application pour lequel le système doit répondre à la question :

– *Qui suis–je ?*

Le système doit trouver l'identité d'une personne parmi celles d'une base de données contenant des personnes déjà enrôlées, et renvoyer l'identité correspondant

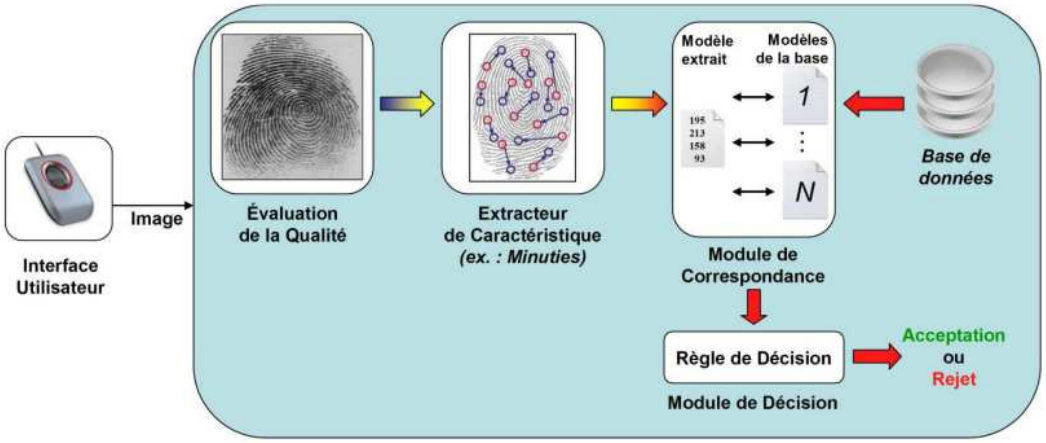


FIGURE 1.4 – Principe de l'identification d'un individu dans un système biométrique.

à la personne se présentant devant le système, ou l'identité « inconnue » si cette personne ne fait pas partie de la base. Il s'agit ici d'une comparaison 1 à n où n représente le nombre de personnes dans la base, également appelée galerie.

Parmi les utilisations possibles d'un système en mode d'identification, on retrouve la recherche d'individus dangereux, ou encore l'accès restreint d'un bâtiment d'une entreprise à ses seuls employés par exemple.

Le problème de l'identification peut être formalisé par :

Soit le vecteur d'entrée X_p définissant les caractéristiques biométriques extraites par le système lorsqu'une personne p se présente devant celui–ci, l'identification revient à déterminer l'identité de $I_t, t \in \{0, 1, 2, \ldots, N\}$ où $I_1, I_2, \ldots, I_N$ sont les identités des individus préalablement enrôlés dans le système, et I_0 indique une identité inconnue. La fonction d'identification f peut ainsi être définie :

$$f(X_p) = \begin{cases} I_k & \text{si } \max_{1 \leq k \leq N} S(X_p, X_{I_k}) \geq s \\ I_0 & \text{sinon} \end{cases}$$

où X_{I_k} est le vecteur de caractéristiques correspondant à l'identité I_k, S est la fonction de similarité, et s un seuil fixé.

Notons que dans le reste de la thèse, nous nous sommes placé dans le cadre de l'*identification* dans un contexte fermé, c'est à dire sous l'hypothèse forte que l'identité recherchée se trouve dans la base de données. Nous nous sommes ainsi affranchi de la notion de seuil inhérente à l'*authentification* ainsi que le seuil d'acceptabilité de l'*identification* présent dans un contexte ouvert. Toutes les méthodes présentées dans la suite peuvent facilement être adaptées à ces deux cas.

1.1.3 Décomposition en modules

Un système biométrique typique est un système de reconnaissance de forme pouvant être représenté par quatre modules :

- le **module de capture** : il est responsable de l'acquisition de la donnée biométrique d'un individu. Il peut s'agir d'un appareil photo, d'une caméra ou encore d'un lecteur d'empreinte digitale. Des exemples sont présentés à la figure 1.5.

- le **module d'extraction de caractéristiques** : il détermine à partir de la donnée acquise par le module de capture la nouvelle représentation des données. Cette nouvelle représentation doit être pertinente, idéalement unique pour chaque personne et invariante aux modifications qui peuvent intervenir sur la capture.

- le **module de similarité** : il compare les données biométriques extraites par le module d'extraction de caractéristiques à un ou plusieurs modèles préalablement enregistrés. Ce module détermine le niveau de similitude (ou de divergence) entre deux empreintes biométriques.

- le **module de décision** : il détermine si le degré de similitude retourné par le module de similarité est suffisant pour déterminer l'identité d'un individu.

1.1.4 Mesure de la performance d'un système biométrique

La performance d'un système biométrique est un élément essentiel à prendre en compte dans le choix d'un tel système. La mesure de performance d'un système biométrique s'articule autour de trois critères :

- le premier critère est le **taux de faux rejet** (« False Reject Rate » ou FRR). Ce taux représente le pourcentage d'individus censés être reconnus par le système mais qui sont rejetés. Le système classe alors deux caractéristiques biométriques provenant de la même personne comme provenant de deux personnes différentes.

- le second critère est le **taux de faux accepté** (« False Acceptance Rate » ou FAR). Ce taux représente le pourcentage d'individus reconnus par le système biométrique alors qu'ils n'auraient pas dû l'être. Le système classe alors deux caractéristiques provenant de deux personnes différentes comme appartenant

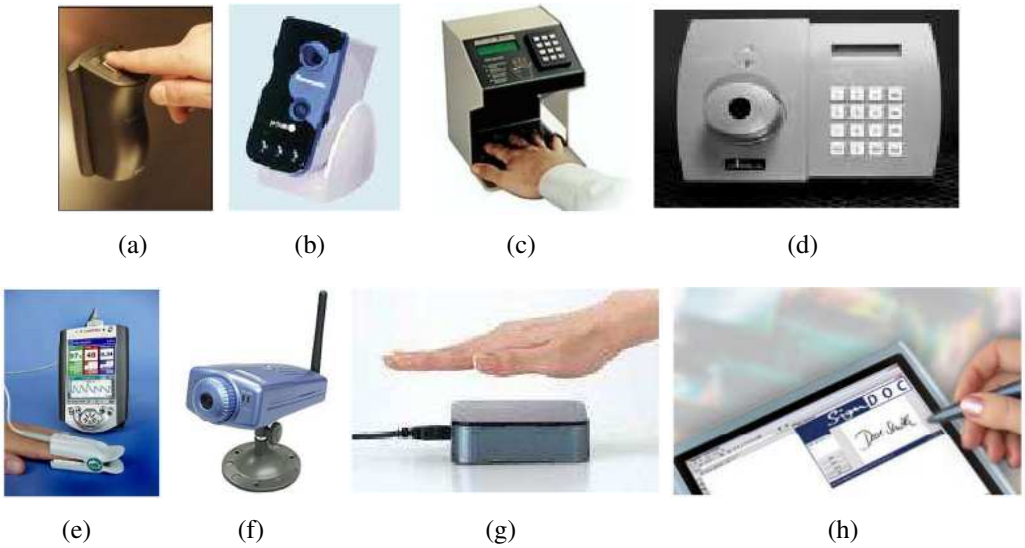


FIGURE 1.5 – Exemples de différents capteurs biométriques concernant : (a) l'empreinte digitale, (b) l'iris, (c) la géométrie de la main, (d) la rétine, (e) la pulsation cardiaque, (f) le visage, (g) la thermographie de la main, et (h) la signature.

à la même personne.

– le dernier critère est le **taux d'égale erreur** (« Equal Error Rate » ou EER). Ce taux est calculé à partir des deux premières quantités et représente traditionnellement un point de mesure de performance. Ce point correspond à l'endroit où FAR = FRR, il représente un compromis entre le nombre de faux acceptés et le nombre de faux rejetés.

La figure 1.6 illustre les taux de faux accepté et de faux rejet à partir des distributions des scores d'un système. Le taux d'égale erreur est illustré à la figure 1.7.

Il existe deux manières de présenter les performances d'un système biométrique selon que l'application soit du type *authentification* ou *identification* :

– Pour une application de type *authentification*, la courbe la plus couramment utilisée est appelée **courbe ROC** (pour « Receiver Operating Characteristic »). Une courbe ROC (voir la figure 1.7) présente le taux de faux rejeté en fonction du taux de faux accepté. C'est une courbe strictement décroissante, qui pour un système performant va avoir tendance à épouser le repère. Le taux d'égale erreur peut être facilement identifiable puisqu'il s'agit de l'intersection de cette courbe avec la droite d'équation $y = x$.

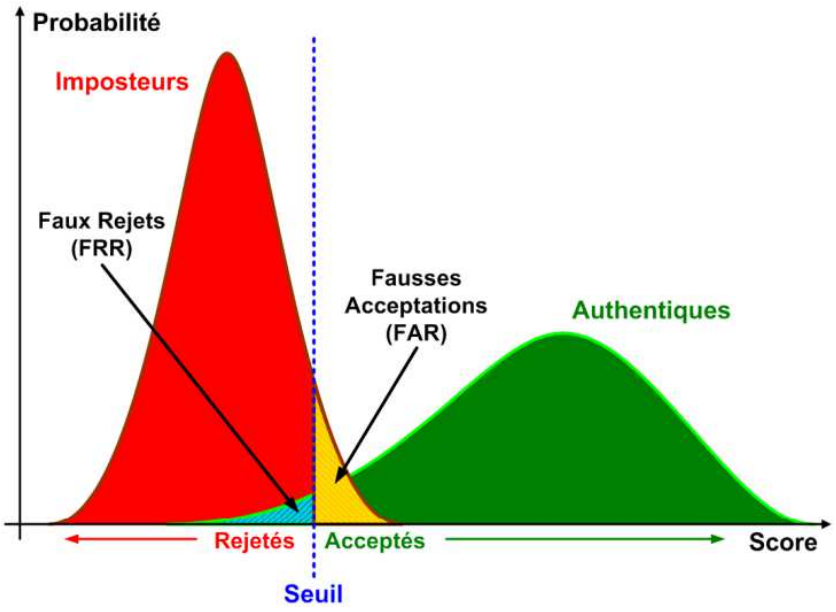


FIGURE 1.6 – Courbes de distribution des imposteurs et des authentiques.

– Pour une application de type *identification*, la courbe la plus utilisée est appelée **courbe CMC** (pour « Cumulative Match Characteristic »). La courbe CMC (voir la figure 1.8) représente le taux de reconnaissance du système en fonction du **rang**. Le rang est une variable définissant à partir de quand l'identification d'un individu est réalisée avec succès. On dit qu'un système reconnaît une personne au rang 0 (aussi appelée rang 1 selon les conventions) si l'individu le plus proche selon le module de similarité correspond bien à l'identité recherchée. S'il s'agit de la deuxième personne la plus proche, elle est alors reconnue au rang 1 (respectivement 2). La courbe CMC est une courbe strictement croissante, dont l'abscisse est comprise entre $[0, n-1]$ où $n-1$ est le nombre d'identités dans la base, et l'ordonnée est comprise entre 0 et 1 (ou entre 0 et 100 selon les conventions). Un des points les plus importants sur ce type de courbe est le taux de reconnaissance pour l'abscisse 0, c'est à dire le nombre de bonnes identifications réussies du premier coup par le système.

1.1.5 Intérêt de la multimodalité

Bien que les techniques biométriques unimodales (c'est à dire utilisant une seule modalité) soient constamment améliorées, leur usage pour des applications très sécurisées n'est pas encore garanti. De plus, ces systèmes peuvent souvent se trouver affectés par les problèmes suivants [149] :

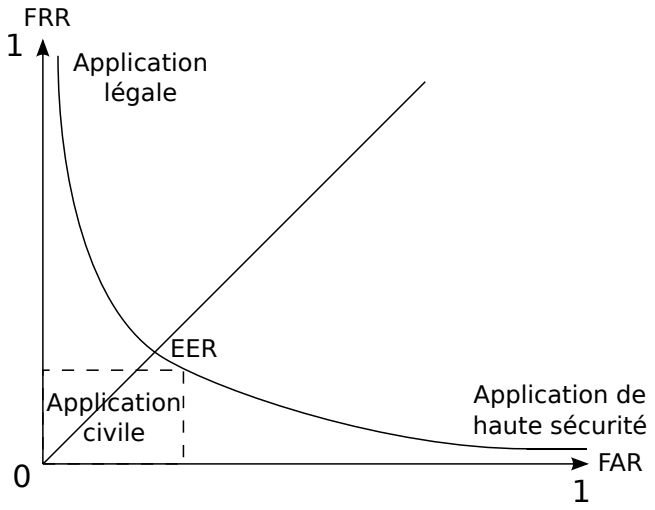


FIGURE 1.7 – Courbe ROC.

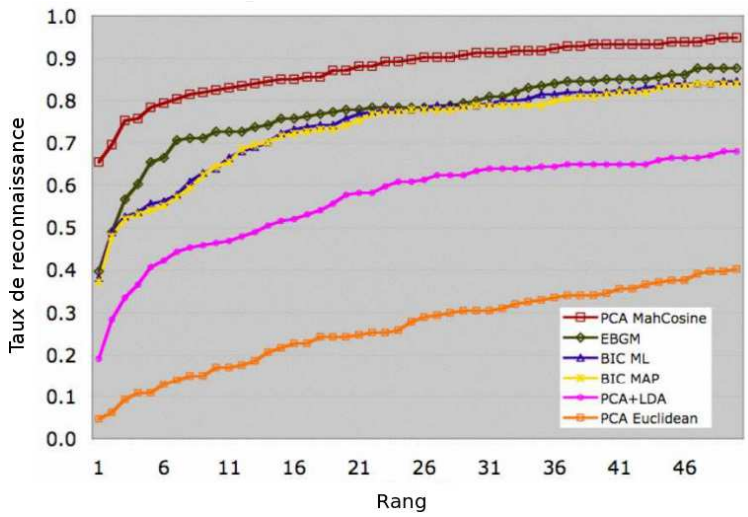


FIGURE 1.8 – Exemple de courbes CMC pour différents algorithmes de reconnaissance faciale.

– Du bruit peut être introduit par un capteur et donc se retrouver dans l'empreinte biométrique d'une personne, ce problème peut venir d'un capteur défaillant par exemple. La qualité de l'échantillon biométrique étant déterminante pour le succès des opérations en aval, la présence de bruit peut sérieusement compromettre les performances du système [64].

– Les modalités biométriques courantes sont censées être universelles. Cependant, il a été montré que toutes ne le sont pas forcément. Ainsi le NIST

(National Institute of Standards and Technologies) a montré qu'environ 2% de la population ne pouvait présenter d'empreinte digitale de bonne qualité. Les raisons peuvent être multiples : malformations, individus effectuant des travaux manuels répétés ce qui a pour effet d'« effacer » l'empreinte. De la même manière, certaines maladies oculaires peuvent rendre les captures d'iris ou de rétine impossibles. Ces problèmes entraînent des erreurs d'enrôlement (« Failure to Enroll ») et/ou des erreurs de capture (« Failure to Capture »).

– Les empreintes biométriques calculées pour un individu peuvent ne pas être suffisamment individuelles. Ainsi, de nombreuses personnes peuvent avoir une apparence faciale suffisamment similaire pour tromper un système biométrique cependant efficace. Citons l'exemple classique de jumeaux, chez qui les visages peuvent être quasiment identiques. Ce manque d'unicité peut entraîner une hausse des taux de fausse acceptation, ce qui peut être inacceptable pour une application hautement sécurisée.

– Certaines modalités biométriques peuvent présenter trop de variance intra–classe. Cette variance exprime le fait que deux empreintes biométriques d'une même personne peuvent avoir une forte dissimilarité. Parmi les facteurs augmentant la variance intra–classe, se trouvent notamment les conditions extérieures qui peuvent modifier grandement l'apparence d'une donnée biométrique, ou encore une mauvaise utilisation du capteur (empreinte non–alignée, visage volontairement tourné, ...) La variance intra–classe peut influer notablement sur les taux de faux rejets.

– Certaines modalités biométriques peuvent être sensibles aux attaques. Ainsi, s'il paraît difficile de recréer la thermographie de la main d'autrui, constituer un masque du visage d'une autre personne est réalisable. De même, des études ont montré qu'il était possible de réaliser de fausses empreintes digitales en récupérant la trace d'une empreinte laissée sur un capteur.

Étant donné ces problèmes, les taux d'erreurs d'un système biométrique unimodal peuvent être rédhibitoires pour une utilisation dont la sécurité est un élément critique. Pour pallier ces problèmes, une solution est l'utilisation de plusieurs modalités pour la reconnaissance. On parle alors de système biométrique multimodal.

Dans cette thèse, le choix a été fait de combiner les modalités visible et infrarouge thermal pour le visage. Alors que la modalité visible pour le visage est une modalité très étudiée mais dont les performances restent toujours moyennes pour des applications réelles, l'ajout de la modalité thermique permet outre l'amélioration intrinsèque des taux de reconnaissance, la diminution du taux de faux acceptés.

Ces deux modalités ont le gros avantage d'être sans contact et bien acceptées par la population (étant donné que le visage est le moyen naturel de reconnaissance entre hommes). Ces deux modalités sont de plus non intrusives, et peuvent être capturées à la volée. Néanmoins, la capture thermale d'un individu peut révéler plus d'informations que sa simple identité. En effet, une trop grande température du corps (en cas de fièvre par exemple) sera visible facilement via la modalité infrarouge thermale, alors qu'elle pourrait rester invisible dans le domaine visible.

Un autre élément important concernant l'utilisation conjointe de ces deux modalités est qu'elles concernent la même partie du corps, permettant ainsi plus de possibilités de fusion que les modalités visage et iris par exemple.

Ces deux modalités présentent cependant leurs challenges respectifs (voir la section 1.2), mais l'utilisation des avantages de l'une peut avantageusement compenser les inconvénients de l'autre.

1.1.6 La modalité infrarouge thermique

Le rayonnement infrarouge est un rayonnement électromagnétique dont la plage de longueurs d'onde se situe au delà du spectre visible par l'œil humain.

Ainsi si la plage communément admise de la lumière visible varie de $400nm$ (couleur violet) à $745nm$ (rouge), le spectre infrarouge peut s'étendre de $745nm$ (proche infrarouge) jusqu'à $100\mu m$.

À l'intérieur de ce spectre, il existe plusieurs sous plages de fréquences auxquelles ont été adjoints les noms de *proche infrarouge*, *infrarouge moyen* et *infrarouge lointain* (voir la figure 1.9). Ces séparations ne sont cependant pas fixes, et certains écarts peuvent exister selon les physiciens. Ces différentes plages d'infrarouge peuvent être considérées comme autant de modalités différentes.

Le proche infrarouge (ou « ShortWave IR ») permet de distinguer encore nettement les traits du visage, et sa texture. Cette modalité est surtout utilisée car elle permet de s'affranchir des problèmes de luminosité. En effet, une capture proche infrarouge d'un objet (et a fortiori d'un visage) sera invariante aux conditions de luminosité.

L'*infrarouge moyen* (« Midwave IR ») et *lointain* (LongWave IR ou LWIR) retranscrit la chaleur émise par les objets. Ces modalités sont également complètement invariantes aux conditions de luminosité. Elles retranscrivent les dégagements de chaleur de l'objet en question ; pour un visage une cartographie thermale est ainsi obtenue. Ces modalités ne sont cependant pas exemptes de défauts concernant la biométrie, voir la section 1.2.

Dans la thèse, nous avons étudié l'infrarouge lointain (grandes longueurs d'ondes) qui retranscrit des zones de chaleur émises par les objets. Cette modalité est peu utilisée dans le contexte de la reconnaissance faciale, bien moins que le proche

infrarouge. Elle permet néanmoins d'apporter de l'information supplémentaire dans le cadre de la fusion de modalités.

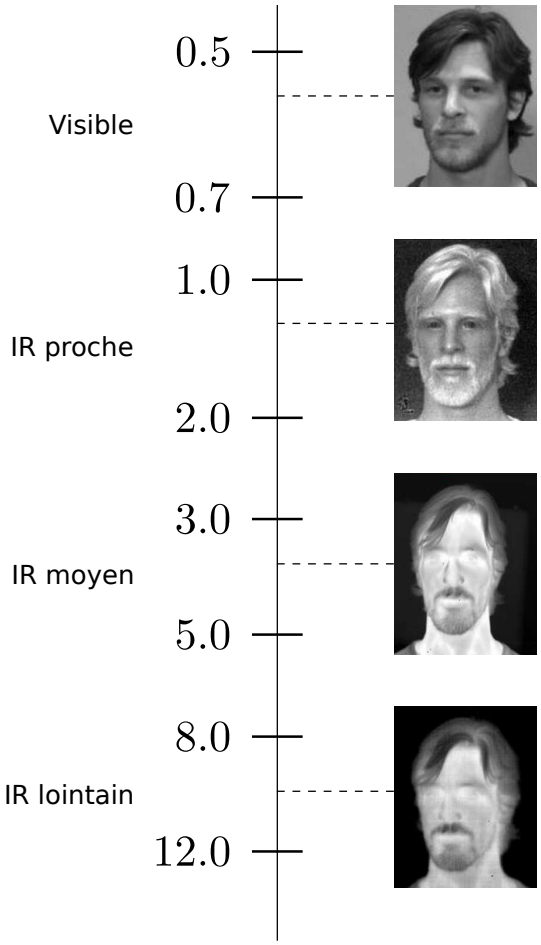


FIGURE 1.9 – Classification des plages infrarouges selon la longueur d'onde (en μm), et le rendu d'une capture faciale associée.

1.2 Difficultés

De nombreuses propriétés du visage ainsi que les conditions dans lesquelles ils ont été capturés rendent le traitement automatique difficile. Dans le cadre de la reconnaissance, le principal problème sous–jacent est la variance intra–classe, c'est à dire la variabilité que peut prendre le visage d'une même personne à cause de différences de luminosité, de pose ... Cette variation intra–classe peut être supérieure

à la variance inter–classe, c'est à dire la variabilité que prennent les visages de différentes personnes. Dans de nombreux systèmes, cette variation intra–classe est considérée comme du bruit (information non désirée) rendant l'objectif de l'application (la reconnaissance) plus difficile. L'extraction de caractéristiques discriminantes est en effet rendue plus compliquée, et les performances globales des systèmes s'en trouvent amoindries.

Nous détaillons ici les principales difficultés rencontrées par un système de reconnaissance faciale automatique dans des conditions réelles.

1.2.1 Illumination

Les variations d'illumination viennent entraîner des variations considérables dans l'apparence d'un visage. Deux types d'éclairage peuvent influencer celle–ci : l'illumination globale (ou ambiante) et l'illumination locale. Alors que l'illumination globale affecte tout le visage de manière uniforme (ou presque), l'illumination locale entraîne la création d'ombre et de zones éclairées et ce de manière non linéaire. La figure 1.2.1 présente un exemple de visage dont la source lumineuse l'éclairant se déplace.

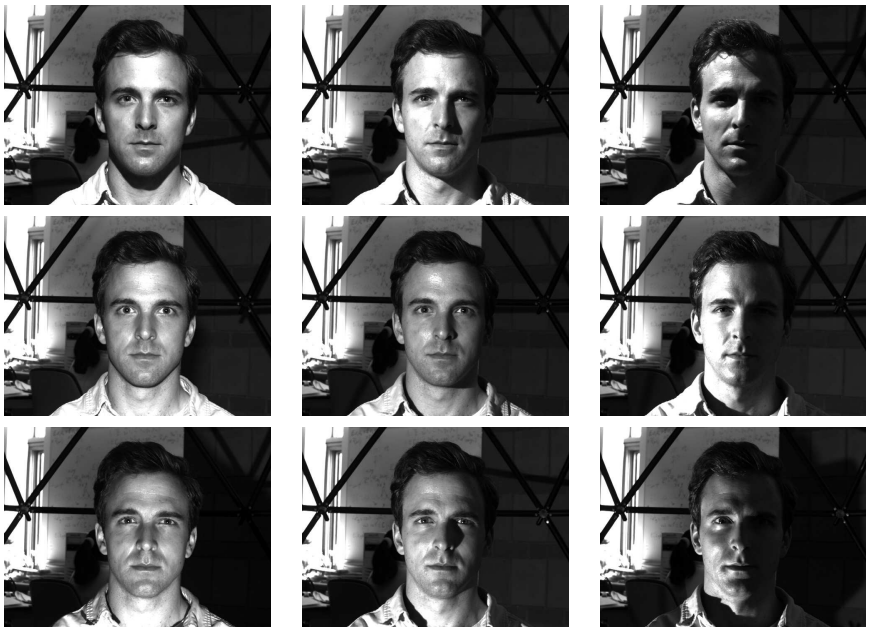


FIGURE 1.10 – Exemple d'un visage d'une même personne subissant un changement de luminosité dont l'angle et l'azimut de la source sont variables.

De nombreuses approches ont été proposées pour gérer ces problèmes de luminosité. Une modélisation implicite de la luminosité dans la création d'un modèle de visage peut être réalisée. L'extraction de caractéristiques invariantes aux changements de luminosité est également une approche largement décrite dans la littérature. Notons enfin qu'un certain nombre d'approches traitent le problème de la luminosité en amont de la reconnaissance par une étape de prétraitement dont l'objectif principal est bien souvent de corriger les artefacts dûs aux variations de luminosité.

1.2.2 Pose

La pose d'un visage définit la rotation qu'a pû subir un visage lors de la capture. Les variations de pose peuvent être de deux types selon le type de rotation : rotation dans le plan où l'axe de rotation est l'axe de la caméra, et rotation hors–plan sinon. La figure 1.11 présente un exemple d'un visage subissant une rotation hors plan. Les variations de pose affectent grandement les systèmes de reconnaissance automatique de visages, c'est pourquoi nombre d'entre eux se limitent aux poses frontales, ou à des poses spécifiques requérant cependant une estimation au préalable. Dans le cas d'une rotation dans le plan, l'apparence du visage n'est pas déformée, et une bonne estimation de l'angle de la rotation peut suffire à recaler l'image par simple rotation inverse, et ainsi obtenir une pose frontale (front en haut de l'image, menton en bas). Le cas de la rotation hors plan est souvent bien plus complexe, sauf si les visages utilisés pour l'enrôlement et la reconnaissance présentent la même pose.

1.2.3 Expressions faciales

L'apparence d'un visage varie grandement en présence d'expressions faciales (Figure 1.12). Les éléments faciaux tels que la bouche ou encore les yeux peuvent alors subir des déformations importantes, pouvant faire échouer un système de reconnaissance faciale fondée par exemple sur des points d'intérêt (ceux–ci pouvant ainsi subir d'importantes translations). La bouche est en général l'élément facial qui varie le plus, mais l'aspect des sourcils peut par exemple être grandement modifié.

1.2.4 Occlusions

Les occlusions partielles apparaissent fréquemment dans des applications réelles, comme illustré sur la figure 1.13. Elles peuvent être causées par une main cachant une partie du visage, par des cheveux longs, des lunettes de vue, de soleil, par tout autre objet (foulard ...), ou encore par une autre personne. Il arrive également qu'une partie du visage en cache une autre, comme dans le cas d'une rotation hors plan par exemple.

FIGURE 1.11 – Exemple d'un visage d'une même personne subissant des variations de pose (hors plan).

FIGURE 1.12 – Variabilité intra–classe dûe à la présence d'expressions faciales.

1.2.5 Température du corps

Les variations de température du corps peuvent altérer grandement le rendu d'un visage capturé dans la modalité infrarouge (grandes longueurs d'onde). Cette modalité reflétant la chaleur émise par les objets, certaines parties du visage (nez ou oreille notamment) peuvent être à des températures différentes, leur rendu peut donc être très différent d'une capture à l'autre. La figure 1.14 présente un exemple d'une même personne dont les captures ont été réalisées à différents moments.

1.2.6 Autres difficultés

D'autres types de difficultés peuvent apparaître pour un système automatique de reconnaissance faciale. Ainsi, des variations peuvent être dûes à la présence de maquillage, d'opérations chirurgicales, de différentes coupes de cheveux, ou encore

FIGURE 1.13 – Variabilité intra–classe dûe à la présence d'occlusions partielles.

FIGURE 1.14 – Variabilité intra–classe dûe à des variations de la température du corps.

la présence (absence) de moustaches, de barbes, ... Un autre point très important est *l'âge* des captures, c'est à dire le moment à laquelle les captures ont été réalisées. En effet, l'apparence d'un visage peut changer au cours du temps (notamment lors de l'adolescence), et un écart de temps important entre deux captures peut engendrer des difficultés de reconnaissance. Cet effet n'est pas nouveau mais la constitution de bases de données s'étalant sur plusieurs années est difficile.

1.3 Principales Bases de Données de Visages

De nombreuses bases de données de visages (publiques ou privées) existent à des fins de recherche. Elles peuvent différer entre elles sur plusieurs points :

- le nombre d'images disponibles est probablement le critère le plus important d'une base de données,
- le nombre d'images disponibles par personne,

- la modalité (2D–visible, 2D–infrarouge, 3D, couleur, niveaux de gris, maillage, carte de profondeur, ...)
- la taille des images,
- les poses et orientations des visages,
- la variation de l'illumination,
- le sexe des personnes présentes,
- la présence d'artefacts (lunettes, barbes, ...),
- la présence d'images statiques ou de vidéos,
- la présence d'un fond uniforme,
- la période entre les prises de vues.

Il est ainsi recommandé de bien choisir la base de données lors des tests d'un algorithme. En effet, certaines sont dotées d'un protocole bien défini permettant ainsi la comparaison directe des résultats. De plus, le choix doit dépendre du problème que l'on souhaite tester : illumination, reconnaissance à travers le temps, expressions faciales .... La disponibilité de nombreuses images différentes par personne peut également être un argument décisif pour la bonne réalisation d'un algorithme.

Les bases de données de visages peuvent être classées en trois catégories selon l'objectif recherché : reconnaissance, détection de visages ou analyse des expressions faciales. Les principales bases de données utilisées dans le domaine de l'analyse de visages sont détaillées en annexe C.

Dans la suite du manuscrit, nous utilisons principalement la base de données Notre–Dame de l'université de Notre Dame (Indiana, USA) car elle présente plusieurs avantages : cette base de données est publique et disponible sur simple demande, elle comporte des images acquises pour les modalités visible, infrarouge (grandes longueurs d'ondes ou thermique) et 3D, et elle dispose d'un protocole bien défini permettant, lorsque celui–ci est respecté, une comparaison aisée des méthodes.

1.4 Chaîne de Traitement

La chaîne complète d'un système de reconnaissance automatique de visages peut se décomposer en deux principaux modules (Figure 1.15) :

- Le module de *détection/normalisation* chargé de détecter le ou les visages présents dans une image ou une vidéo. Ce module peut également être chargé du suivi de visages précédemment détectés. Ce module se charge également une fois le(s) visage(s) détecté(s) d'estimer leur pose, ainsi que leur taille, de manière à pouvoir les normaliser, c'est à dire les transformer géométriquement par rotation, mise à l'échelle, pour qu'ils soient normalisés correctement pour le module de reconnaissance.

– Le module de *reconnaissance* peut se décomposer en trois étapes : la première consistant en un prétraitement des images de visages afin de les améliorer pour l'étape suivante. C'est à cette étape que des corrections éventuelles de luminosité ou le retrait d'artefacts a lieu. La seconde étape est l'extraction de l'empreinte biométrique du visage. Les algorithmes classiques retournent généralement un vecteur de caractéristiques. La dernière étape se charge de comparer l'empreinte biométrique testée à une base de données d'empreintes biométriques dont les identités associées sont connues. La base de données dans laquelle le système va chercher la correspondance est généralement appelée *galerie*.

Notons que pour chacun des modules et sous–modules, de nombreux travaux ont été réalisés. Nous pouvons néanmoins identifier les deux modules les plus sensibles : le module de détection de visages ainsi que le module d'extraction de caractéristiques.

Citons pour le premier les travaux de Viola et Jones [285] utilisant les caractéristiques de Haar et une cascade de classifieurs faibles. Citons également les travaux de Garcia et Delakis utilisant une approche fondée sur les réseaux de neurones convolutionnels [115].

Le second module sensible selon nous est le module d'extraction de caractéristiques discriminantes, invariantes, et rapides à comparer. Ce module est plus largement détaillé dans la suite de la thèse.

1.5 Plan

Le manuscrit se décompose en trois parties ;

– la première partie introduit la biométrie (ce chapitre) et dresse un état de l'art des techniques utilisées pour la reconnaissance de visages. Nous insistons notamment sur la réduction de dimension (voir l'annexe A pour une revue des enjeux et des principales méthodes de réduction de dimension),

– la seconde partie explore les techniques de représentation de visages utilisées tout au long de la thèse,

– la troisième partie détaille les résultats expérimentaux obtenus à l'aide des méthodes mises en œuvre, ainsi que les résultats de la fusion des modalités visible et infrarouge thermale.

Plus précisément :

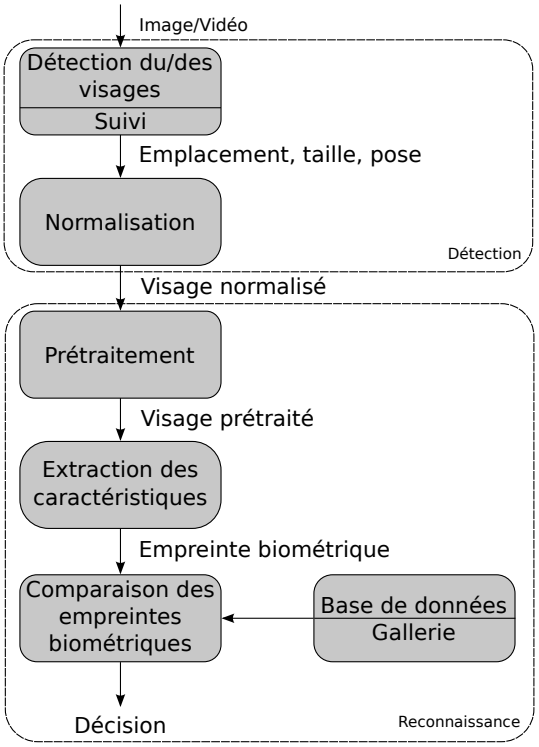


FIGURE 1.15 – Chaîne de traitement d'un système de reconnaissance faciale.

- Le chapitre 2 présente un état de l'art des techniques de reconnaissance de visages selon qu'elles soient locales, globales ou hybrides, en insistant particulièrement sur les techniques de réduction de dimension, dépassant du cadre de l'étude de la biométrie.

- Le chapitre 3 décrit en détail les réseaux de neurones convolutionnels, leur architecture, l'apprentissage de tels réseaux, leur utilisation ainsi que les optimisations possibles.

- Le chapitre 4 concerne les techniques parcimonieuses mises en œuvre durant la thèse, de la décomposition d'un signal sur un dictionnaire jusqu'à l'apprentissage de ces dictionnaires. Une méthode de classification fondée sur la parcimonie est également présentée.

- Le chapitre 5 détaille les résultats expérimentaux obtenus à l'aide des réseaux de neurones convolutionnels, leur variante où les couches ont bénéficié d'un préapprentissage, ainsi que par les méthodes parcimonieuses.

- Le chapitre 6 décrit les différents niveaux de fusion considérés pendant la thèse ainsi que les résultats expérimentaux obtenus.

Enfin, nous dresserons une conclusion globale sur ces travaux et présenterons les perspectives futures.

Chapitre 2

Les principales techniques de reconnaissance faciale

La reconnaissance automatique de visages est un sujet central dans la recherche sur l'analyse de visages. Le nombre d'applications possibles fait qu'il a pris beaucoup d'importance depuis plusieurs années. Deux types d'application peuvent être différenciées : celles dites du *monde ouvert* et celles dites du *monde fermé*. Les applications dites du *monde fermé* ne traitent qu'avec un nombre limité de personnes *connues*, tandis que les applications dites du *monde ouvert* peuvent traiter avec des personnes inconnues. Rappelons également qu'un système biométrique peut être utilisé pour une *authentification* (ou *vérification*) ou une *identification*. Le système en identification doit trouver l'identité de l'individu présenté au système et essaie donc de répondre à la question *Qui suis–je ?* Le système en authentification reçoit une identité et doit prendre la décision si oui ou non l'image correspond à l'identité, répondant ainsi à la question *Suis–je bien la personne que je prétend être ?* Dans les deux cas, le problème revient cependant à un problème de classification.

Dans ce chapitre, nous décrivons brièvement quelques techniques parmi les plus importantes ou les plus populaires utilisées en reconnaissance de visages (voir les résumés de l'état de l'art [120] et [308], ou le livre [76] pour plus de détails). Les approches existantes peuvent être grossièrement divisées en trois groupes : les approches *locales*, les approches *globales* ainsi que les approches *hybrides*. Les principales méthodes de ces trois approches sont décrites dans la suite.

Nous insistons notamment sur les méthodes de réduction de dimension faisant partie des approches globales. Les méthodes de réduction de dimension entrent en effet dans le cadre plus vaste des traitements de données en général (et pas seulement des visages) et de la reconnaissance d'objets.

Notons que la plupart des méthodes nécessitent une localisation précise du visage, un recadrage géométrique (pour que celui–ci apparaisse toujours dans la même position), ainsi qu'une mise à l'échelle pour que le visage ait une taille adéquate.

2.1 Approches locales

Les approches locales de la reconnaissance de visages sont basées sur des modèles et reposent sur un traitement séparé des différentes régions de l'image du visage. Les modèles utilisés reposent sur les connaissances que l'on possède à priori de la morphologie des visages. La plupart du temps, cela implique la détection/extraction de caractéristiques faciales locales.

Brunelli et Poggio [47] proposent une technique qui extraie automatiquement un ensemble de 35 caractéristiques géométriques d'une image de visage (voir la figure 2.1). Ces ensembles de caractéristiques sont ensuite comparés deux à deux via la distance de Mahalanobis pour réaliser la reconnaissance.

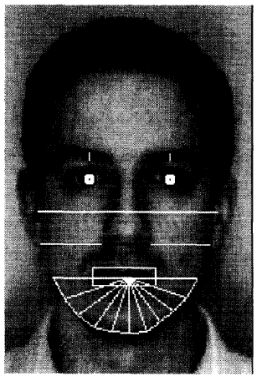


FIGURE 2.1 – Localisation des caractéristiques géométriques utilisées dans [47].

Une autre approche géométrique a été proposée par Takàcs [270]. Des cartes binaires de contour sont extraites des images de visage via un filtre de Sobel. La similarité entre deux contours est ensuite calculée en utilisant une variante de la distance de Hausdorff. Cette approche a été étendue par Gao *et al.* [114] qui ont transformé les cartes de contours en cartes de lignes de contours (ou LEM pour *Line Edge Maps*) contenant des listes de segments (voir la figure 2.2). La distance utilisée pour mesurer la similarité est la même que celle de Takàcs.

L'approche de Heisele *et al.* [132] commence par détecter la région contenant le visage dont dix points caractéristiques sont extraits. Les zones autour de ces points sont ensuite extraites (voir la figure 2.3), et concaténées pour former le vecteur caractéristique du visage. La classification est finalement réalisée grâce à l'utilisation d'une machine à vecteurs de support (ou SVM pour « Support Vector Machine »).

Price et Gee proposent également une méthode [235] se basant sur des zones extraites du visage. Ici, trois régions sont considérées : une bande rectangulaire comprenant les yeux et le nez, une deuxième bande rectangulaire ne contenant que les

FIGURE 2.2 – Cartes de contours utilisées dans [114].

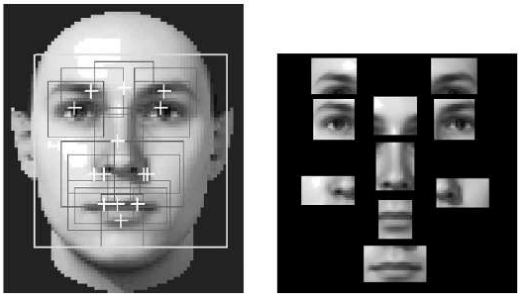


FIGURE 2.3 – Approche proposée dans [132].

yeux, et enfin une région contenant le visage entier. Une analyse linéaire discriminante (voir l'annexe A.2.4) est ensuite appliquée à chacune de ces régions (voir la figure 2.4).

Samaria *et al.* [250] présentent une approche basée sur les chaînes de Markov cachées (HMM pour *Hidden Markov Models*). Le visage est segmenté en sous–bandes partiellement recouvrantes, ces sous–bandes étant ensuite concaténées en un vecteur de grande taille ou compressées par DCT. Puis, pour chaque classe (individu), un HMM est créé modélisant la distribution probabiliste des sous–bandes. Les images de visages sont finalement classées en appliquant l'algorithme de Viterbi pour comparer la séquence des sous–bandes de l'image avec les modèles appris. Cette approche a été étendue aux 2D–HMM [217].

Perronnin *et al.* proposent dans [233] une approche basée sur le modèle des 2D–HMM où les expressions faciales et l'illumination sont modélisées indépendamment.

Les approches bayésiennes ont également été explorées via les travaux de Liu et Wechsler [191], où est proposé un cadre bayésien unifiant les méthodes les plus populaires de reconnaissance de visages.

Les machines à vecteur de support (SVM) sont également utilisées pour la reconnaissance faciale par Guo *et al.* dans [121]. Deux bases y sont utilisés, la base

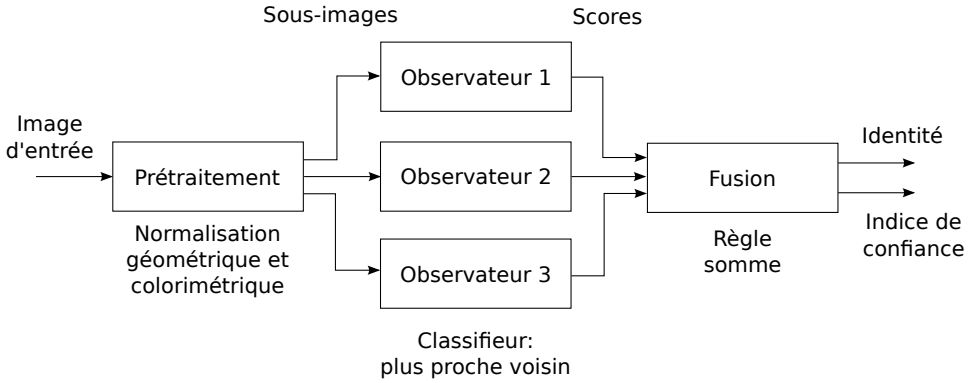


FIGURE 2.4 – Approche de Price et Gee [235].

AT&T ainsi qu'une base « maison ». L'approche est comparée à l'ACP.

Les modèles actifs d'apparence (ou AAM pour « Active Appearance Models ») sont présentés par Cootes *et al.* dans [69]. Ils consistent en la création d'un modèle statistique d'un visage, voir la figure 2.5. Le modèle est ensuite déformé pour « coller » au plus près des traits du visage. La reconnaissance est effectuée sur le résidu calculé correspondant à l'erreur de prédiction du modèle.

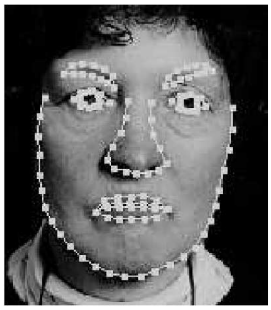


FIGURE 2.5 – Modèle actif d'apparence.

Les LBP (pour « Local Binary Patterns ») ont également été utilisés pour la reconnaissance faciale, notamment dans [26]. Le visage est subdivisé en sous–régions carrées de taille égale sur lesquelles sont calculées les caractéristiques LBP. Les vecteurs obtenus sont ensuite concaténés pour obtenir le vecteur de caractéristiques final. Des extensions des LBP comme les MB–LBP (pour « Multi-Scale Block Binary Pattern ») ont été proposées et appliquées aux visages par Liao *et al.* [188] (voir la figure 2.6).

Le gros avantage des méthodes locales de reconnaissance de visages est qu'elles peuvent modéliser facilement les variations de pose, d'illumination ou encore d'expressions que peut subir un visage. Cependant, elles nécessitent souvent le place-

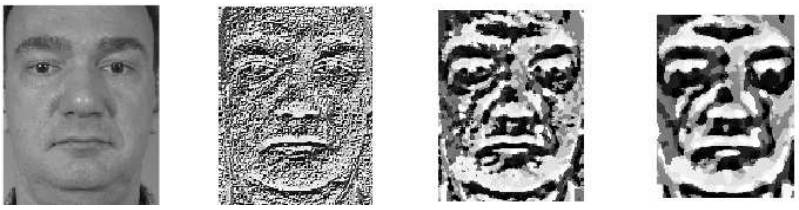


FIGURE 2.6 – Caractéristiques (MB)LBP pour un visage, respectivement pour un masque de taille 3×3, 9×9 et 15×15.

ment manuel de nombreux points d'intérêts pour une bonne précision, et sont donc lourdes à mettre en œuvre.

2.2 Approches globales

Les approches globales prennent l'image du visage comme un tout et utilisent des techniques d'analyse statistique bien connues. L'idée est généralement de projeter l'image d'entrée du visage, préalablement vectorisée, dans un espace de plus faible dimension, où la reconnaissance est supposée être plus aisée. La projection est souvent conçue pour ne sélectionner que les caractéristiques importantes et suffisamment discriminantes pour différentier les personnes entre elles.

Un des avantages des méthodes globales est qu'elles sont rapides à mettre en œuvre, les calculs reposant sur des opérations matricielles relativement simples. Cependant, étant donné qu'elles considèrent le visage comme un tout, elles sont sensibles aux conditions de luminosité, de pose ou encore d'expression faciale.

La plupart de ces méthodes réalisent une analyse de sous–espaces de visage (ou de la variété définie par les visages). Cette analyse découle de la constatation d'un fait relativement simple : la classe des visages réside dans un sous–espace de l'espace de l'image d'entrée. Prenons par exemple une image de taille 100×100 en niveaux de gris. Le nombre de configurations possibles est égal à 256^{10000}. Cependant, parmi toutes ces configurations possibles, seule une petite partie correspond aux visages. L'information contenue dans les images de visages est donc très redondante, la dimension de ces images peut donc être réduite en se concentrant uniquement sur ce qui nous intéresse (les visages). Le sous–espace est souvent appelé *espace de visages* (ou « *facespace* »).

Les méthodes globales peuvent se décomposer en deux types de techniques : les techniques *linéaires* et les techniques *non linéaires*.

2.2.1 Techniques linéaires

Les techniques linéaires réalisent une projection linéaire des visages (espace dont la dimension est égale à la dimension des images, donc grande) sur un espace de plus faible dimension. Cependant, ces techniques linéaires sont sensibles aux conditions de luminosité notamment, et plus généralement aux variations non convexes. Ainsi, l'utilisation de distances classiques dans l'espace projeté ne permet pas toujours de réaliser une bonne classification entre les classes « visages » et « non visages ».

La plus connue de ces approches est la technique dite des *Eigenfaces* présentée par Turk et Pentland dans [282]. Une ACP est réalisée sur un ensemble d'apprentissage d'images de visages. Les principaux vecteurs propres résultant de l'ACP définissent le nouvel espace. Les images de visages sont ensuite projetés sur cet espace, et les vecteurs obtenus sont utilisés pour la classification.

De nombreux travaux ont été réalisés sur le choix des vecteurs propres à retenir pour définir le nouvel espace. Ainsi, Kirby *et al.* [158] proposent un critère basé sur l'énergie des valeurs propres associées aux vecteurs propres. Les vecteurs propres correspondant aux plus grandes valeurs propres sont retenus jusqu'à ce que la somme des valeurs propres dépasse un certain seuil de l'énergie totale (90% dans [158]). Martinez *et al.* montrent dans [200] que les taux de reconnaissance peuvent être améliorés en ignorant les premiers vecteurs propres (ceux dont les valeurs propres associées sont les plus grandes), ceux–ci encodant souvent les variations d'illumination. Voir l'annexe A.2.1 pour une description plus complète de l'ACP ainsi que pour sa dérivation.

Une autre approche bien connue présentée par Belhumeur *et al.* [165] réalise une Analyse Discriminante Linéaire (LDA), elle est ainsi souvent nommée *Fisherfaces*. En effet, cette technique consiste à maximiser sur un ensemble d'apprentissage le critère de Fisher, à savoir le quotient de la variance inter–classe par la variance intra–classe. Ainsi, contrairement à la technique des *Eigenfaces* où la meilleure représentation (celle maximisant la variance) est recherchée, le but est ici une meilleure séparation des classes. Cependant, étant donné que le nombre d'images est souvent inférieur à leur dimension, la matrice de variance intra–classe peut être singulière, et son inversion pose donc problème. Ce problème est connu sous le nom de *Small sample size problem*. Des méthodes ont été proposées pour contourner ce problème, la plus utilisée étant de réaliser une ACP au préalable pour diminuer la dimension des échantillons. Voir l'annexe A.2.4 pour une description plus complète de l'analyse discriminante linéaire.

De nombreuses variantes à ces méthodes linéaires ont été proposées dans la

littérature [245], [277], [207], [269], [307].

D'autres techniques linéaires ont également été utilisées pour le calcul de vecteurs caractéristiques :

- l'analyse en composantes indépendantes (ICA) dans [30] (voir annexe A.2.3),
- la factorisation de matrices non négatives (NMF) dans [49] ou [290],
- l'analyse discriminante bilinéaire (BDA) dans [286],
- la technique dite de « Vecteurs communs discriminants » (DCV) dans [59].

Certaines méthodes proposées ne reposent pas sur un seul sous–espace, mais sur plusieurs, chacun étant caractéristique à une variation [279], [301], [278], [297]. Par exemple, Pentland *et al.* [231] calculent un sous–espace pour chaque orientation et chaque échelle d'un visage ainsi qu'autour de certaines caractéristiques détectées (voir la figure 2.7). Un nouveau visage est ensuite identifié en le projetant sur tous les sous–espaces et en sélectionnant celui étant le plus proche d'un vecteur de la galerie.

Cette technique reposant sur plusieurs sous–espaces est généralisée dans [284] où des tenseurs à quatre dimensions correspondant à la classe, la pose, les conditions d'illumination et l'expression faciale sont calculés pour une base d'apprentissage donnée et permettent ainsi une meilleure robustesse de la classification.

FIGURE 2.7 – Caractéristiques autour desquelles est réalisée une ACP dans [231].

Bien que ces méthodes linéaires soient assez efficaces, elles manquent de précision dès lors que les images de visages subissent des transformations non linéaires. Une simple modification de la luminosité transforme celui–ci de façon non linéaire étant donné la complexité de la forme.

2.2.2 Techniques non linéaires

Des techniques globales non linéaires ont été développées, souvent à partir des techniques linéaires. Ainsi l'Analyse en Composantes Principales à Noyaux (ou « *Kernel–PCA* ») [255], [139] et l'Analyse Discriminante Linéaire à Noyaux (ou « *Kernel–LDA* ») [205] utilisent la notion mathématique des noyaux pour étendre les techniques linéaires que sont l'ACP et la LDA (voir annexe A.3.1).

D'autres techniques non linéaires ont également été utilisées dans le contexte de la reconnaissance faciale :

- le *MultiDimensional Scaling* (MDS) dans [156] ou [42],
- l'Isomap dans [302],
- les *diffusion maps* dans [123],
- le *Local Linear Embedding* (LLE) dans [288] ou [226],
- les *Laplacian eigenmaps* dans [129], [227] ou [237],
- le *Hessian LLE* dans [156],
- le *Local Tangent Space Analysis* (LTSA) dans [289],
- les approches neuronales dans [273] ou [108] (autoencodeurs), dans [176] (cartes de Kohonen), et dans [90] (réseaux de neurones convolutionnels).

L'utilisation de ces méthodes de projection de l'espace des images sur l'espace de caractéristiques est non linéaire et permet ainsi dans une certaine mesure de réduire la dimension des images de meilleure façon. Cependant, bien que ces méthodes permettent souvent l'amélioration des taux de reconnaissance sur des jeux de tests donnés, elles sont trop flexibles pour être robustes à de nouvelles données, contrairement aux méthodes linéaires.

2.3 Approches hybrides

Les méthodes hybrides résultent de l'association des méthodes locales et des méthodes globales. Elles combinent la détection de caractéristiques locales avec l'extraction de caractéristiques globales. Ces techniques essaient finalement de tirer partie des avantages des deux types de méthodes citées plus haut.

L'approche appelée Analyse en Composantes Locales (LCA pour *Local Component Analysis*) a été proposée par Penev et Atick [229]. Plusieurs analyses en composantes principales sont réalisées pour extraire différentes caractéristiques locales (voir la figure 2.8). Celles–ci sont ensuite combinées et une procédure minimisant l'erreur de reconstruction avec une contrainte parcimonieuse permet de réaliser la reconnaissance.

L'approche dite de l'Elastic Bunch Graph Matching (EBGM) a été proposée par Wiskott *et al.* [295]. Les visages sont représentés par des Face Bunch Graph (FBG),

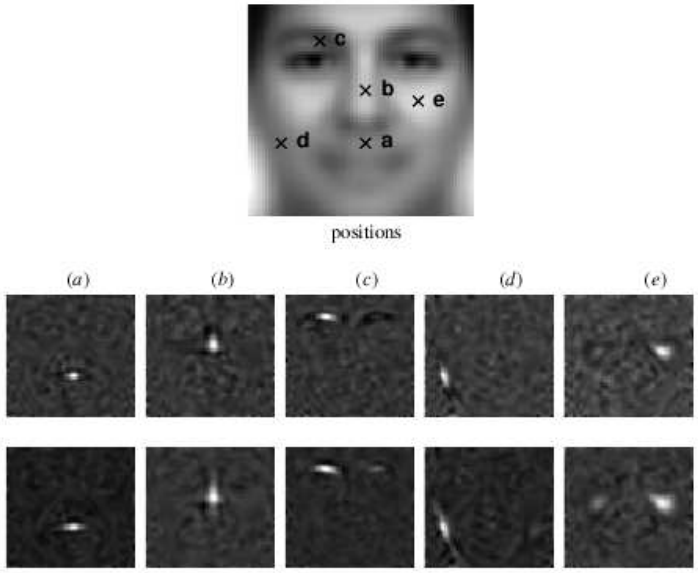


FIGURE 2.8 – Approche *Local Component Analysis*.

où chaque noeud du graphe correspond à une certaine caractéristique faciale (l'oeil droit ou gauche par exemple), voir la figure 2.9. A chaque noeud du graphe est associée l'apparence du voisinage de la caractéristique via un *jet*. Les jets représentent l'ensemble des 40 coefficients issus de la convolution du voisinage de la caractéristique par un filtre de Gabor spécifique (voir la figure 2.10). Les arêtes du graphe sont pondérées par la distance relative des caractéristiques adjacentes. Une fois que le graphe est créé pour chaque personne de la base d'apprentissage, un algorithme spécifique de mise en correspondance permet d'identifier une personne inconnue. L'algorithme essaie itérativement de faire correspondre le graphe créé à chaque graphe de la base d'apprentissage en minimisant une fonction de coût prenant en compte à la fois une mesure de similarité géométrique ainsi qu'une mesure de similarité de l'apparence modélisée via les jets.

Perlibakas présente plus récemment l'algorithme LogGabor PCA dans [232]. Une convolution par des ondelettes de Gabor orientées est réalisée autour de certains points caractéristiques du visage. Les vecteurs ainsi créés contiennent à la fois la localisation ainsi que les amplitudes des énergies locales. Une Analyse en Composantes Principales est ensuite réalisée afin de réduire la dimension de ces vecteurs.

Pentland *et al.* présentent dans [231] l'approche dite des espaces propres modulaires (Modular Eigenspaces) Cette technique réalise une Analyse en Composantes Principales et une classification sur des régions distinctes du visage, comme les yeux, le nez, la bouche ou encore le visage entier. La zone de la bouche subit de grosses déformations dues aux expressions faciales, ainsi l'ajout de cette région au

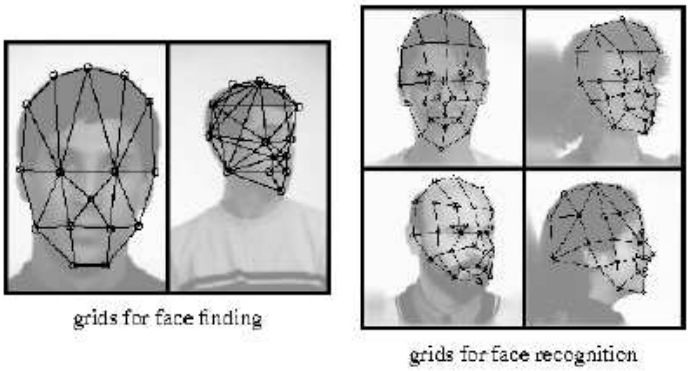


FIGURE 2.9 – Graphe appliqué aux visages pour l'approche EBGM.

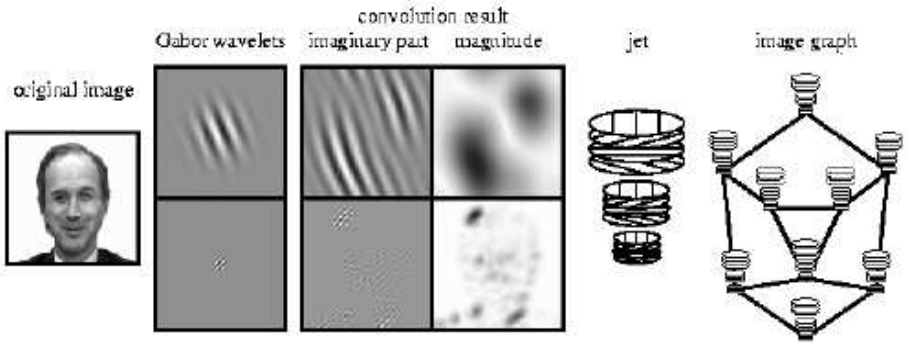


FIGURE 2.10 – Création des *jets* pour l'approche EBGM.

processus entier fait décroître les taux de reconnaissance.

L'approche proposée par Cootes *et al.* [70] met en œuvre les Modèles Actifs d'Apparence (AAM pour *Active Appareance Models*). Cette méthode modélise indépendamment la forme et la texture d'un visage en appliquant une Analyse en Composantes Principales. Les vecteurs obtenus pour la forme et la texture sont ensuite utilisés pour la reconnaissance. Un nouveau visage qui doit être identifié est adapté au modèle par un processus d'optimisation itératif. Les paramètres de forme et de texture obtenus sont ensuite comparés à ceux de la base. Latinis *et al.* [175] appliquent cette méthode pour la première fois à la reconnaissance de visages. Edwards *et al.* [93] proposent des méthodes basées sur les AAMs pour la reconnaissance faciale.

2.4 Conclusion

Nous avons présenté dans ce chapitre les principales approches utilisées dans la littérature pour la reconnaissance faciale automatique. Les méthodes peuvent principalement se classer en deux catégories : les métodes globales et les méthodes locales (les méthodes hybrides essayant de faire un lien entre ces types d'approches).

Les méthodes globales présentent un certain nombre d'avantages :
- Le problème de la reconnaissance faciale automatique est transformé en un problème d'analyse de sous–espaces de visages, pour lequel de nombreuses méthodes statistiques existent.
- Les méthodes globales sont souvent applicables à des images basse résolution ou de mauvaises qualités.

Certains inconvénients se posent cependant avec les méthodes globales :
- Il est nécessaire de disposer de suffisamment de données représentatives des visages.
- Il n'y a pas d'*a priori* sur le physique d'un visage.
- Ces méthodes ne sont robustes qu'à des variations limitées (pose, illumination, expression).

De la même manière les méthodes locales présentent certains avantages :
- Le modèle créé possède des relations intrinsèques bien définies avec les visages réels.
- Les modèles créés peuvent prendre en compte explicitement les variations telles que la pose, l'illumination ou les expressions. La reconnaissance est ainsi plus efficace dans le cas de fortes variations.
- La connaissance *a priori* sur les visages peut être intégrée aux modèles afin d'améliorer leur efficacité.

Les méthodes locales présentent cependant quelques inconvénients :
- La construction du modèle, reposant souvent sur la détection de points caractéristiques faciaux, peut être laborieuse.
- L'extraction des points caractéristiques peut être difficile dans le cas de variations de pose, d'illumination, d'occlusion ...
- Les images doivent être de relativement bonne qualité, et/ou être de résolution suffisante afin de pouvoir extraire les points caractéristiques.

Le tableau 2.4 résume quelques résultats obtenus via des algorithmes classiques de reconnaissance de visages sur certaines bases de données de la littérature. Il met également en exergue la difficulté de comparer les résultats entre les différentes

approches. En effet, de nombreux paramètres doivent être pris en compte afin de bien appréhender l'efficacité d'une méthode : la base de données utilisée, la taille des images, la présence de variations dans les échantillons ainsi que le nombre d'images utilisées pour l'enrôlement et/ou les tests.

Afin de comparer les algorithmes, certains concours ont eu lieu sur des bases de données bien définies, munies de protocoles clairs. Ainsi, le challenge FRGC [234] a permis la comparaison de nombreuses méthodes issues d'entreprises ou de laboratoires internationaux.

Réf.	Méthode	Base de données	Taille des images	Nb. Images	*Time lapse*	Taux (%)	Expr.	Ill.	Pose
[200]	PCA	AR	85×60	100–250	N	70		N	N
	LDA	AR	85×60	100–250	N	88		N	N
[164]	Fisherfaces	YALE		144–16	N	99.6	O	O	N
[303]	Direct LDA	ORL	112×92	200–200	N	90.8	O	O	O
[194]	DF–LDA	ORL	112×92	200–200	O	96		O	N
		UMIST	112×92	160–415	N	98		N	N
[59]	DCV	Yale	126×152	15–150	N	97.33		O	N
		AR	229×299	350–350	O	99.35			
[30]	ICA	FERET	60×50	425–421	O	89	O	N	N
[190]	PDBNN	SCR	80×20	320–1280	N	100	O	O	O
		FERET		200–200	N	99	O	O	N
		ORL			N	96		O	O
[98]	RBF	ORL	160×120	300–300		98.1	O		O
[171]	HMM	FERET	128×128	500–500	N	97	O	N	N
[192]	Gabor EFM	FERET	128×128	200–100	N	99	O	N	N
		ORL	128×128	200–200	N	100	O	N	O
[296]	EBGM	FERET	256×384	250–250	N	80	O		O
[116]	WPA	MIT	480×640	155–155		80.5	O	O	
		FERET	256×384	200–400		89			
[271]	IFS	ORL	112×92	200–	N	95			
[92]	IFS	MIT	480×640	90–90		90			O
[63]	PCA	UND		166–166	N	98	O	O	N
[266]	PCA	Equinox	99×132	770–2310	O	93	O	O	N
[50]	Th–Spectrum	Equinox		225–2500		86.8	O		O
[280]	Hyperspectral	Propriétaire		200–1200	O	92	N	O	N
[114]	LEM	Bern		40–160	N	72.09	O		O
		AR		112–336		86.03		O	N
		Yale		15–150		85.45		O	N
[157]	ICA	AR, Yale, ORL, Bern, FERET	46×56	1685–1490		98		O	O
[186]	LDA/GSVD	CMU_PIE		68–1360	N	99.53		O	N
	LDA/QR	YaleB/Pose00		80–432	N	98.03		O	N
[117]	Cone Models	YaleB	36×42	450–4050	N	97	N	O	O
[220]	Sous–espaces	ATR		2821–804	N	98.7	N	N	O
[119]	*EigenLights*	CMU–PIE		5304–5304	N	36	N	O	O

TABLE 2.1 – Comparatif de plusieurs méthodes de l'état de l'art. La colonne *Nb. Images* indique le nombre d'images utilisées pour l'enrôlement et le nombre d'images utilisées pour les tests, la colonne *Time lapse* indique si les images d'enrolement et de tests ont été capturées avec un intervalle significatif, les colonnes *Expr.*, *Ill.* et *Pose* indiquent si les images possèdent des variations d'expression faciale, d'illumination ou de pose (*O* pour Oui, *N* pour Non).

Deuxième partie

Réseaux de neurones convolutionnels et décompositions parcimonieuses

Chapitre 3

Réseaux de neurones convolutionnels

La première partie de la thèse a vu l'utilisation de réseaux de neurones convolutionnels pour la reconnaissance faciale. Dans ce chapitre sont détaillés les perceptrons multi–couches et le modèle de réseau de neurones convolutionnels avec ses différents modules. Pour chacun de ceux–ci, les principales formules de propagation avant et arrière sont données. Différentes optimisations pour l'apprentissage des réseaux de neurones convolutionnels sont ensuite décrites, ainsi qu'une méthode de préapprentissage de couches de convolution.

3.1 Introduction

Les perceptrons multi–couches (ou MLP pour « *Multi–Layer Perceptron* », voir Section 3.2) ont montré leur efficacité comme technique d'apprentissage pour la classification de données. Ils sont en effet capables d'approximer des fonctions non–linéaires complexes afin de traiter des données de grande dimension.

Dans le cadre de la classification d'images, deux approches sont possibles :

- Extraire des caractéristiques directement des données. Classiquement, ces caractéristiques sont extraites par un algorithme choisi par l'utilisateur. Les vecteurs de caractéristiques obtenus sont ensuite présentés en entrée d'un réseau de neurones.
- Présenter l'image en entrée d'un réseau de neurones. L'image nécessite cependant d'être vectorisée, c'est à dire mise sous forme d'un vecteur dont la dimension est égale au nombre de pixels de l'image.

Dans le premier cas, le réseau se contente d'effectuer une classification des vecteurs de caractéristiques. Le point sensible (l'extraction des caractéristiques) est

laissé à la discrétion de l'utilisateur, et le choix de l'algorithme permettant l'extraction des caractéristiques est crucial.

Dans le deuxième cas, plusieurs problèmes se posent :

- Classiquement, les couches d'un réseau de neurones sont complètement connectées, c'est à dire que la valeur d'un neurone d'une couche n va dépendre des valeurs de *tous* les neurones de la couche $n-1$. Ainsi le nombre de connexions (et donc de poids, de paramètres) peut être très grand. Par exemple pour une imagette de taille 15×15, la dimension de l'entrée d'un MLP est de 225. Si la couche cachée comporte 100 neurones, alors le nombre de paramètres de cette couche est de $100 \times 225 = 22500$. Le nombre de paramètres va ainsi augmenter exponentiellement avec la dimension de l'entrée (des images). Cette grande complexité du réseau impose d'avoir de nombreux échantillons d'apprentissage, ce qui n'est souvent pas le cas. Le réseau va donc avoir tendance à faire un surapprentissage, et proposera donc une mauvaise capacité de généralisation.
- Un autre défaut des MLP pour une application à des images est qu'ils sont peu ou pas invariants à des transformations de l'entrée, ce qui arrive très souvent avec des images (légères translations, rotations ou distorsions).
- Enfin, les MLP ne prennent pas en compte la corrélation entre pixels d'une image, ce qui est un élément très important pour la reconnaissance de formes.

Les réseaux de neurones convolutionnels (ou CNN pour « *Convolutional Neural Network* », voir Section 3.3) sont une extension des MLP permettant de répondre efficacement aux principaux défauts des MLP. Ils sont conçus pour extraire automatiquement les caractéristiques des images d'entrée, sont invariants à de légères distorsions de l'image, et implémentent la notion de partage des poids permettant de réduire considérablement le nombre de paramètres du réseau. Ce partage des poids permet en outre de prendre en compte de manière forte les corrélations locales contenues dans une image. Les réseaux de neurones convolutionnels ont initialement été inspirés par la découverte faite par Hubel et Wiesel [142] de neurones sensibles aux aspects locaux et sélectifs en orientation dans le système visuel du chat.

La première utilisation des réseaux de neurones convolutionnels a été réalisée par Fukushima avec son *Neocognitron* [110], [111], [112], [113]. Les poids sont forcés à être égaux pour détecter des lignes, des points ou des coins à tous les endroits possibles de l'image, implémentant de fait l'idée du partage des poids [247].

Une avancée importante a été effectuée par Y. Lecun *et al.* [178] avec l'utilisation d'un réseau de neurones convolutionnels dont l'apprentissage a été réalisé par propagation arrière (*backpropagation*). Ce modèle a notamment été appliqué avec succès pour la reconnaissance de caractères manuscrits[72] (voir la Figure 3.9 pour l'architecture détaillée du réseau LeNet5).

3.2 Perceptron Multi–Couches

3.2.1 Le modèle du perceptron

Le perceptron a été introduit en 1958 par Franck Rosenblatt [243]. Il s'agit d'un neurone artificiel inspiré par la théorie cognitive de Friedrich Hayek et celle de Donald Hebb [130]. Dans sa version la plus simple, le perceptron n'a qu'une seule sortie y à laquelle toutes les entrées x_i sont connectées (voir Figure 3.1), ses entrées et sorties étant booléennes.

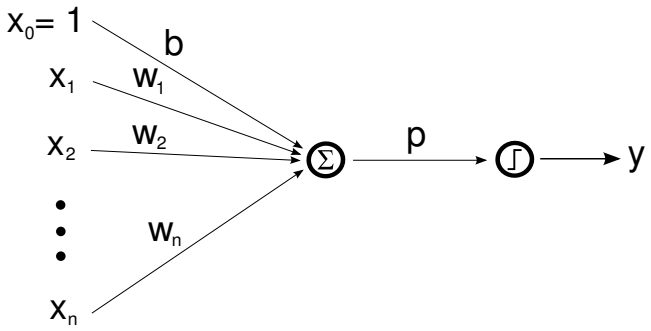


FIGURE 3.1 – Modèle du perceptron.

La somme pondérée des entrées par les poids w_i associés aux entrées est appelée *potentiel*, noté p.

$$p = \sum w_i x_i$$

Ce potentiel est alors soumis à une fonction seuil Φ de type Heavyside :

$$y = \begin{cases} 0 & \text{si} \quad s < 0 \\ 1 & \text{si} \quad s \geq 0 \end{cases}$$

De nombreuses variantes ont été développées, notamment la version la plus couramment utilisée où les entrées et sorties sont des nombres flottants. Les valeurs de sortie -1 et 1 remplacent fréquemment les valeurs 0 et 1. Une valeur particulière, appelée *biais* a également été introduite. Ce biais peut être vu comme une entrée x_0 supplémentaire dont la valeur est toujours de 1. Celui–ci a notamment été introduit pour un ajustement automatique du seuil, ou encore afin de pouvoir classer facilement un vecteur d'entrée dont toutes les composantes seraient nulles.

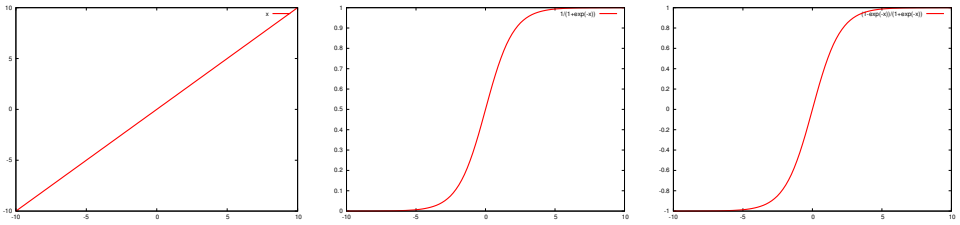
Le principal obstacle de ce modèle est la détermination des poids. Pour pallier ce problème, l'algorithme de rétropropagation du gradient a été mis au point. Soient S_0 et S_1 deux sous–ensembles de $\mathbb{R}^N$ représentant les exemples négatifs et positifs de l'échantillon à apprendre. L'algorithme consiste à présenter successivement les éléments de S_0 et S_1 et de mettre à jour les poids pour que la sortie *se rapproche* de la sortie désirée.

Cet algorithme repose sur le calcul du gradient de sortie puis sur la rétropropagation de celui–ci à travers la fonction de seuil puis des poids. C'est pourquoi la fonction de seuil doit être dérivable. La fonction d'activation de Heavyside est donc remplacée par des fonctions d'activation lui ressemblant et qui sont dérivables. Les principales fonctions d'activation Φ sont :

$$\text{linéaire} \qquad y = \Phi(p) = p \tag{3.1}$$

$$\text{sigmoïde} \qquad y = \Phi(p) = \frac{1}{1 + e^{-cp}} \quad (c > 0) \tag{3.2}$$

$$\text{tangente hyperbolique} \qquad y = \Phi(p) = \frac{1 - e^{-cp}}{1 + e^{-cp}} \tag{3.3}$$



(a) linéaire (b) sigmoïde (c) tangente hyperbolique

FIGURE 3.2 – Fonctions d'activation classiques.

La figure 3.2 montre les trois fonctions d'activation classiques. Il faut remarquer que la fonction linéaire est dans $]-\infty, +\infty[$, la fonction sigmoïde dans $]0, 1[$ et la fonction tangente hyperbolique dans $]-1, 1[$.

La propagation d'un vecteur d'entrée (x_i) à travers un perceptron s'écrit donc :

$$p = \sum w_i x_i \tag{3.4}$$

$$y = \Phi(p) \tag{3.5}$$

L'apprentissage classique d'un perceptron est la régression où la fonction de coût est de la forme :

$$L = \frac{1}{2}\|o - d\|^2$$

où o est la sortie obtenue pour un échantillon et d est la sortie désirée (ou label).
Le gradient de l'erreur en sortie est donc exprimé par :

$$\frac{\partial L}{\partial y} = o - d$$

Le gradient de l'erreur pour le potentiel est :

$$\frac{\partial L}{\partial p} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial p} \tag{3.6}$$

$$= \frac{\partial L}{\partial y}\Phi'(p) \tag{3.7}$$

Le gradient de l'erreur pour un poids w_i est :

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial p}\frac{\partial p}{\partial w_i} \tag{3.8}$$

$$= \frac{\partial L}{\partial p}x_i \tag{3.9}$$

Et le gradient de l'erreur pour l'entrée x_i est :

$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial p}\frac{\partial p}{\partial x_i} \tag{3.10}$$

$$= \frac{\partial L}{\partial p}w_i \tag{3.11}$$

Cependant, Minsky et Papert relèvent dans [206] que le perceptron échoue pour des problèmes de classification simples, comme ceux où les classes ne sont pas linéairement séparables. Le cas d'école est la classification du XOR, où les motifs $(0,0)$ et $(1,1)$ appartiennent à une classe tandis que les motifs $(1,0)$ et $(0,1)$ appartiennent à une autre classe.

Ces problèmes suggèrent l'utilisation de plusieurs perceptrons, qui organisés en couches forment le modèle du perceptron multi–couches. Ce modèle est capable de traiter des problèmes non–linéaires.

3.2.2 Le modèle de Perceptron Multi–Couches

Dans le modèle du Perceptron Multi–Couches, les perceptrons sont organisés en couches. Les perceptrons multi–couches sont capables de traiter des données qui ne sont pas linéairement séparables. Avec l'arrivée des algorithmes de rétropropagation [247], ils deviennent le type de réseaux de neurones le plus utilisé. Les MLP sont généralement organisés en trois couches [40], la couche d'entrée, la couche intermédiaire (dite couche cachée) et la couche de sortie. L'utilité de plusieurs couches cachées n'a pas été démontrée. La figure 3.3 illustre la structure d'un MLP présentant quatre neurones en entrée, trois neurones sur la couche cachée et deux en sortie. Lorsque tous les neurones d'une couche sont connectés aux neurones de la couche suivante, on parle alors de couches complètement connectées. Les équations de

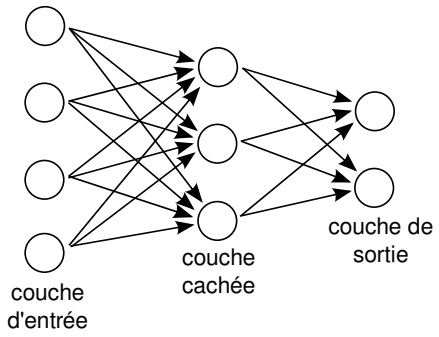


FIGURE 3.3 – Modèle du Perceptron Multi–Couches.

propagation décrites plus haut s'appliquent à tous les neurones. Cependant, le passage d'une couche à l'autre peut être formalisé sous forme matricielle.

Soit un MLP dont le nombre de neurones sur la couche d'entrée est n_0, n_1 sur la couche cachée et n_2 sur la couche de sortie. Les poids de la couche cachée peuvent s'écrire sous la forme d'une matrice $W \in \mathbb{R}^{n_1 \times n_0}$. Pour un vecteur $X \in \mathbb{R}^{n_0}$ présenté en entrée, le vecteur potentiel V et le vecteur de sortie Y pour la couche cachée s'écrivent donc :

$$V = WX \tag{3.12}$$
$$Y = \Phi(V) \tag{3.13}$$

avec Φ une fonction d'activation. La sortie de la couche cachée devient ensuite l'entrée de la dernière couche.

Rétropropagation du gradient pour un Perceptron Multi–Couches : La rétropropagation du gradient s'applique de manière récursive de la couche de sortie à la couche d'entrée. L'initialisation se fait donc par le calcul de l'erreur à la sortie de la dernière couche. Soit l'échantillon d'apprentissage $X \in \mathbb{R}^{n_0}$, et le vecteur de sortie désiré $D \in \mathbb{R}^{n_2}$. L'erreur sur la couche de sortie peut être définie comme précédemment :

$$L = \frac{1}{2}\|Y_2 - D\|^2$$

où le vecteur Y_2 représente la sortie du MLP (i.e. la sortie de la couche 2). De manière immédiate, le gradient de l'erreur en sortie peut être calculé :

$$\frac{\partial L}{\partial Y_2} = Y_2 - D$$

La rétropropagation du gradient s'applique alors récursivement de la couche de sortie à la couche d'entrée. Pour une couche dont X est le vecteur d'entrée, V le po-

tentiel, W la matrice de poids et Y la sortie, la rétropropagation s'écrit donc :

$$\frac{\partial L}{\partial V} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial V} \tag{3.14}$$

$$= \Phi'(V)\frac{\partial L}{\partial Y} \tag{3.15}$$

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial V}\frac{\partial V}{\partial X} \tag{3.16}$$

$$= W^T\frac{\partial L}{\partial V} \tag{3.17}$$

Mise à jour des poids : Une fois la rétropropagation du gradient effectuée pour toutes les couches, les matrices de poids sont mises à jour :

$$W \leftarrow W - \lambda dW$$

où λ est le taux d'apprentissage.

3.3 Modèle du Réseau de Neurones Convolutionnels

Dans cette section, nous détaillons les principaux types de neurones d'un réseau de neurones convolutionnels, leur construction, les méthodes pour les faire apprendre, leur modélisation.

Nous adoptons la démarche de décrire ces neurones sous forme modulaire pour faciliter la compréhension (et aussi le codage de ceux–ci). Un module possède donc une entrée (voire plusieurs), et une sortie, l'opération (convolution, subsampling, ...) réalisée par ce module étant atomique.

Par exemple, un neurone classique de convolution peut être décomposé en 3 *modules* distincts :

- un module de *convolution* réalisant la convolution d'une image d'entrée X par un noyau K,
- un module dit de *biais* réalisant essentiellement la somme du résultat du précédent module avec le biais b,
- un module assurant la non–linéarité (ou *squashing module*) du neurone, réalisant le passage de la sortie du précédent module à travers une fonction d'activation.

La construction d'un neurone particulier se réalise donc simplement en cumulant plusieurs modules, typiquement : convolution–biais–squashing pour un neurone de convolution.

La propagation avant (*fprop*) et arrière (*bprop*) d'un vecteur d'entrée X pour un perceptron multicouches décomposé en modules dans le cadre de l'*Energy Based Learning* proposé par Y. LeCun est présenté à la figure figure 3.4.

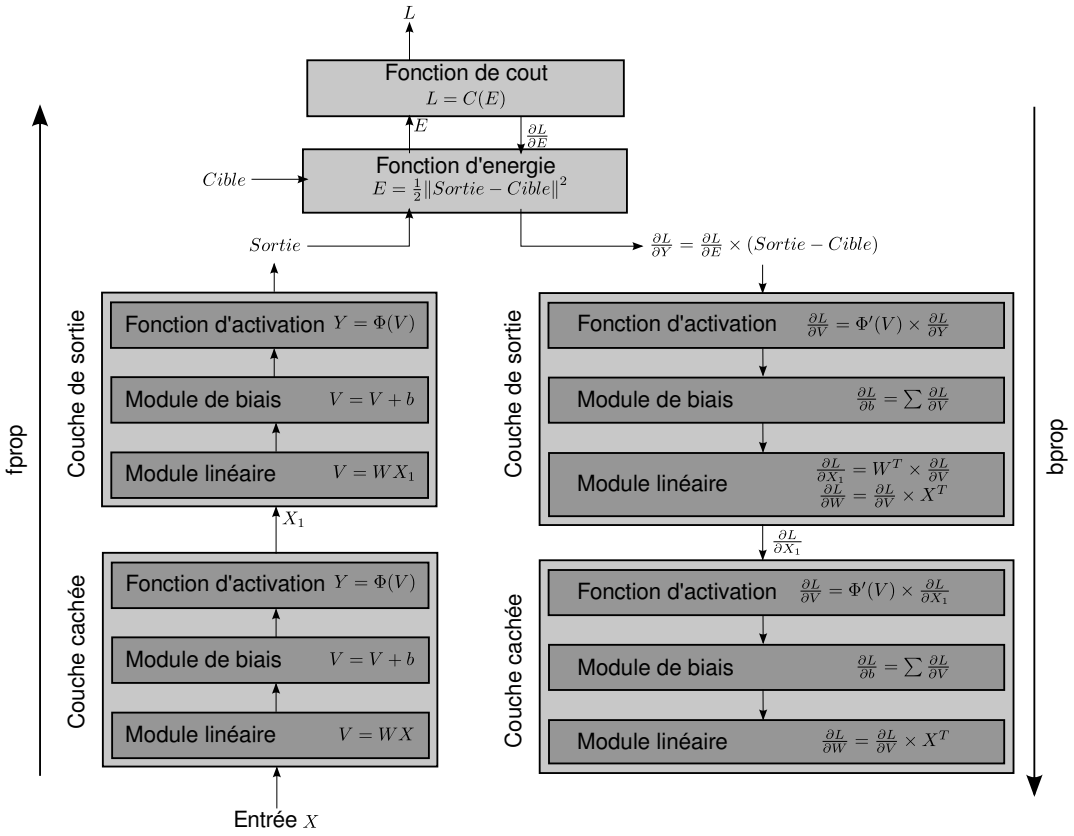


FIGURE 3.4 – Modèle basé énergie.

Cette vision modulaire d'un neurone permet une grande simplicité dans leur construction, une certaine souplesse dans leur manipulation, et l'implémentation est largement facilitée.

Dans la suite de cette section, nous décrivons les principaux modules utilisés lors de la construction d'un réseau de neurones convolutionnels (modules de convolution, subsampling, biais, module non–linéaire et d'autres modules existant dans la littérature). Pour chaque type de module, les fonctions de propagation avant (*fprop*) et arrière (*bprop*) sont dérivées. L'organisation de tels réseaux en couches ainsi que leur architecture globale sont également détaillées.

3.3.1 Module de Convolution

Les modules dits de convolution prennent une image X de taille $w_i \times h_i$ en entrée et la convolue par un noyau K de taille $w_k \times h_k$. Le masque de convolution est appliqué à toutes les positions sur l'image d'entrée telles qu'il se trouve toujours

dans l'image (voir figure 3.5). Ainsi les dimensions de l'image de sortie sont :

$$w_o = w_i - \lceil \frac{w_k}{2} \rceil \tag{3.18}$$

$$h_o = h_i - \lceil \frac{h_k}{2} \rceil \tag{3.19}$$

Plus formellement, le potentiel du module est une image Y (aussi appelée carte de convolution) dont les valeurs par propagation avant (ou *forward* propagation, ou encore *fprop*) sont calculées ainsi :

$$Y(i,j) = \sum_{(u,v)\in K} K(u,v)X(i+u, j+v)$$

où $0 \leq u \leq w_k$ et $0 \leq v \leq h_k$.

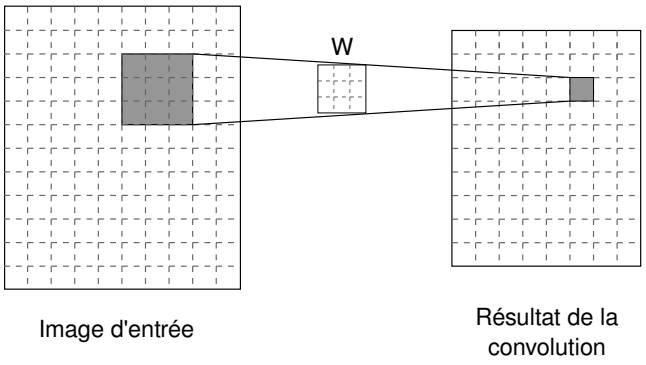


FIGURE 3.5 – Convolution d'une image de taille 12×10 par un noyau 3×3.

Rétropropagation du gradient pour un module de convolution : La rétropropagation du gradient (ou *backpropagation*, ou plus simplement *bprop*) pour un module de convolution est similaire à celle utilisée pour un perceptron (voir Section 3.2.1), modifiée cependant pour prendre en compte le partage des poids. Supposons connu le gradient $\frac{\partial L}{\partial Y}$ pour la carte de convolution Y, les gradients pour le masque de convolution et l'entrée X se calculent donc ainsi :

$$\frac{\partial L}{\partial W}(u,v) = \sum_{(i,j)\in Y} \frac{\partial L}{\partial Y}(i,j)X(i+u, j+v) \tag{3.20}$$

$$\frac{\partial L}{\partial X}(i,j) = \sum_{\left\{ \begin{smallmatrix} i & = & y+u \\ j & = & x+v \end{smallmatrix} \right.} M(y+u, x+v) \tag{3.21}$$

$$\text{où} \quad M(y+u, x+v) = K(u,v)\frac{\partial L}{\partial Y}(y,x) \tag{3.22}$$

avec les indices x, y, u, v, i, j positifs ou nuls. La rétropropagation du gradient pour un module de convolution est résumée à l'algorithme 1.

Algorithme 1: Rétropropagation du gradient pour un module de convolution.

Entrées : Carte de sortie Y, noyau W, et dérivée partielle de la couche de sortie $\frac{\partial L}{\partial Y}$
Sorties : Gradients $\frac{\partial L}{\partial W}$ et $\frac{\partial L}{\partial X}$
pour $i \leftarrow Y_h$ **faire**
  pour $j \leftarrow Y_w$ **faire**
    pour $u \leftarrow W_h$ **faire**
      pour $v \leftarrow W_w$ **faire**
        $\frac{\partial L}{\partial X}(i+u, j+v) = \frac{\partial L}{\partial X}(i+u, j+v) + \frac{\partial L}{\partial Y}(i,j)W(u,v)$
        $\frac{\partial L}{\partial W}(u,v) = \frac{\partial L}{\partial W}(u,v) + \frac{\partial L}{\partial Y}(i,j)X(i+u, j+v)$

Le masque de convolution du module est ensuite mis à jour par descente de gradient :

$$W \leftarrow W - \lambda \frac{\partial L}{\partial W}$$

3.3.2 Module de Subsampling

Le module de sous–échantillonnage (ou *subsampling*) prend en entrée une image X de taille $w_i \times h_i$ et la sous–échantillonne par un facteur $w_k \times h_k$, ce qui revient à réduire la taille de l'image par un facteur w_k en largeur, et par un facteur h_k en hauteur. Le module de subsampling le plus couramment utilisé est celui réalisant un moyennage de dimension s, on parle alors de *average pooling*. L'opération revient à convoluer l'image par un noyau de taille $w_k \times h_k$, dont les $w_k \times h_k$ valeurs sont égales, en appliquant un pas de w_k selon la largeur et un pas de h_k selon la hauteur (voir figure 3.6). Le résultat de chaque pas de « convolution » est alors multiplié par le poids unique w du module de subsampling.

Plus formellement :

$$Y(i,j) = w \sum_{(u,v) \in K} X(h_k \times i + u, w_k \times j + v)$$

Une dimension classique pour un module de subsampling est 2×2, où la dimension de l'image d'entrée est divisée par 4.

Rétropropagation du gradient pour un module de subsampling : Supposons connu le gradient de l'erreur $\frac{\partial L}{\partial Y}$, les équations relatives à la rétropropagation du

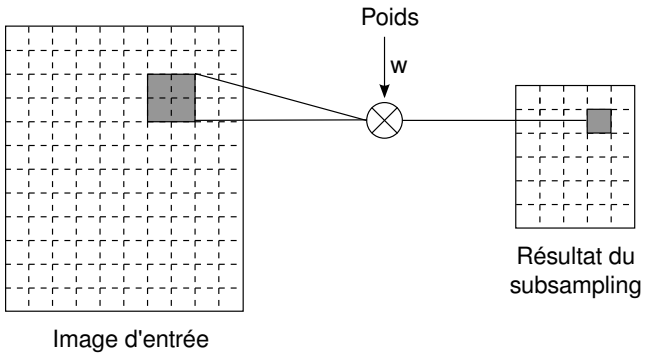


FIGURE 3.6 – Sous–échantillonnage 2×2 d'une image de taille 12×10.

gradient pour un module de subsampling sont donc :

$$\frac{\partial L}{\partial w} = \sum_{(i,j)\in Y} \sum_{(u,v)\in K} \frac{\partial L}{\partial Y}(i,j) X(h_k \times i + u, w_k \times j + v) \quad (3.23)$$

$$\frac{\partial L}{\partial X}(i,j) = w \times \frac{\partial L}{\partial Y}(\lfloor \frac{i}{h_k} \rfloor, \lfloor \frac{j}{w_k} \rfloor) \quad (3.24)$$

Le poids du module est ensuite mis à jour par descente de gradient :

$$w \leftarrow w - \lambda \frac{\partial L}{\partial w}$$

3.3.3 Module de Biais

Le module du biais est à la fois très simple et essentiel. Simple car il ne réalise qu'une addition du biais b à toutes les valeurs de l'image d'entrée X, essentiel car il permet (comme pour un MLP) d'ajuster automatiquement le seuil d'activation d'un neurone, ou encore de classer facilement une image d'entrée dont toutes les composantes seraient nulles.

La sortie d'un module de biais est donc une image Y de même taille que l'image d'entrée X :

$$Y(i,j) = X(i,j) + b$$

Rétropropagation du gradient pour un module de biais : Supposons connu le gradient de l'erreur $\frac{\partial L}{\partial Y}$ pour l'image de sortie, les équations relatives à la rétropropagation du gradient pour un module de biais sont donc :

$$\frac{\partial L}{\partial b} = \sum_{(i,j)\in Y} \frac{\partial L}{\partial Y}(i,j) \quad (3.25)$$

$$\frac{\partial L}{\partial X}(i,j) = \frac{\partial L}{\partial Y}(i,j) \quad (3.26)$$

Une fois le gradient de l'erreur calculé pour le biais, la mise à jour s'effectue simplement :

$$b \leftarrow b - \lambda \frac{\partial L}{\partial b}$$

3.3.4 Module non linéaire

Le module non linéaire est un élément essentiel pour un réseau de neurones dont le but serait de pouvoir traiter des données de façon non linéaire.

La sortie d'un module non linéaire est un vecteur Y de même taille que le vecteur d'entrée X :

$$Y(i,j) = \Phi(X(i,j))$$

où Φ est la fonction d'activation non linéaire. Le choix de cette fonction Φ est un élément important à ne pas négliger lors de la mise en place du réseau (voir la section 3.4.3).

Rétropropagation du gradient pour un module non linéaire : Supposons connu le gradient de l'erreur $\frac{\partial L}{\partial Y}$, l'équation relative à la rétropropagation du gradient pour un module non linéaire est :

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial X} \tag{3.27}$$

$$= \frac{\partial L}{\partial Y} \cdot \Phi'(X) \tag{3.28}$$

où $\cdot$ représente la multiplication terme à terme.

3.3.5 Autres types de modules

Des types de modules autres que les modules de convolution et de subsampling existent, tels les modules dits de convolution inverse ou d'upsampling introduits dans [248]. La figure 3.7 montre le principe du module de convolution inverse. Ce type de module est essentiellement utilisé pour de la reconstruction d'images.

La sortie d'un module de convolution inverse se calcule ainsi :

$$Y(i,j) = \sum_{\left\{ \begin{smallmatrix} i & = & y+u \\ j & = & x+v \end{smallmatrix} \right.} M(y+u, x+v) \tag{3.29}$$

$$\text{où} \quad M(y+u, x+v) = K(u,v)X(y,x) \tag{3.30}$$

Le module d'upsampling est l'exact inverse du module de subsampling, où l'image d'entrée est sur–échantillonnée par un facteur w_k en largeur et h_k en hauteur, le tout

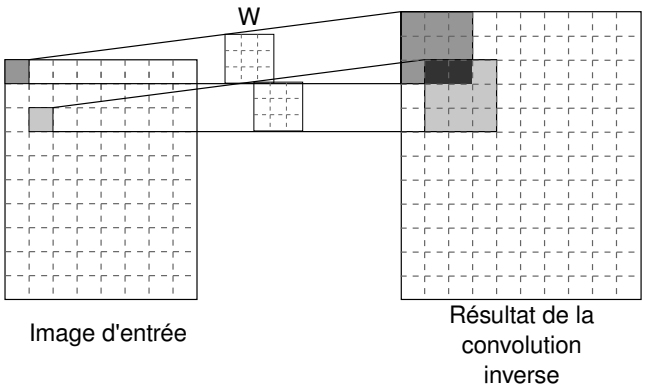


FIGURE 3.7 – Principe de la convolution inverse d'une image de taille 10×8 par un noyau de taille 3×3. Le résultat est une image de taille 12×10.

étant multiplié par le poids w du module :

$$Y(i,j) = w\ X(\lfloor \frac{i}{h_k} \rfloor, \lfloor \frac{j}{w_k} \rfloor)$$

Enfin, des variantes du module de subsampling ont été proposées dans [151]. Elles consistent pour la plupart à ne pas considérer un moyennage local (*average–pooling*) mais de prendre le maximum sur le voisinage (*max–pooling*), la valeur absolue de l'image d'entrée (*abs–pooling*), ou encore de considérer une normalisation locale du contraste.

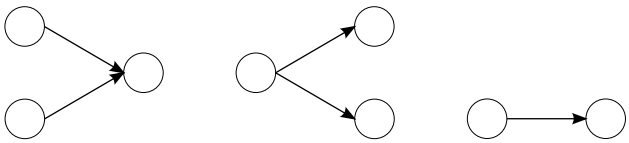
3.3.6 Organisation du réseau en couches

Les réseaux de neurones convolutionnels sont organisés en couches. Une couche consiste en un nombre n_l de neurones du même type (convolution, subsampling, upsampling, neurones de type perceptron...), ces neurones étant eux–même une succession de modules, dont les principaux sont décrits plus haut.

De nombreux cas de connexions peuvent se présenter dans la construction d'un réseau, tous se ramenant aux trois principaux types de connexions présentés à la figure 3.8 :

- la fusion (figure 3.8(a)), où un neurone d'une couche l a deux entrées (ou plus) sur la couche $l-1$,
- la division (figure 3.8(b)), où deux (ou plusieurs) neurones de la couche l ont une entrée commune sur la couche $l-1$,
- le transfert, où un neurone de la couche l n'a qu'une seule entrée sur la couche $l-1$, et cette entrée n'est connectée qu'à un seul neurone de la couche l.

Ces cas diffèrent pour les calculs de la propagation avant et arrière (respectivement *fprop*, et *bprop*).



(a) fusion (b) division (c) transfert

FIGURE 3.8 – Les trois types de connexions entre neurones.

fprop : Dans le cas de la fusion (figure 3.8(a)), les résultats du premier module (convolution, subsampling ou autre) du neurone de la couche l sont calculés séparément pour les deux neurones de la couche $l-1$, puis additionnés. S'appliquent ensuite les autres modules du neurone. Dans le cas de la division ou du transfert, les calculs s'effectuent de manière séparée en suivant les équations spécifiques aux modules des neurones de la couche l.

bprop : Dans les cas de la fusion et du transfert, les calculs sont effectués de manière classique. Dans le cas de la division, les gradients rétropropagés de la couche l sont additionnés.

Architecture générale : Une architecture typique, comme le réseau LeNet5 [72] (voir la figure 3.9), débute par plusieurs successions de couches de convolution et de subsampling suivies par des couches de neurones complètement connectés. Les premières couches peuvent être vues comme la partie se chargeant de l'extraction de caractéristiques, les couches de neurones complètement connectés se chargeant de la classification de ces caractéristiques. Le réseau LeNet5 est composé d'une première couche de convolution C_1 qui extrait des caractéristiques bas niveau (typiquement des gradients). Puis une couche de subsampling S_2 réalise un sous–échantillonnage de ces caractéristiques. Une deuxième couche de convolution C_3 extrait des caractéristiques de plus haut niveau et les fusionne selon un schéma bien défini. Une deuxième couche de subsampling réalise une nouvelle fois un sous–échantillonnage sur ces caractéristiques. Puis une succession de couches de neurones complètement connectés se charge de la classification de ces caractéristiques.

Architecture détaillée : Les couches sont reliées entre elles par des matrices de connexion. Ces matrices sont de taille $n_{l-1} \times n_l$ où n_l est le nombre de neurones de la couche l et n_{l-1} le nombre de neurones de la couche précédente. Ces matrices sont des matrices booléennes où un 1 en (i, j) spécifie que le neurone j de la couche l est connectée au neurone i de la couche $l-1$, et où un 0 spécifie que les deux neurones en question ne sont pas connectés entre eux. La matrice de connexion définie entre

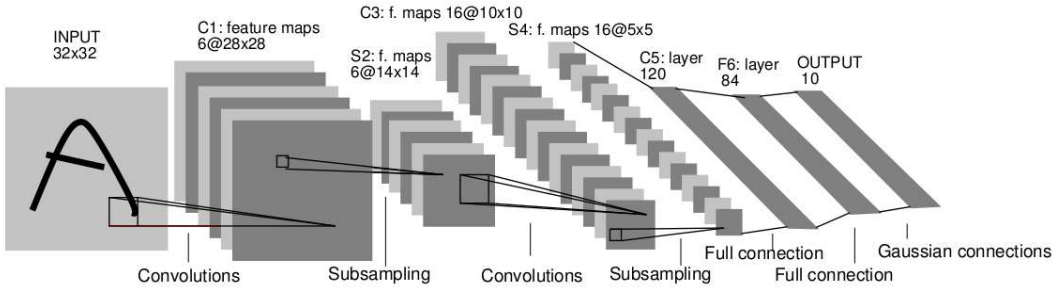


FIGURE 3.9 – Architecture du réseau LeNet5 [72].

deux couches va déterminer à partir de quelles cartes vont être extraites les caractéristiques ainsi que leur éventuelle fusion. Un exemple de matrice de connexion entre les couches S_2 et C_3 pour le réseau LeNet5 [72] (voir la figure 3.9) est présenté au tableau 3.1.

S_2 \ C_3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE 3.1 – Matrice de connexion entre les couches S_2 et C_3 du réseau LeNet5 [72].

Dans le cas particulier de la matrice de connexion présentée au tableau 3.1, les neurones de la couche C_3 prennent des combinaisons de 3, 4 ou 6 caractéristiques et les fusionne. De cette matrice dépend donc la pertinence des caractéristiques haut–niveau extraites, ainsi que les noyaux de convolution appris lors de l'apprentissage.

Une couche de convolution est généralement suivie directement par une couche de subsampling. Le nombre de neurones de cette couche de subsampling est égal au nombre de neurones de convolution de la couche précédente, et chaque neurone de subsampling n'est connecté qu'à un seul neurone de convolution. Cette couche de subsampling a principalement pour effet de rendre, dans une certaine mesure, les caractéristiques extraites invariantes à de petites transformations affines (comme des translations ou des rotations).

Étant donné que les deux couches sont presque tout le temps liées de cette façon,

il n'est pas rare de les considérer comme une seule couche dite de convolution–subsampling réalisant l'extraction de caractéristiques puis le sous–échantillonnage.

L'entrée est souvent considérée comme une couche à part entière (couche 0), la matrice de connexion de la première couche étant alors une matrice C_{1-0} de taille $n_1 \times n_0$ ne comportant que des 1.

Notons enfin que, à part pour les Perceptrons Multi–Couches où des techniques d'élagage ont été développées [179], l'architecture globale d'un réseau de neurones convolutionnels est souvent déterminé manuellement, en essayant plusieurs architectures, et en analysant leurs performances *a posteriori*. Cette étape peut être fastidieuse étant donné le grand nombre de paramètres régissant l'architecture d'un réseau de neurones convolutionnels :

- la taille de l'image (ou des images) d'entrée,
- le nombre de couches,
- la nature des couches (convolution, subsampling ou autre),
- la dimension des poids (notamment pour les couches de convolution et de subsampling),
- le nombre de neurones par couche,
- les différentes fonctions d'activation,
- les matrices de connexion reliant les couches entre elles.

3.4 Optimisation

Dans cette section, nous nous attachons à décrire des méthodes permettant une meilleure minimisation de la fonction de coût associée au réseau. La plupart de ces méthodes ont été mises en œuvre lors de la thèse pour l'apprentissage des réseaux de neurones convolutionnels.

3.4.1 Réordonner les échantillons d'apprentissage

Étant donné que le réseau va apprendre plus avec des échantillons inconnus ou inattendus, il est préférable de présenter ceux–ci le plus souvent possible. En effet, un échantillon bien classé lors de l'apprentissage n'apportera que peu de gradient, donc le réseau ne modifiera que peu ses paramètres. Dans le cadre d'un apprentissage *batch* (où les poids du réseau sont mis à jour avec les gradients accumulés de tous les échantillons d'apprentissage), l'ordre de présentation des échantillons importe peu. Dans le cadre d'une approche stochastique pour l'apprentissage, les paramètres du réseau sont mis à jour après chaque passage d'une échantillon d'apprentissage. Cependant, il n'y a pas de méthodes efficaces pour déterminer à priori quel est l'échantillon qui apportera le plus d'information, sauf peut–être la recherche

exhaustive (et donc coûteuse). Une méthode simple consiste donc à mélanger l'ordre de passage des échantillons après chaque passage de l'ensemble d'apprentissage. Ainsi, les échantillons successifs n'appartiennent pas à la même classe, et contiennent donc potentiellement plus d'information.

3.4.2 Normaliser les entrées

Pour un apprentissage plus rapide, les entrées doivent subir autant que possible un prétraitement :

– leur moyenne doit être nulle,
– leur variance égale,
– les échantillons d'apprentissage décorrélés autant que possible.

En effet, une moyenne nulle assure que les valeurs du vecteur d'entrée ne sont pas toutes du même signe. Une variance égale pour tous les échantillons d'apprentissage assure qu'aucune valeur n'est aberrante, chose qui pourrait saturer un neurone et donc augmenter (ou diminuer) disproportionnellement son poids. Enfin des échantillons décorrélés permettent au réseau de ne pas trop se focaliser sur des informations redondantes, et donc permettent une meilleure généralisation du réseau. Une analyse en composantes principales (aussi appelée transformée de Karhunen–Loeve) sur l'ensemble de données d'apprentissage peut aider dans cette tâche. Une vue schématique de la transformation des entrées est présentée à la figure 3.10.

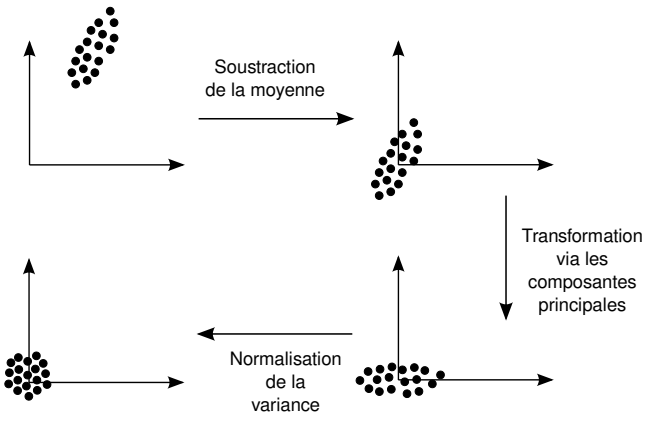


FIGURE 3.10 – Schéma de la normalisation des entrées.

3.4.3 Choisir la fonction d'activation

La fonction d'activation non–linéaire des neurones est ce qui permet au réseau de traiter les problèmes non–linéaires. Le choix de la fonction d'activation est un élément crucial, aussi bien pour la rapidité de calcul lors de l'apprentissage que

pour obtenir une bonne généralisation. Prenons l'exemple de données d'apprentissage dont les labels (les sorties désirées) sont 1 ou -1. Si la fonction d'activation choisie est la fonction tangente hyperbolique $\Phi(x) = \frac{1-e^{-x}}{1+e^{-x}}$, alors la sortie du neurone ne sera 1 ou -1 qu'à l'infini (les asymptotes de Φ). Un échantillon d'apprentissage, même bien classé lors de la phase d'apprentissage *produira* donc toujours un gradient en sortie. Si l'ensemble d'apprentissage n'est pas très bien construit (autant d'échantillons positifs que de négatifs), ou que le processus d'apprentissage est poussé trop loin (typiquement trop d'itérations ou taux d'apprentissage pas adapté), alors le réseau va avoir tendance à faire un sur–apprentissage de l'ensemble d'apprentissage. Ce sur–apprentissage va donc fortement pénaliser les capacités de généralisation du réseau pour classer un ensemble de test.

Une fonction d'activation recommandée dans [177] est de la forme (voir la figure 3.11) :

$$\Phi(x) = 1.7159\ tanh(\frac{2}{3}x)$$

Cette fonction d'activation présente plusieurs caractéristiques intéressantes :
- $\Phi(\pm 1) = \pm 1$,
- la dérivée seconde de Φ a un maximum de 1 en $x = 1$,
- symétriquement, la dérivée seconde de Φ a un minimum de -1 en $x = -1$,
- le gain de la fonction est proche de 1.

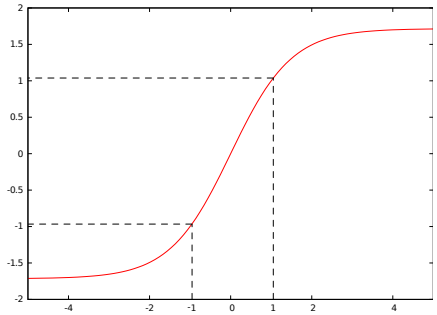


FIGURE 3.11 – Fonction d'activation recommandée ($1.7159\ tanh(\frac{2}{3}x)$).

Si les entrées sont de moyenne 0 et de variance 1 (voir plus haut pour la normalisation des entrées), et que les poids des neurones n'ont pas été initialisés de manière aberrante, alors les sorties d'une couche seront également de moyenne proche de 0 et de variance proche de 1. Étant donné que ces sorties sont également les entrées de la couche suivante, il n'est donc pas nécessaire de les re–normaliser.

Notons de plus que l'ajout d'un faible terme linéaire à la fonction d'activation peut parfois aider à sortir le réseau de régions plates (où les valeurs calculées par la fonction sigmoïde sont très faibles) :

$$\Phi(x) = a\ \tanh(bx) + cx$$

3.4.4 Initialiser les poids

L'initialisation des poids peut avoir un effet significatif sur la phase d'apprentissage. En effet, si les poids sont trop grands, alors les neurones via leur fonction d'activation vont saturer. Les gradients rétropropagés vont donc être faibles, et l'apprentissage lent. De même, si les poids sont trop petits, les gradients vont être également faibles. En effet, la fonction d'activation de type sigmoïde présentée plus haut peut être décomposée en une partie quasi–linéaire comprise entre -1 et 1, et le reste étant non–linéaire. Ainsi, une initialisation correcte des poids permet deux choses :

- les sorties d'un neurone ont une moyenne et une variance du même ordre que l'entrée (si les entrées ont été normalisées correctement, voir plus haut),
- le réseau va d'abord apprendre la partie linéaire de la transformation, avant la partie non–linéaire plus délicate.

Ainsi, si l'initialisation décrite plus haut a été appliquée, alors les entrées du réseau sont de moyenne nulle et de variance 1. Pour assurer le même résultat à toutes les couches, les sorties des neurones doivent être de moyenne nulle (ou quasi–nulle) et de variance proche de 1. Prenons l'exemple d'une couche de m neurones complètement connectés, dont le vecteur d'entrée X est composé de valeurs $x_i \in \mathbb{R}^n$ qui sont décorrélées et de variance égale à 1. On a donc $Var(x_i) = \frac{1}{n}\sum_i {x_i}^2$, les sorties $y_j \in \mathbb{R}^m$ des neurones de la couche (avant les modules de biais et le module non–linéaire) sont de la forme :

$$y_j = \sum_i x_i \, w_{ij}$$

Les y_j compose ainsi le vecteur de sortie $Y \in \mathbb{R}^m$ dont la variance peut s'écrire :

$$\begin{aligned}
Var(Y) &= \frac{1}{m} \sum_j {y_j}^2 && (3.31)\\
&= \frac{1}{m} \sum_j \sum_i \left(x_i \, w_{ij}\right)^2 && (3.32)\\
&= \frac{1}{m} \underbrace{\sum_i {x_i}^2}_{n} \sum_j {w_{ij}}^2 && (3.33)\\
&= \frac{n}{m} \sum_i {w_{ij}}^2 && (3.34)\\
&= n \, Var(w_i) && (3.35)
\end{aligned}$$

Or si l'on veut que la variance de la sortie soit égale à 1 ($Var(Y) = 1$), il faut que $Var(w_i) = 1/n$, d'où :

$$\sigma_{w_i} = n^{-\frac{1}{2}}$$

Les poids doivent donc être initialisés selon une loi uniforme de moyenne nulle et d'écart–type $n^{-1/2}$ où n est le nombre d'entrées du neurone.

3.4.5 Choisir le taux d'apprentissage

Le taux d'apprentissage λ contrôle la vitesse à laquelle les poids sont mis à jour selon la formule générale

$$w(t+1) = w(t) - \lambda \frac{\partial E}{\partial w}$$

Le choix du taux d'apprentissage est crucial, étant donné qu'un taux d'apprentissage trop faible va entraîner un apprentissage long, tandis qu'un taux d'apprentissage trop grand va entraîner une divergence du réseau. De plus, dans le dernier cas, le phénomène de sur–apprentissage va apparaître ; le réseau va apprendre trop des premiers exemples d'apprentissage et se spécialiser sur ceux–ci, entraînant une très mauvaise capacité de généralisation du réseau.

Il convient de rappeler tout d'abord les deux principales méthodes d'apprentissage :

- le mode *batch* : chaque échantillon de l'ensemble d'apprentissage subit une propagation avant, puis une propagation arrière à travers le réseau. Les poids ne sont cependant pas mis à jour tout de suite, les gradients propres à chaque poids sont accumulés. Une fois tous les échantillons passés, les poids sont mis à jour avec la somme des gradients obtenus.
- le mode *stochastique* : chaque échantillon de l'ensemble d'apprentissage subit une propagation avant, puis une propagation arrière à travers le réseau. Les poids sont alors mis à jour avec les gradients obtenus avec cet échantillon. Les gradients sont ensuite remis à zéro avant le passage de l'échantillon suivant.

Dans la suite, nous appelons *époque* un passage complet (fprop + bprop) de l'ensemble des échantillons d'apprentissage. Ainsi, une itération d'apprentissage en mode batch corrrespond à une époque.

Même si le choix du mode d'apprentissage dépend essentiellement des données que le réseau a à traiter, le mode stochastique offre potentiellement de meilleures perspectives, notamment si il existe une certaine redondance dans l'ensemble d'apprentissage. Par exemple, supposons que l'utilisateur dispose d'un ensemble de 1000 échantillons d'apprentissage composé de 10 fois les mêmes 100 échantillons. Alors, une époque en mode batch est potentiellement 10 fois plus lente qu'un apprentissage en mode stochastique.

De nombreux travaux ont été réalisés quant au choix du taux d'apprentissage. La modification du taux d'apprentissage en cours d'apprentissage est également un

sujet ayant vu la réalisation de nombreuses recherches. L'adaptation du taux d'apprentissage peut globalement se diviser en deux approches : les méthodes globales et les méthodes locales. Les méthodes globales essaient d'ajuster un paramètre d'apprentissage global à tous les poids, tandis que les méthodes locales affectent des taux d'apprentissage différents pour chaque poids. Généralement, les méthodes globales se concentrent sur la vitesse d'apprentissage, alors que les méthodes locales se penchent sur la capacité de généralisation des réseaux. L'heuristique communément admise pour l'évolution du taux d'apprentissage au cours de l'apprentissage est que si le vecteur de poids oscille alors le taux d'apprentissage est diminué, sinon il est augmenté. Cette heuristique générale fonctionne pour un apprentissage batch, cependant le vecteur de poids oscille tout le temps lorsque l'apprentissage est stochastique, la rendant ainsi caduque dans le cas d'un tel apprentissage.

Salomon *et al.* proposent dans [249] une méthode permettant de faire évoluer le taux d'apprentissage. Après chaque itération, deux mises à jour du vecteur de poids sont effectuées séparément : une avec le taux d'apprentissage légèrement augmenté, et l'autre avec le taux d'apprentissage légèrement diminué. Le vecteur de poids réalisant la plus faible erreur suite à une propagation avant est retenu et utilisé pour l'itération suivante. Cette méthode fonctionne pour un apprentissage en mode batch ou stochastique. Sa lourdeur (la méthode nécessite en effet deux propagations avant supplémentaires pour calculer l'erreur obtenue avec les deux vecteurs de poids) la handicape cependant sérieusement.

L'heuristique dite « bold driver » a été employée par Battiti *et al.* dans [31] ou encore par Vogl *et al.* dans [287]. L'erreur est calculée après chaque époque, et si elle diminue, alors le taux d'apprentissage global est légèrement augmenté. En revanche, si l'erreur augmente, alors le taux d'apprentissage global est largement diminué. Cette heuristique ne fonctionne malheureusement que pour un apprentissage en mode batch.

L'ajout d'un terme appelé *momentum*

$$\Delta w(t+1) = \lambda \frac{\partial E_{t+1}}{\partial w} + \mu \Delta w(t)$$

lors de la mise à jour des poids peut accroître la vitesse d'apprentissage. En effet, ce terme peut diminuer les pas d'apprentissage le long de la surface où la courbure est grande, et donc augmenter ces taux lorsque la courbure est faible [236]. Le momentum permet également parfois d'éviter de tomber dans un minimum local, le terme agissant comme un terme d'inertie. Le terme μ représente la force du terme momentum. Il a été démontré que l'ajout d'un terme momentum permet l'amélioration de la convergence en mode batch ; l'utilité pour le mode stochastique n'a cependant

pas été démontrée.

Il est également clair que les taux d'apprentissage doivent idéalement être différents pour chaque poids pour le mode stochastique de l'apprentissage. En effet, la surface de la fonction d'erreur influe directement sur les taux d'apprentissage. Ainsi, les poids des couches basses évoluent généralement moins vite que les poids des couches supérieures, les taux d'apprentissage respectifs doivent donc correspondre. En effet, les dérivées secondes de la fonction d'erreur par rapport aux poids sont généralement plus faibles dans les couches basses que dans les couches supérieures.

Une analyse des dérivées secondes de la fonction d'erreur par rapport au poids permet donc de régler les taux d'apprentissage de chacun des poids.

La courbure de la surface d'erreur peut s'exprimer à l'aide de la matrice Hessienne des paramètres du réseau (les poids). Une analyse des éléments propres de la Hessienne permet d'estimer les directions dont la courbure est la plus forte, et donc d'ajuster les taux d'apprentissage selon ces directions. Cependant, pour un réseau dont le nombre de paramètres est n, la matrice Hessienne H est une matrice de taille $n \times n$. Le nombre de paramètres pouvant être très grand pour un réseau de neurones, l'analyse en éléments propres (inversion d'une matrice de taille $n \times n$) de la Hessienne peut devenir impossible.

Aussi, plusieurs méthodes d'estimation des valeurs propres et vecteurs propres de cette Hessienne ont été mis au point, sans calculer explicitement ces matrices.

La plus simple permet de définir un taux d'apprentissage différent pour chaque poids à partir des dérivées secondes rétropropagées. Ceci est possible dans la mesure où une fonction de coût L a été définie préalablement. Dans le cas d'une régression classique où la fonction de coût est de la forme :

$$L = E \quad \text{avec l'énergie } E = \|output - target\|^2$$

les dérivées secondes en sortie s'expriment sous la forme :

$$\frac{\partial^2 L}{\partial Y^2} = \frac{\partial^2 E}{\partial Y^2}$$

l'initialisation des dérivées secondes pour la dernière couche est donc :

$$\frac{\partial^2 E}{\partial Y^2} = \mathbf{1}$$

Dans la suite, nous décrivons en détail le calcul de la rétropropagation pour une couche de neurones complètement connectés, et nous donnons les formules adéquates pour les principaux modules qui constituent une réseau de neurones convolutionnels.

Rétropropagation des dérivées secondes pour une couche de neurones complètement connectés Soit une couche de neurones complètement connectés dont le vecteur d'entrée est X, V le potentiel, W la matrice de poids et Y la sortie. Rappelons les équations de propagation avant :

$$V = WX \tag{3.36}$$
$$Y = \Phi(V) \tag{3.37}$$

La rétropropagation des dérivées secondes s'écrit donc :

$$\frac{\partial^2 L}{\partial V^2} = \frac{\partial}{\partial V}\left(\frac{\partial L}{\partial V}\right) \tag{3.38}$$
$$= \frac{\partial}{\partial V}\left(\Phi'(V)\frac{\partial L}{\partial Y}\right) \tag{3.39}$$
$$= \Phi''(V)\frac{\partial L}{\partial Y} + \Phi'(V)\frac{\partial^2 L}{\partial V \partial Y} \tag{3.40}$$

En utilisant l'approximation de Gauss–Newton (voir [43] pour plus de détails), on peut éliminer le terme en $\Phi''(V)$, ce qui donne :

$$\frac{\partial^2 L}{\partial V^2} = \Phi'(V)\left(\frac{\partial^2 L}{\partial V \partial Y}\right) \tag{3.41}$$
$$= \Phi'(V)\frac{\partial}{\partial Y}\left(\frac{\partial L}{\partial V}\right) \tag{3.42}$$
$$= \Phi'(V)\frac{\partial}{\partial Y}\left(\Phi'(V)\frac{\partial L}{\partial Y}\right) \tag{3.43}$$
$$= \left(\Phi'(V)\right)^2 \frac{\partial^2 L}{\partial Y^2} \tag{3.44}$$

où $\left(\Phi'(V)\right)^2$ représente la multiplication terme à terme. De même, la rétropropaga-

tion des dérivées secondes vers la d'entrée se calcule par :

$$\begin{aligned}
\frac{\partial^2 L}{\partial X^2} &= \frac{\partial}{\partial X}\left(\frac{\partial L}{\partial X}\right) && (3.45)\\
&= \frac{\partial}{\partial X}\left(W^T \frac{\partial L}{\partial V}\right) && (3.46)\\
&= W^T \frac{\partial^2 L}{\partial X \partial V} && (3.47)\\
&= W^T \frac{\partial^2 L}{\partial V \partial X} && (3.48)\\
&= W^T \frac{\partial}{\partial V}\left(\frac{\partial L}{\partial X}\right) && (3.49)\\
&= W^T \frac{\partial}{\partial V}\left(W^T \frac{\partial L}{\partial V}\right) && (3.50)\\
&= \left(W^T\right)^2 \frac{\partial^2 L}{\partial V^2} && (3.51)
\end{aligned}$$

où $\left(W^T\right)^2$ représente la multiplication terme à terme de la matrice W^T et non le produit matriciel.

En fixant les dérivées secondes sur la dernière couche à des valeurs positives, on peut constater à partir des équations plus haut que toutes les dérivées secondes dans le réseau vont être positives.

Rétropropagation des dérivées secondes pour un module de convolution De la même manière, les dérivées secondes peuvent être calculées pour les autres modules. Ainsi pour un module de convolution, la rétropropagation des dérivées secondes est résumée dans l'algorithme 2 :

Algorithme 2: Rétropropagation des dérivées secondes pour un module de convolution.

Entrées : Carte de sortie Y, noyau W, et dérivée partielle de la couche de sortie $\frac{\partial^2 L}{\partial Y^2}$

Sorties : Dérivées secondes $\frac{\partial^2 L}{\partial W^2}$ et $\frac{\partial^2 L}{\partial X^2}$

pour $i \leftarrow Y_h$ **faire**
  pour $j \leftarrow Y_w$ **faire**
    pour $u \leftarrow W_h$ **faire**
      pour $v \leftarrow W_w$ **faire**
        $\frac{\partial^2 L}{\partial X^2}(i+u, j+v) = \frac{\partial^2 L}{\partial X^2}(i+u, j+v) + \frac{\partial^2 L}{\partial Y^2}(i,j) W(u,v)^2$
        $\frac{\partial^2 L}{\partial W^2}(u,v) = \frac{\partial^2 L}{\partial W^2}(u,v) + \frac{\partial^2 L}{\partial Y^2}(i,j) X(i+u, j+v)^2$

Rétropropagation des dérivées secondes pour un module de subsampling Les dérivées secondes pour un module de subsampling s'écrivent :

$$\frac{\partial^2 L}{\partial w^2} = \sum_{(i,j)\in Y} \sum_{(u,v)\in K} \frac{\partial^2 L}{\partial Y^2}(i,j) X(h_k \times i + u, w_k \times j + v)^2 \quad (3.52)$$

$$\frac{\partial^2 L}{\partial X^2}(i,j) = w^2 \times \frac{\partial^2 L}{\partial Y^2}(\lfloor \frac{i}{h_k} \rfloor, \lfloor \frac{j}{w_k} \rfloor) \quad (3.53)$$

Rétropropagation des dérivées secondes pour un module de biais Les dérivées secondes pour un module de biais s'écrivent :

$$\frac{\partial^2 L}{\partial X^2}(i,j) = \frac{\partial^2 L}{\partial Y^2}(i,j) \quad (3.54)$$

$$\frac{\partial^2 L}{\partial w^2} = \sum_{(i,j)\in Y} \frac{\partial^2 L}{\partial Y^2}(i,j) \quad (3.55)$$

Rétropropagation des dérivées secondes pour un module non linéaire Il est très pratique de différencier le module non linéaire de son module principal associé (comme un module de convolution pour une couche de convolution). Ainsi, si la propagation avant s'écrit $Y = \Phi(X)$, les dérivées secondes pour un tel module s'expriment par :

$$\frac{\partial^2 L}{\partial X^2} = \Phi'(X)^2 \frac{\partial^2 L}{\partial Y^2} \quad (3.56)$$

où les multiplications de $\Phi'(X)^2$ sont réalisées terme à terme.

Notons que la méthode décrite ici fonctionne bien pour un apprentissage en mode stochastique, pour des réseaux de grandes tailles. Pour un apprentissage batch, différentes méthodes efficaces pour estimer les paramètres de la Hessienne ont été proposées. Elles sont fondées notamment sur les méthodes de Gauss–Newton, Levenberg–Marquardt ou encore la méthode de Broyden-Fletcher-Goldfarb-Shanno (BFGS). La plupart de ces méthodes utilisent une recherche par ligne (ou « *line search* »), ce qui devient impossible dans le cadre d'un apprentissage stochastique. De plus, il n'est possible de calculer explicitement la Hessienne que pour des réseaux de faible taille, ceux–ci n'étant pas vraiment ceux nécessitant d'être accélérés.

Méthodologie La mise en œuvre de la méthode s'effectue en deux étapes. Premièrement, la diagonale de la matrice hessienne peut être estimée grâce à un moyennage des dérivées secondes sur plusieurs échantillons. La stabilité de ce moyennage

intervient après un relativement faible nombre d'échantillons (par exemple, dans nos tests une stabilité satisfaisante a été obtenue après le passage de 100 échantillons de la base de chiffres manuscrits MNIST composée de 60000 échantillons d'apprentissage). Ce moyennage s'effectue simplement par :

$$\left(\frac{\partial^2 L}{\partial w^2}\right)_{moy} \leftarrow \left(\frac{\partial^2 L}{\partial w^2}\right)_{moy} + \gamma \left(\frac{\partial^2 L}{\partial w^2}\right)_{p}$$

où $\left(\frac{\partial^2 L}{\partial w^2}\right)_{p}$ représente la dérivée seconde de l'erreur L par rapport au poids w pour le pattern p. Une valeur typique de γ est le nombre de patterns que l'on a décidé d'utiliser pour calculer ce moyennage.

Deuxièmement, une fois ce moyennage effectué, le taux d'apprentissage de chaque poids peut être calculé par :

$$\lambda_w = \frac{\lambda}{\frac{\partial^2 L}{\partial w^2}_{moy} + \mu}$$

où μ est un paramètre permettant au taux d'apprentissage de ne pas « exploser » dans le cas où $\frac{\partial^2 L}{\partial w^2}_{moy}$ serait trop faible. Le paramètre λ est le taux d'apprentissage global, et est classiquement diminué après chaque époque (voir les méthodes plus haut concernant la dynamique du taux d'apprentissage global en mode batch).

Cette méthode peut être vue comme un prétraitement du réseau et est résumée dans l'algorithme 3 :

Algorithme 3: Calcul des taux d'apprentissage .

Initialisation : $\left(\frac{\partial^2 L}{\partial w^2}\right)_{moy} \leftarrow 0$

Ordonnancement aléatoire des échantillons p_i d'apprentissage

pour $k \leftarrow \{1, N\}$ **faire**

- fprop de l'échantillon p_k
- calcul des dérivées secondes pour l'échantillon p_k
- mise à jour du vecteur moyen :
- $\left(\frac{\partial^2 L}{\partial w^2}\right)_{moy} \leftarrow \left(\frac{\partial^2 L}{\partial w^2}\right)_{moy} + \frac{1}{N}\left(\frac{\partial^2 L}{\partial w^2}\right)_{p_k}$

Affectation des taux d'apprentissage pour chaque poids : $\lambda_w = \frac{\lambda}{\frac{\partial^2 L}{\partial w^2}_{moy}} + \mu$

Notons que le vecteur moyen des dérivées secondes est relativement stable au cours des époques. Celui ci peut néanmoins être recalculé entre deux époques avec la même méthode afin éventuellement de l'affiner.

3.5 Ensemble d'apprentissage et validation

Le nombre d'échantillons disponibles avant toute procédure d'apprentissage est un élément très important à prendre en compte pour la réalisation (voire le succès) d'un apprentissage. L'heuristique communément admise est que, plus l'on dispose d'échantillons, meilleur sera l'apprentissage. L'ensemble des échantillons provient généralement d'une base de données. Cette base est typiquement divisée avant la phase d'apprentissage en trois parties *distinctes* :

- l'ensemble d'apprentissage,
- l'ensemble de validation,
- et l'ensemble de test.

L'ensemble d'apprentissage regroupe les échantillons de la base de données qui vont servir effectivement à l'apprentissage du réseau. C'est en fonction des caractéristiques des échantillons de cet ensemble que vont être modifiés les poids du réseau afin que celui–ci converge vers un minimum (global dans le meilleur des cas) de la fonction de coût associée au réseau.

L'ensemble de validation est constitué d'échantillons de la base servant à réaliser régulièrement au cours de l'apprentissage des validations de celui–ci. Cet élément est essentiel pour obtenir une bonne généralisation du réseau. En effet, il a été montré que si la courbe d'erreur de l'ensemble d'apprentissage diminue presque tout le temps, il n'en est pas de même pour la courbe d'erreur de l'ensemble de validation (voir la figure 3.12). Typiquement, celle-ci après avoir diminuée remonte : c'est le phénomène de sur–apprentissage. L'ensemble de validation permet donc de réaliser un arrêt prématuré de l'apprentissage. En effet, lorsque l'erreur sur cet ensemble ne diminue plus, le point auquel le réseau est le plus capable de généraliser a été atteint.

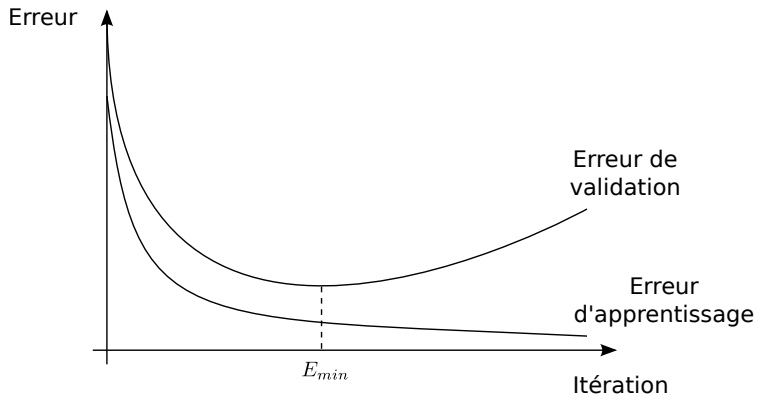


FIGURE 3.12 – Évolution typique de l'erreur d'apprentissage et de validation.

Le dernier ensemble d'échantillons est l'ensemble de test. Celui–ci permet une fois l'apprentissage terminé d'évaluer la qualité du réseau obtenu. L'intérêt de cet

ensemble par rapport à l'ensemble de validation (grâce auquel une bonne généralisation est censée avoir été obtenue) est que l'utilisation de l'ensemble de validation fait partie du processus d'apprentissage, et à ce titre peut introduire un biais dans l'apprentissage. En utilisant des données indépendantes pour les tests (l'idéal étant d'avoir des données provenant d'une autre base de données), ceux–ci ne sont donc pas biaisés.

S'il n'existe pas de protocole bien défini avec la base de données, la méthode classique consiste alors en la séparation de manière aléatoire de la base de données. Un ratio typique est :

- 80% de la base de données pour l'apprentissage,
- 10% pour la validation,
- et 10% pour les tests.

Cependant, ces ratios peuvent changer selon le nombre d'échantillons dans la base d'apprentissage. Ainsi, si celui–ci est faible, il est préférable d'adopter la méthode dite « *Leave–One–Out* ». Cette méthode consiste à ne sélectionner qu'un seul échantillon de la base pour former l'ensemble de test, et de réitérer le processus entier d'apprentissage un nombre suffisant de fois avec une séparation aléatoire de la base afin d'obtenir des résultats significatifs. Notons que dans tous les cas, réaliser plusieurs apprentissages avec des séparations différentes de la base de données permet d'obtenir des résultats plus significatifs.

En ce qui concerne la biométrie, il n'est pas rare que les bases de données possèdent un protocole, définissant clairement les données à utiliser pour l'apprentissage et les données de tests. Le gros avantage de ces protocoles est qu'ils permettent de comparer les algorithmes entre eux sans que les résultats ne soient biaisés. Un utilisateur voulant tester une nouvelle méthode sur une telle base n'a donc pas besoin de tester d'autres algorithmes pour se comparer à l'état de l'art, pour peu que les résultats de ceux–ci aient été publiés.

3.6 Préapprentissage

Les successions de couches de convolution et de subsampling présentes au début d'un réseau de neurones convolutionels ont pour tâche d'extraire des caractéristiques pertinentes de l'image d'entrée. Cette hiérarchie dans l'extraction de caractéristiques est classiquement apprise par descente de gradient. Cependant, il est nécessaire de refaire tout l'apprentissage (y compris de ces filtres) à chaque nouvel apprentissage. De plus, les données d'apprentissage peuvent poser un problème si elles sont trop peu nombreuses. En effet, pour qu'un apprentissage classique d'un réseau de neurones soit réussi, il est nécessaire que celui–ci extraie uniquement les caractéristiques pertinentes, cela nécessitant de nombreuses données d'apprentissage (qui ne sont malheureusement pas toujours disponibles).

Une possibilité de contourner ces problèmes est d'entraîner l'extracteur de caractéristiques indépendamment de manière non supervisée. Ce type d'apprentissage est alors réalisé couche par couche en utilisant différents algorithmes [135], [133].

Ce type d'approche est symptomatique de la problématique dite du *Deep Learning* [39]. Les architectures profondes présentent en effet certains intérêts, notamment celui d'abstraction d'informations à partir de données. Leur apprentissage reste cependant le principal verrou étant donné le grand nombre de couches.

Le principe, plutôt simple, consiste à initialiser les paramètres de chaque couche en utilisant un algorithme non supervisé, comme un RBM (pour « *Restricted Boltzmann Machine* ») ou un réseau de neurones auto–encodeur [38], [241]. Une fois les poids des couches initialisés, le réseau entier est optimisé via une descente de gradient supervisée.

Idéalement, les premières couches doivent extraire de l'image d'entrée des caractéristiques locales, discriminantes et robustes à certains artefacts (bruit). Apprendre de tels filtres de manière non supervisée permet de mieux contrôler la qualité des caractéristiques extraites. En effet, avec un réseau de neurones convolutionnels entraîné de manière classique (où toutes les couches sont entraînées par descente de gradient), certains hyperparamètres (comme le nombre de filtres de la première couche) peuvent être difficiles à régler a priori. De plus, un apprentissage non supervisé de ces filtres permet de « forcer » plus facilement les filtres à extraire certaines caractéristiques.

L'aspect local des filtres est déjà assuré par les noyaux de convolution ; restent les aspects discriminants et robustes des filtres. La théorie de la parcimonie répond aux deux derniers points. En effet, on dit d'une représentation d'un signal qu'elle est *parcimonieuse* si ce signal peut être décomposé en une combinaison linéaire des éléments de la base définissant le nouvel espace, dont de nombreux coefficients sont nuls. Ainsi, une telle représentation permet de décomposer le signal en un ensemble restreint d'éléments de base, chacun représentant une caractéristique atomique (et donc discriminante) du signal de départ. De plus, de nombreux travaux concernant le débruitage d'images à l'aide de représentations parcimonieuses ont montré que l'approche permettait de gérer de façon efficace le bruit.

De nombreux travaux [104] [221] [223], fondés sur l'intuition que les images naturelles peuvent généralement être décrites selon un faible nombre de primitives (par exemple des lignes, des coins ou des caractéristiques locales [105]), ont également montré que la parcimonie est appropriée pour la décomposition d'images naturelles.

Les éléments de base (ou vecteurs) sont souvent appelés *atomes* et ils sont regroupés au sein d'un *dictionnaire* (voir le chapitre 4 pour plus de détails).

Soit $\boldsymbol{s}$ un signal donné, et $\boldsymbol{\Phi}$ un dictionnaire dont les M colonnes ϕ_i sont les atomes, alors la décomposition parcimonieuse du signal revient à chercher le vecteur

$\boldsymbol{x}$ tel que :

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \quad \text{sachant que } \boldsymbol{s} = \sum_{i=1}^{M} x_i \boldsymbol{\phi}_i$$

Plusieurs méthodes de décomposition parcimonieuse ont été mises en œuvre pour la résolution de ce problème (plus de détails dans le chapitre 4). Les plus utilisées sont notamment les méthodes dites de *Basis Pursuit* ou encore de *Matching Pursuit*). Ces méthodes cependant cherchent à décomposer le signal au sens d'une combinaison linéaire, et mettent en œuvre bien souvent un processus itératif trop lent vis–à–vis du but recherché (la convolution). Le but ici est en effet d'obtenir une décomposition parcimonieuse via une convolution du signal par des filtres, ce qui est un procédé bien plus rapide, et permet de s'intégrer naturellement à l'architecture des réseaux de neurones convolutionnels.

Algorithme de préapprentissage Le but est d'apprendre une banque de filtres permettant, par des opérations rapides (comme la convolution), de décomposer le signal en une nouvelle représentation parcimonieuse (ou quasi–parcimonieuse, où de nombreux coefficients ne seraient pas forcément nuls mais proches de zéro).

Plusieurs algorithmes non supervisés ont vu le jour à ces fins. Ils reposent tous sur le même schéma d'auto–encodeur :

- un encodeur chargé de transformer une entrée en vecteur (parcimonieux autant que possible),
- un décodeur se chargeant de la reconstruction de l'entrée à partir du vecteur parcimonieux.

L'encodeur ici se charge d'apprendre les filtres permettant de décomposer de façon rapide le signal d'entrée en une représentation parcimonieuse. Quelque soit l'algorithme d'apprentissage utilisé, cet encodeur possède pour chaque signal d'entrée le code parcimonieux correspondant. Ce code parcimonieux permet de reconstruire le signal via le décodeur.

Les codes parcimonieux de chaque signal peuvent au préalable être obtenus par exemple via des algorithmes fondés sur les approches dites de *matching pursuit* ou de *basis pursuit*.

Dans les tests effectués lors de la thèse, nous avons utilisé l'algorithme PSD (pour « *Predictive Sparse Decomposition* ») proposé dans [155]. Cet algorithme apprend la représentation parcimonieuse (via le décodeur) ainsi que les filtres permettant de la prédire (via l'encodeur) en même temps. Tous les paramètres (de l'encodeur et du décodeur) sont obtenus par descente de gradient. Cet algorithme a cependant été légèrement modifié pour correspondre aux données de tests (des visages).

L'algorithme consiste en un auto–encodeur dont le vecteur d'entrée $Y \in \mathbb{R}^n$ est codé sous forme d'un vecteur parcimonieux $Z \in \mathbb{R}^m$, avec $m > n$. La reconstruction à partir de Z est ensuite réalisée pour obtenir le vecteur de sortie $\tilde{Y}$ (Figure 3.13).

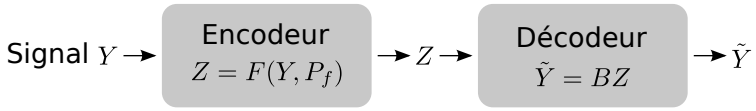


FIGURE 3.13 – Vue schématique de l'algorithme PSD.

Le coût de l'erreur de reconstruction est définie de manière classique par :

$$L(Y, B, Z) = \|Y - BZ\|_2^2$$

où $B \in \mathbb{R}^{n \times m}$ est la matrice de reconstruction. Les colonnes de B sont contraintes à être de norme 1 pour éviter toute divergence ou solution triviale.

Le coût associé à la parcimonie du vecteur Z s'exprime par :

$$L(Z) = \|Z\|_1$$

Idéalement, c'est la pseudo norme l^0 qui devrait être utilisée. Cette pseudo norme représente en effet le nombre de termes non nuls d'un vecteur donné. Cependant, l'utilisation de cette norme rend le problème de minimisation combinatoire (NP–complet), d'où la relaxation convexe de cette contrainte, et l'utilisation de la norme l^1 [62].

Enfin, le coût associé à l'encodage de X par l'auto-encodeur est défini par :

$$L(Y, P_f) = \|Z - F(Y, P_f)\|_2^2 \tag{3.57}$$

$$\text{avec } F(Y, P_f) = G \tanh(WY + D) \tag{3.58}$$

où $F(Y, P_f)$ représente la fonction non linéaire encodant le vecteur d'entrée Y, $P_f = \{G, W, D\}$ représentant les paramètres de l'encodeur. La matrice $W \in \mathbb{R}^{m \times n}$ correspond aux filtres de l'encodeur, la matrice $D \in \mathbb{R}^m$ est une matrice de biais, et la matrice $G \in \mathbb{R}^{m \times m}$ est une matrice de gains permettant de mettre à l'échelle la sortie de la fonction F.

Ainsi, la fonctionnelle à minimiser est de la forme :

$$L(Y, Z; B, P_f) = \|Y - BZ\|_2^2 + \lambda \|Z\|_1 + \alpha \|Z - F(Y, P_f)\|_2^2 \tag{3.59}$$

où les paramètres λ et α permettent de donner plus ou moins d'importance aux différentes erreurs.

En minimisant cette fonction de coût, l'encodeur parvient donc à produire de bonnes approximations des codes parcimonieux, et permet une bonne reconstruction de l'entrée via le décodeur.

La méthode d'apprentissage utilisée dans [155] se fait par descente de gradient. Il est effectué pour chaque échantillon de la base d'apprentissage à la fois sur l'encodeur, sur le code et sur le décodeur en alternant classiquement les deux étapes :

- minimisation de l'équation 3.59 selon Z en gardant les paramètres P_f et B constants,
- en utilisant le code parcimonieux trouvé précédemment, mise à jour des paramètres P_f et B. Mise à l'échelle de B pour que les colonnes aient une norme de 1.

Cette approche par descente de gradient permet d'apprendre en même temps les paramètres de l'encodeur et du décodeur. La descente de gradient n'est cependant pas idéale pour l'obtention de codes parcimonieux. En effet, le grand nombre de paramètres nécessite de nombreux réglages qui peuvent vite s'avérer fastidieux. De plus, les codes Z obtenus ne sont pas strictement parcimonieux, beaucoup de termes étant proches de zéro mais non nuls.

Néanmoins, un bon apprentissage permet de récupérer les paramètres de l'encodeur afin d'initialiser une couche de convolution d'un réseau de convolution. Celle–ci est alors légèrement modifiée pour prendre en compte les paramètres G, W et D.

Enfin, il est possible d'effectuer une préapprentissage de la deuxième couche de convolution du réseau, en utilisant les sorties de la première couche et de les utiliser comme entrées de l'algorithme PSD.

3.7 Conclusion

Dans ce chapitre, nous avons présenté les réseaux de neurones convolutionnels. Ces réseaux sont capables d'extraire des caractéristiques d'images présentées en entrée et de classifier ces caractéristiques. Ils sont fondés sur la notion de « champs récepteurs » (*receptive fields*) extrayant des caractéristiques locales des images, ainsi que sur les cellules « simples » et « complexes » du système visuel hiérarchisant en une succession de couches les caractéristiques extraites pour construire une représentation complexe d'un objet. Les réseaux de neurones convolutionnels sont ainsi capables d'extraire de manière non linéaire des caractéristiques bas–niveau de l'image, de les fusionner via la succession de couches pour produire des caractéristiques de plus haut–niveau. Les réseaux de neurones implémentent l'idée de partage des poids, permettant de réduire de beaucoup le nombre de paramètres libres de l'architecture. Ce partage des poids permet en outre de réduire les temps de calcul, l'espace mémoire nécessaire, et également d'améliorer les capacités de généralisation du réseau. L'utilisation de neurones de subsampling permet également

d'améliorer les performances du réseau en octroyant une certaine invariance à de petites transformations de l'entrée (translations ou distorsions).

Une avancée importante concernant les réseaux de neurones convolutionnels a été réalisée par Lecun *et al.* [178] avec l'utilisation de l'algorithme de rétropropagation du gradient pour l'apprentissage. En effet, l'extraction des caractéristiques par le réseau n'a plus à être choisie « manuellement », le réseau apprenant automatiquement les caractéristiques « utiles » pour la minimisation de l'énergie définie. La rétropropagation du gradient permet en outre de réaliser l'apprentissage non plus couche par couche, mais en une seule fois, où tous les paramètres du réseau sont mis à jour en même temps.

Les réseaux de neurones convolutionnels présentent cependant un certains nombre de limitations. En premier lieu, les hyperparamètres du réseau sont difficiles à évaluer a priori. En effet, le nombre de couches, les nombre de neurones par couche ou encore les différentes connexions entre couches sont des éléments cruciaux et essentiellement déterminés par une bonne intuition ou par une succession de tests/calcul d'erreurs (ce qui est coûteux en temps).

Le nombre d'échantillons d'apprentissage est également un élément déterminant, et il arrive souvent que celui–ci soit trop faible en comparaison du nombre de paramètres (poids) du réseau. Des solutions existent comme augmenter artificiellement leur nombre (en appliquant des distorsions contrôlées sur les échantillons originaux par exemple), ou encore en réduisant le nombre de paramètres libres (en réalisant un préapprentissage des premières couches par exemple).

Les méthodes de préapprentissage (telle que celle présentée dans ce chapitre) permettent de fixer certains poids au début de l'apprentissage supervisé. Les couches ayant ainsi subi un préapprentissage peuvent rester fixes, les poids ne nécessitant plus de mises à jour. Ainsi, un préapprentissage à l'aide de méthodes parcimonieuses permet aux premières couches du réseau d'extraire des caractéristiques discriminantes. Le nombre de neurones par couche est cependant grandement augmenté, ce qui peut ralentir l'apprentissage final. Le problème de la détermination automatique des matrices de connexion entre couches pour de tels préapprentissages reste également ouvert, celles–ci devant encore être déterminées manuellement.

Dans le chapitre suivant, nous présentons plus en détail la théorie de la parcimonie, de la décomposition de signaux sur des dictionnaires jusqu'à l'apprentissage de ceux–ci à partir des données. Une méthode de classification pour l'identification biométrique fondée sur la parcimonie [298] est également présentée.

Chapitre 4

Représentations parcimonieuses

4.1 Introduction

La parcimonie est devenue un concept très important dans le traitement du signal et des images. Cetaines tâches comme la compression, la restauration, l'estimation, la détection ou encore la séparation de sources ont connu un essor important grâce à des *a priori* parcimonieux.

Cette thèse a fait l'objet de l'étude de méthodes fondées uniquement sur la parcimonie dans le cadre biométrique. Les caractéristiques parcimonieuses extraites de l'image d'un visage ne sont donc plus utilisées au sein d'un réseau de neurones convolutionnels (comme au chapitre précédent).

Ce chapitre rappelle brièvement la terminologie du monde de la parcimonie, détaille les principales méthodes de décomposition parcimonieuse d'un signal sur un dictionnaire, ainsi que l'apprentissage de celui–ci à partir de données d'apprentissage. Enfin, une méthode efficace de classification utilisant la parcimonie est présentée dans le cadre de l'identification biométrique.

4.2 Représentations parcimonieuses

Dans cette section, nous nous attachons à fournir les principales définitions concernant l'étude de la parcimonie afin de mieux appréhender la suite du chapitre. Celles–ci vont ainsi être utilisées tout au long du chapitre. Dans la suite, nous considérons une image $\boldsymbol{x} \in \mathcal{H}$ (espace de Hilbert) de taille $\sqrt{N} \times \sqrt{N}$ réarrangée en un vecteur dans $\mathbb{R}^N$.

Définition 4.2.1 *(Atome).*
Un atome est un signal élémentaire de représentation d'un signal ou d'une image.

L'exemple le plus connu d'atome est la sinusoïde intervenant dans la décomposition de Fourier d'un signal. De nombreux autres atomes ont été créés afin de répondre à des besoins spécifiques. Les atomes sont regroupés au sein d'un dictionnaire.

Définition 4.2.2 *(Dictionnaire).*
Un dictionnaire $\mathbf{\Phi}$ est une collection d'atomes $(\boldsymbol{\phi}_i)_{i\in\mathcal{I}}$ avec $\mathcal{I}$ un ensemble dénombrable. Le dictionnaire peut être trié selon la fréquence (dictionnaire de Fourier), la position (dictionnaire de Dirac), la position–échelle (dictionnaire d'ondelettes), etc.

Dans le cadre de cette thèse, le dictionnaire $\mathbf{\Phi}$ est une matrice de taille $N \times M$ dont les M colonnes représentent les atomes $\boldsymbol{\phi}_m$ (de taille N, et souvent de norme 1) du dictionnaire. Lorsque le nombre de colonnes est supérieur au nombre de lignes ($M > N$), le dictionnaire est dit **redondant**. Dans ce cas, l'équation $\boldsymbol{x} = \mathbf{\Phi}\boldsymbol{\lambda}$ conduit à un système sous–déterminé, possédant une infinité de solutions pour $\boldsymbol{\lambda}$.

4.3 Décomposition d'un signal

Étant donné un signal $\boldsymbol{x} \in \mathbb{R}^N$ (ou une image de taille $\sqrt{N} \times \sqrt{N}$), nous cherchons sa décomposition selon un dictionnaire $\mathbf{\Phi}$ composé de M vecteurs $\boldsymbol{\phi}_m$ recouvrant $\mathbb{R}^N$. Définissons tout d'abord la norme l^p d'un vecteur $\boldsymbol{x}$:

$$\|\boldsymbol{x}\|_p = \left(\sum_i |x_i|^p \right)^{1/p}$$

avec le cas particulier de la « norme l^0 » (définie comme étant le nombre d'éléments non nuls de $\boldsymbol{x}$) :

$$\|\boldsymbol{x}\|_0 = \sum_{0\leq i<N} a_i \quad \text{où } a_i = \begin{cases} 1 & \text{si } x_i \neq 0 \\ 0 & \text{sinon} \end{cases}$$

Lorsque le dictionnaire est redondant ($M > N$), il y a un nombre infini de coefficients α_i possibles pour décomposer le signal selon le dictionnaire :

$$\boldsymbol{x} = \sum_{m=1}^{M} \alpha_m \boldsymbol{\phi}_m$$

Dans le cadre d'une décomposition parcimonieuse du signal, la décomposition optimale est celle possédant le moins de termes différents de zéro (ou dualement le maximum de termes nuls).

Ainsi le problème s'écrit :

$$\min_{\boldsymbol{\lambda}} \|\boldsymbol{\lambda}\|_0 \quad \text{sachant que } \boldsymbol{x} = \sum_{m=1}^{M} \lambda_m \boldsymbol{\phi}_m \tag{4.1}$$

Malheureusement ce problème d'optimisation combinatoire est NP–difficile [216].

Dans la communauté du traitement du signal, deux approches sont essentiellement utilisées afin de trouver des solutions (sous–optimales) à ce problème :

– La première approche consiste à modifier le terme de pénalité parcimonieuse ($\|\boldsymbol{\lambda}\|_0$) afin de rendre le problème convexe. L'exemple le plus connu est l'approche de la poursuite de bases (BP pour « *Basis Pursuit* ») [62] qui remplace la norme l^0 par une norme l^1. Malgré cette substitution de norme, BP présente la même solution que le problème d'approximation parcimonieuse optimale sous certaines conditions (voir [81] pour plus de détails). En pratique, la présence de bruit dans le signal conduit à l'approche appelée BPDN (pour « *Basis Pursuit DeNoising* ») [62] qui réalise un compromis entre l'erreur quadratique moyenne de reconstruction (MSE pour « *Mean–Squared Error* ») et la contrainte de parcimonie. Le problème s'écrit donc :

$$\min_{\boldsymbol{\lambda}} \left(\|\boldsymbol{x} - \sum_{m=1}^{M} \lambda_m \boldsymbol{\phi}_m\|_2^2 + \mu \|\boldsymbol{\lambda}\|_1 \right)$$

où μ est un paramètre permettant de réaliser un compromis entre l'attache aux données $\|\boldsymbol{x} - \sum_{m=1}^{M} \lambda_m \boldsymbol{\phi}_m\|_2^2$ et le critère de parcimonie $\|\boldsymbol{\lambda}\|_1$. L'approche BPDN fournit ainsi l'approximation la plus parcimonieuse pour une qualité de reconstruction donnée. De nombreux algorithmes ont été développés pour la résolution de ce problème (aussi connu sous le nom de *Lasso* pour « *Least absolute shrinkage and selection operator* »), fondés par exemple sur la méthode du point intérieur [202] ou encore sur des seuillages itératifs.

– La deuxième approche utilisée dans la communauté du traitement du signal est fondée sur des algorithmes gloutons construisant une représentation parcimonieuse du signal [281]. L'exemple classique d'un algorithme glouton est l'algorithme dit du « *Matching Pursuit* » (MP) [199]. A chaque itération, l'algorithme sélectionne l'atome minimisant le résidu entre le signal et la reconstruction obtenue à l'itération précédente. Bien que les algorithmes gloutons ne soient pas optimaux en général, l'algorithme MP (et ses variantes) permet d'obtenir en pratique de bonnes représentations parcimonieuses. La variante de l'algorithme appelée « *Orthogonal Matching Pursuit* » (OMP) est présentée ci–après.

4.3.1 Approches de « *Basis Pursuit* »

Afin de contourner la difficulté de la norme l^0 (Équation 4.1), tout un pan de la littérature s'est penché sur la relaxation convexe de la norme. Dans [83], il est suggéré de remplacer le problème l^0 (Eq.4.4) par sa relaxation convexe l^1 :

$$\min_{\boldsymbol{\lambda}} \|\boldsymbol{\lambda}\|_1 \quad \text{sachant que } \boldsymbol{x} = \sum_{m=1}^{M} \lambda_m \boldsymbol{\phi}_m \tag{4.2}$$

Ce problème connu sous le nom de « Poursuite de base » (ou BP) a notamment été formalisé par Donoho *et al.* [85], [86], [84]. Les minimisations l^0 et l^1 ne sont en général pas équivalentes. De nombreux travaux ont essayé de caractériser les hypothèses suffisantes (et nécessaires) sur $\boldsymbol{\Phi}$ pour que le minimiseur de BP coïncide avec celui de l'équation (4.1) ; on parle alors d'identifiabilité l^1 [46].

L'approche la plus couramment utilisée pour BP est fondée sur le seuillage. L'utilisation de seuillages a initialement été développée pour le débruitage d'images ayant subi une transformée en ondelettes (Figure 4.1).

FIGURE 4.1 – Transformée en ondelettes de *Lena* à une échelle.

En effet, les ondelettes offrent une bonne technique pour réduire le bruit dans une image. Les coefficients issus des ondelettes traduisent les discontinuités qui correspondent aux détails de l'image considérée. La sous–bande d'approximation représente l'information utile de l'image, tandis que les autres sous–bandes représentent les hautes fréquences. Le bruit est donc concentré dans les sous–bandes représentant les détails. Un seuillage des coefficients des sous–bandes de détails élimine les éléments les plus fins de l'image, permettant alors la réduction du bruit dans le fond. En fait, il existe plusieurs méthodes de seuillage des coefficients d'ondelettes telles que le seuillage doux (« *Soft Thresholding* ») et le seuillage dur (« *Hard Thresholding* »).

Pour rappel, le seuillage dur peut être formalisé ainsi :

$$HT_T(x) = \begin{cases} x & \text{si } |x| \geq T \\ 0 & \text{sinon} \end{cases}$$

De même, le seuillage doux est formalisé par :

$$ST_T(x) = \begin{cases} x - sign(x)T & \text{si } |x| \geq T \\ 0 & \text{sinon} \end{cases}$$

Relation avec l'opérateur proximal

L'utilisation du seuillage dans le cas d'une pénalité l^1 peut être retrouvé via l'opérateur proximal. La notion d'opérateur proximal a été introduite par Moreau [209], [210], [211]. Elle correspond à une projection généralisée d'un vecteur sur un polytope de dimension quelconque, et donc pas forcément sur la boule l^2. D'autres travaux récents [101] permettent de relier l'opérateur proximal au problème de la décomposition parcimonieuse. Soit la fonction de pénalité suivante :

$$\Psi(\boldsymbol{\lambda}) = \sum_{m=1}^{M} \psi_m(\lambda_m)$$

En supposant :

(i) ψ_i *est une fonction convexe, paire, positive, et croissante sur* $[0, +\infty[$ *;*

(ii) ψ_i *est continue sur* $\mathbb{R}$*, avec* $\psi_i(0) = 0$ *;*

(iii) ψ_i *est dérivable sur* $]0, +\infty[$*, elle n'est pas nécessairement différentiable en* 0 *et admet une dérivée positive à droite en* 0*,* $\psi'_{i+}(0) = \lim_{h \to 0} \frac{\psi_i(h)}{h} \geq 0$.

Proposition 4.3.1 *Sous les hypothèses* (i)–(iii)*, l'opérateur proximal de* $\kappa\psi_i$*,* $\kappa > 0$*, a une unique solution continue et impaire découplée en chaque coordonnée* i *:*

$$prox_{\kappa\psi_i}(x[i]) = \begin{cases} 0 & \textit{si } |x[i]| \leq \kappa\psi'_{i+}(0) \\ x[i] - \kappa\psi'_i(x[i]) & \textit{si } |x[i]| > \kappa\psi'_{i+}(0) \end{cases}$$

Des preuves de ce résultat peuvent être trouvées dans [100]. Des résultats similaires sont également décrits dans [28], [67], [219].

L'opérateur proximal revient donc à réaliser une opération de seuillage terme–à–terme. Dans le cas de la norme l^1, $\Psi(\alpha) = \sum_i |\alpha_i| = \|\alpha\|_1$, nous retrouvons le seuillage doux (Figure 4.2) :

$$prox_{\gamma\|.\|_1} = \boldsymbol{ST}_\gamma(\alpha) = \begin{cases} \alpha_i - sign(\alpha_i)\gamma & \text{si } \alpha_i \geq \gamma \\ 0 & \text{sinon} \end{cases}$$

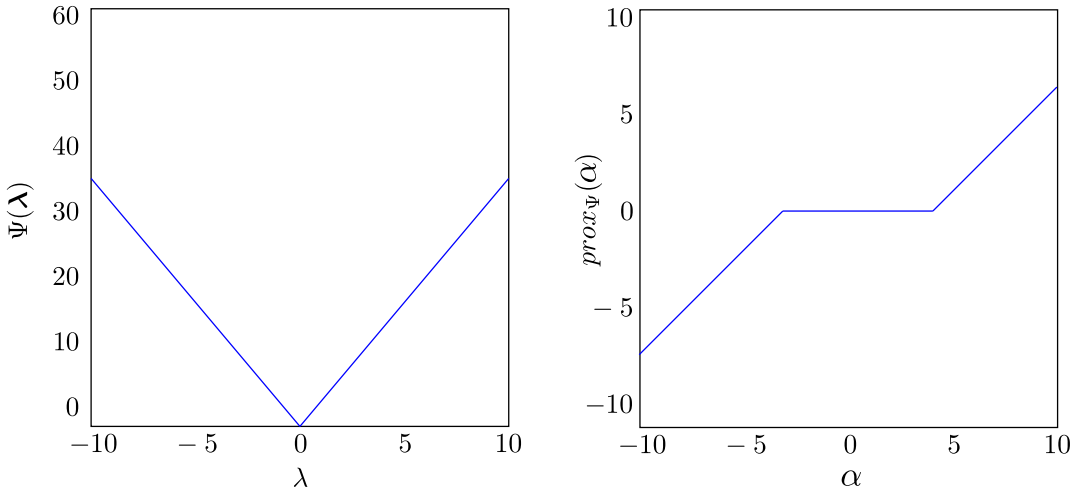


FIGURE 4.2 – Pénalité de parcimonie ψ_i (gauche), et opérateur proximal associé (droite).

Décomposition et schéma explicite–implicite

Les méthodes les plus couramment utilisées reposent sur des algorithmes itératifs faisant intervenir individuellement le calcul des opérateurs proximaux des itérés. Nous ne détaillons ici que l'approche mise en œuvre lors de cette thèse.

L'itération explicite–implicite (ou FB pour « *Forward–Backward* ») est une généralisation du gradient projeté pour les problèmes sous contraintes. Parmi les algorithmes ayant vu le jour, citons les algorithmes à un pas ne faisant intervenir que l'itération précédente dans la descente de gradient. Ce type d'algorithme peut ainsi s'écrire :

$$\boldsymbol{\lambda}^{(t+1)} = prox_{\mu_t F_1}\left(\boldsymbol{\lambda}^{(t)} - \mu_t \nabla F_2(\boldsymbol{\lambda}^{(t)})\right) \tag{4.3}$$

En particulier, lorsque $F_1(\boldsymbol{\lambda}) = \|\boldsymbol{\lambda}\|_p$, $p \geq 1$ et $F_2(\boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{\lambda}\|^2$, on retrouve l'équation classique typique de la parcimonie [73], [106] :

$$\min_{\boldsymbol{\lambda}} \|\boldsymbol{\lambda}\|_p \quad \text{sachant que } \boldsymbol{x} = \sum_{m=1}^{M} \lambda_m \boldsymbol{\phi}_m \tag{4.4}$$

Les algorithmes à un pas ne prennent cependant en compte que la dernière itération, perdant ainsi l'information des itérés précédents. Ainsi, Nesterov [218] propose une version à pas multiples prenant en compte la totalité des itérés. Des applications de ce type d'algorithme peuvent être trouvées dans [292]. En 2009, les auteurs de [34] ont proposé un schéma à pas multiples n'utilisant que les deux derniers itérés. Les auteurs montrent que la vitesse de convergence est alors la même que celle de

l'approche de Nesterov (en $O(1/t^2)$), quand celle des versions à un pas n'est qu'en $O(1/t)$.

L'algorithme mis en œuvre pour la résolution de l'équation (4.2) utilise les deux notions évoquées ci–dessus, à savoir une itération à pas multiples sur le calcul de l'opérateur proximal dans le cas d'une norme l^1. Il est résumé par l'algorithme 4.

Algorithme 4: Algorithme FISTA (*Fast Iterative Shrinkage–Thresholding Algorithm*).

Entrées : Dictionnaire $\mathbf{\Phi} \in \mathbb{R}^{N \times M}$, Signal $\boldsymbol{x} \in \mathbb{R}^N$, Nombre d'itérations L, Seuil γ
Sorties : Vecteur parcimonieux $\boldsymbol{\lambda} \in \mathbb{R}^M$
Initialisation
$\boldsymbol{\lambda}_0 = \mathbf{0}$, $\boldsymbol{y}_1 = \mathbf{0}$, $t_1 = 1$
pour $l = 1, 2, \ldots, L$ **faire**
- $\boldsymbol{y}_n = \boldsymbol{\lambda}_{n-1} + \mu \mathbf{\Phi}^T \left(\boldsymbol{x} - \mathbf{\Phi} \boldsymbol{\lambda}_{n-1} \right)$
- $\boldsymbol{\lambda}_n = ST_\gamma(\boldsymbol{y}_n)$
- $t_{n+1} = \frac{1+\sqrt{(1+4t_n^2)}}{2}$
- $\boldsymbol{y}_{n+1} = \boldsymbol{\lambda}_n + \left(\frac{t_n - 1}{t_{n+1}} \right) \left(\boldsymbol{\lambda}_n - \boldsymbol{\lambda}_{n-1} \right)$

$\boldsymbol{\lambda} = \boldsymbol{\lambda}_L$
retourner $\boldsymbol{\lambda}$

4.3.2 Approches de « *Matching Pursuit* »

L'algorithme appelé « *Orthogonal Matching Pursuit* » (ou OMP) est une variante très utilisée de l'algorithme de *Matching Pursuit* [74] [199]. Il s'agit d'un algorithme glouton sélectionnant itérativement l'atome minimisant l'erreur de reconstruction. À chaque itération, l'algorithme OMP calcule une nouvelle approximation $\hat{\boldsymbol{x}}^t$ du signal. L'erreur d'approximation $\boldsymbol{r}^t = \boldsymbol{x} - \hat{\boldsymbol{x}}^t$ est alors utilisée à l'itération suivante pour déterminer l'atome à sélectionner. La sélection de l'atome utilise le produit scalaire entre le résidu actuel $\boldsymbol{r}^t$ et les vecteurs colonnes $\boldsymbol{\phi}_i$ de $\mathbf{\Phi}$. Les indices des atomes sélectionnés sont stockés dans l'index Γ^t, où t est le compteur des itérations. Étant donné que l'algorithme sélectionne un atome à chaque itération et que $\Gamma^0 = \varnothing$, l'ensemble Γ^t contient t indices. Soit le produit scalaire :

$$\alpha_i^t = \boldsymbol{\phi}_i^T \boldsymbol{r}^t$$

Le nouvel élément est alors choisi comme étant celui ayant l'amplitude α_i^t la plus grande (et n'appartenant pas déjà à Γ^t) :

$$i_{max}^t = \arg_i \max |\alpha_i^t|$$

et l'ensemble des indices correspondant à des éléments différents de zéro Γ^t devient donc :

$$\Gamma^{t+1} = \Gamma^t \cup i^t_{max}$$

L'algorithme itère la sélection de nouveaux atomes jusqu'à ce que :

- le nombre maximal d'atomes ait été atteint (ce nombre maximal d'atomes étant un hyperparamètre de l'algorithme) ou,
- l'erreur de reconstruction passe au dessous d'un seuil (hyperparamètre défini par l'utilisateur).

L'algorithme OMP peut ainsi être résumé (Algorithme 5) :

Algorithme 5: Algorithme OMP.

Entrées : Dictionnaire $\mathbf{\Phi} \in \mathbb{R}^{N \times M}$, Signal $\boldsymbol{x}$, Nombre maximal d'atomes L, Tolérance tol

Sorties : Vecteur parcimonieux $\boldsymbol{\lambda}$

Initialisation :

$\boldsymbol{\lambda}^0 = \mathbf{0}$, $\boldsymbol{r}^0 = \boldsymbol{s}$, $\Gamma^0 = \varnothing$, $n = 0$

répéter

- $\alpha_i = \phi_i^T \boldsymbol{r}^{t-1} \quad \forall i \notin \Gamma^{t-1}$
- $i_{max} = \arg_i \max |\alpha_i|$
- $\Gamma^t = \Gamma^{t-1} \cup i_{max}$
- $\boldsymbol{\lambda}^t_{\Gamma^t} = \mathbf{\Phi}^+_{\Gamma^t} \boldsymbol{x}$
- $\hat{\boldsymbol{x}}^t = \mathbf{\Phi} \boldsymbol{\lambda}^t_{\Gamma^t}$
- $\boldsymbol{r}^t = \boldsymbol{x} - \hat{\boldsymbol{x}}^t$
- $t = t + 1$

jusqu'à $\|\boldsymbol{r}^t\| \,/\, \|\boldsymbol{r}^0\| < tol$ **ou** $t > L$

$\boldsymbol{\lambda} = \boldsymbol{\lambda}^t$

retourner $\boldsymbol{\lambda}$

où $\mathbf{\Phi}^+_{\Gamma^t}$ représente la pseudo inverse de la sous matrice $\mathbf{\Phi}_{\Gamma^t}$ créée à partir des colonnes de $\mathbf{\Phi}$ dont les indices sont définis par Γ^t. D'autres implémentations de l'algorithme OMP existent dans la littérature [43], [169].

4.4 Apprentissage de dictionnaires

4.4.1 Dictionnaires prédéfinis

Le contenu d'une image naturelle est souvent complexe et ne peut être représentée de façon optimale via une transformée unique. La transformée de Fourier est par exemple efficace pour la représentation parcimonieuse de textures globalement

oscillantes, alors que les ondelettes sont plus performantes avec des singularités isolées.

La représentation optimale dépend de l'espace auquel les signaux sont censés être apparentés. Parmi les espaces définis et créés pour certains types de signaux, citons :

- les séries de Fourier optimales pour les textures globalement oscillantes ;
- les ondelettes optimales pour les images à variation bornée [66] ;
- les ridgelets conçues pour caractériser les images régulières le long des lignes [55] ;
- les curvlets [53], [54], [56], les bandlets [230], [197], [198], les contourlets [80] ou les shearlets [122] conçues pour les images dites *cartoon* (régulières par morceau) ;
- les brushlets [203] et les wave–atoms [77] optimales pour les textures localement oscillantes.

Ces espaces de représentation sont optimaux (ou quasi optimaux) pour le type d'images pour lesquelles ils ont été créés. L'analyse d'une image naturelle ne peut cependant pas reposer sur un seul de ces espaces, étant donnée la diversité morphologique trop importante contenue dans ces images. Aussi, il est courant de construire le dictionnaire final par « concaténation » de plusieurs transformées. Le choix d'un dictionnaire est en effet une étape clef pour l'obtention d'une représentation parcimonieuse. Une telle concaténation permet donc d'obtenir des représentations parcimonieuses optimales au sens du « sous–dictionnaire » selon la partie de l'image considérée. Le principe de construction d'un tel dictionnaire est schématisé à la figure 4.3.

4.4.2 Méthodes d'apprentissage de dictionnaires

Bien qu'il soit possible de construire un dictionnaire pour décomposer un signal de façon parcimonieuse à l'aide de bases préexistantes, celui–ci peut ne pas représenter au mieux toute la diversité morphologique contenue dans les signaux à traiter. De plus, la représentation obtenue n'est pas nécessairement la plus parcimonieuse. C'est pourquoi un apprentissage du dictionnaire peut améliorer la représentation du signal lorsqu'il est décomposé sur celui–ci.

Les approches pour la conception de dictionnaires par apprentissage sont pour la plupart basées sur des itérations comprenant deux étapes :

- la première étape consiste, étant donné un dictionnaire à trouver les composantes parcimonieuses de la décomposition du signal. Cette étape est régulièrement appelée « inversion » ou encore « *sparse coding* » (section 4.3),
- la seconde étape consiste, une fois le code parcimonieux trouvé à l'étape précédente, à mettre à jour les atomes utilisés de façon à minimiser la fonction

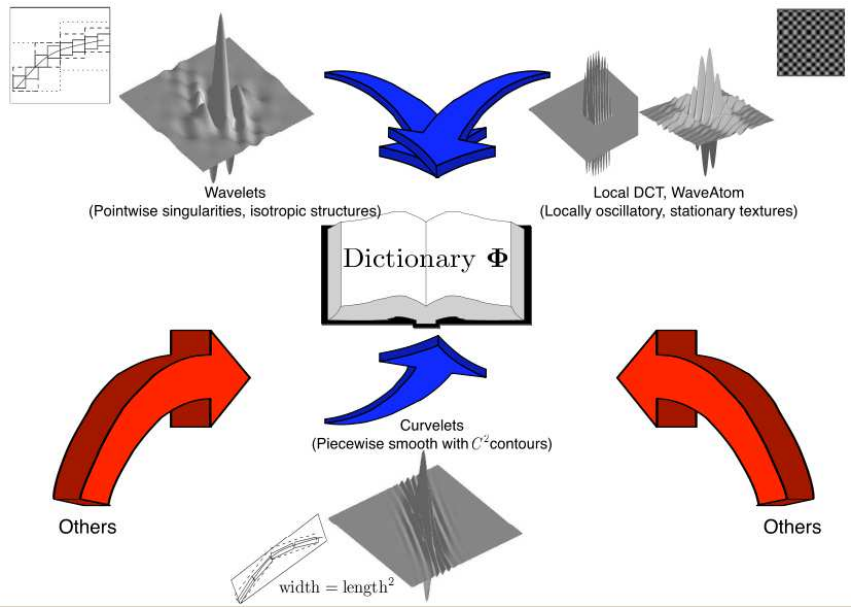


FIGURE 4.3 – Vue schématique de la diversité morphologique et du choix du dictionnaire associé (tiré de [99]).

de coût associée.

Les différentes méthodes d'apprentissage de dictionnaires diffèrent principalement au niveau de la seconde étape, à savoir la façon dont les atomes sont mis à jour, ainsi que la procédure utilisée pour la modification du dictionnaire. Dans la suite, nous présentons les principales méthodes proposées dans la littérature pour l'apprentissage de dictionnaires.

Les méthodes de Maximum de Vraisemblance

Les méthodes utilisées dans [221], [184], [222], [223], [185] utilisent un modèle probabiliste pour la construction de la matrice $\boldsymbol{D}$ représentant le dictionnaire. Le dictionnaire peut classiquement être initialisé de deux manières :

- Les colonnes représentant les atomes sont initalisées aléatoirement, sous réserve que la norme de chacune soit constante ;
- Les colonnes sont initialisées avec des échantillons d'apprentissage.

Le modèle proposé suggère que, pour chaque exemple $\boldsymbol{y}$ de la base d'apprentissage, la relation

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{v}$$

soit vraie avec une représentation parcimonieuse $\boldsymbol{x}$ et un vecteur de résidu $\boldsymbol{v}$ de moyenne nulle et de variance σ^2. Étant donné la matrice d'exemples $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^{N}$,

ces travaux considèrent la fonction de vraisemblance $\mathcal{P}(\boldsymbol{Y}|\boldsymbol{D})$ et cherchent donc à obtenir le dictionnaire la maximisant.

La formalisation du problème conduit dans [222] à l'équation :

$$D = \arg\max_{\boldsymbol{D}} \sum_{i=1}^{N} \max_{\boldsymbol{x}_i} \{\mathcal{P}(\boldsymbol{y}_i, \boldsymbol{x}_i|\boldsymbol{D})\} \tag{4.5}$$

$$= \arg\min_{\boldsymbol{D}} \sum_{i=1}^{N} \min_{\boldsymbol{x}_i} \{\|\boldsymbol{D}\boldsymbol{x}_i - \boldsymbol{y}_i\|^2 + \lambda\|\boldsymbol{x}\|_1\} \tag{4.6}$$

Cette formalisation pénalise les entrées de $\boldsymbol{x}_i$ mais pas celles de $\boldsymbol{D}$. Il arrive donc que les entrées du dictionnaire augmentent pour que les coefficients tendent vers zéro. Ce problème a été contourné en contraignant les normes des colonnes de $\boldsymbol{D}$ à avoir une norme l^2 constante.

La méthode se présente sous forme itérative. À chaque itération, deux étapes successives sont appliquées :

- calcul des coefficients de $\boldsymbol{x}_i$ par descente de gradient, et
- mise à jour du dictionnaire selon [223]

$$\boldsymbol{D}^{(t+1)} = \boldsymbol{D}^{(t)} - \eta \sum_{i=1}^{N} \left(\boldsymbol{D}^{(t)}\boldsymbol{x}_i - \boldsymbol{y_i}\right) \boldsymbol{x}_i^T \tag{4.7}$$

Ce type d'approche a également été utilisé dans [96], [162], [95], [97] et [213].

La méthode MOD

La technique d'apprentissage de dictionnaire MOD (pour « *Modeling of Optimal Directions* »), est présentée par Engan *et al.* [96], [95], [97]. L'apport principal de cette méthode (par rapport à celle présentée plus haut) est la façon simple de mettre à jour les atomes du dictionnaire.

Pour un ensemble d'apprentissage dont les codes parcimonieux sont connus, l'erreur peut être définie par $\boldsymbol{e}_i = \boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{x}_i$. L'erreur quadratique moyenne pour l'ensemble d'apprentissage est ainsi définie par :

$$\|\boldsymbol{E}\|_F^2 = \|[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N]\|^2 = \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 \tag{4.8}$$

où les échantillons $\boldsymbol{y}_i$ sont les colonnes de $\boldsymbol{Y}$ (même chose pour $\boldsymbol{X}$), et $\|E\|_F$ représente la norme de Frobenius ($\|E\|_F = \sqrt{\sum_{ij} E_{ij}^2}$).

La recherche du minimum de l'équation (4.8) permet de mettre à jour le dictionnaire. Cette recherche s'effectue en dérivant l'équation (4.8) selon $\boldsymbol{D}$ pour obtenir

$$(\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X})\,\boldsymbol{X}^T = 0$$

et ainsi :

$$\boldsymbol{D}^{(t+1)} = \boldsymbol{Y}\boldsymbol{X}^{(t)^T}\left(\boldsymbol{X}^{(t)}\boldsymbol{X}^{(t)^T}\right)^{-1} \tag{4.9}$$

MOD est en fait assez proche de la méthode de Olshausen et Field décrite plus haut. En effet, l'équation (4.7) peut être écrite

$$\begin{aligned}
\boldsymbol{D}^{(t+1)} &= \boldsymbol{D}^{(t)} + \eta\boldsymbol{E}\boldsymbol{X}^{(t)^T} && (4.10)\\
&= \boldsymbol{D}^{(t)} + \eta\left(\boldsymbol{Y} - \boldsymbol{D}^{(t)}\boldsymbol{X}^{(t)}\right)\boldsymbol{X}^{(t)^T} && (4.11)\\
&= \boldsymbol{D}^{(t)}\left(\boldsymbol{I} - \eta\boldsymbol{X}^{(t)}\boldsymbol{X}^{(t)^T}\right) + \eta\boldsymbol{Y}\boldsymbol{X}^{(t)^T} && (4.12)
\end{aligned}$$

Lorsque $t \to +\infty$ et que η est suffisamment faible, l'état obtenu par l'équation (4.7) devient exactement la matrice mise à jour de MOD (Équation 4.9). De plus, dans la méthode de Olshausen et Field, les coefficients de $\boldsymbol{x}_i$ sont obtenus par descente de gradient, ce qui permet « seulement » de s'approcher de la solution optimale.
À noter que dans les deux méthodes, une étape de normalisation des colonnes de $\boldsymbol{D}$ est nécessaire.

L'approche du Maximum *A Posteriori*

Dans [162], [97], [213], et [163], un point de vue probabiliste similaire au maximum de vraisemblance est adopté. Cependant, plutôt que de considérer la fonction de vraisemblance $\mathcal{P}(\boldsymbol{Y}|\boldsymbol{D})$, l'*a posteriori* $\mathcal{P}(\boldsymbol{D}|\boldsymbol{Y})$ est utilisé. Via la règle de Bayes, il est obtenu $\mathcal{P}(\boldsymbol{D}|\boldsymbol{Y}) = \mathcal{P}(\boldsymbol{Y}|\boldsymbol{D})\mathcal{P}(\boldsymbol{D})$. L'expression de vraisemblance précédemment utilisée peut ainsi être augmentée de l'*a priori* $\mathcal{P}(\boldsymbol{D})$.

Lorsqu'aucun *a priori* n'est choisi, la formule de mise à jour revient à celle utilisée par Olshausen et Field (Équation 4.7). Un *a priori* contraignant $\boldsymbol{D}$ à avoir une norme de Frobenius fixe revient à :

$$\boldsymbol{D}^{(t+1)} = \boldsymbol{D}^{(t)} + \eta\boldsymbol{E}\boldsymbol{X}^T + \eta.trace\left(\boldsymbol{X}\boldsymbol{E}^T\boldsymbol{D}^{(t)}\right)\boldsymbol{D}^{(t)}$$

Les deux premiers termes sont les mêmes que dans l'équation (4.7), le dernier terme compensant les déviations de la contrainte. Cependant un tel *a priori* permet aux colonnes d'avoir des normes différentes, certaines ont ainsi de faibles normes et ont donc tendance à être sous–utilisées.

Ainsi, un deuxième *a priori* a été défini. Il permet de contraindre les colonnes à avoir une norme l^2 unitaire. La nouvelle formule de mise à jour devient donc :

$$\boldsymbol{d}_i^{(t+1)} = \boldsymbol{d}_i^{(t)} + \eta\left(\boldsymbol{I} - \boldsymbol{d}_i^{(t)}\boldsymbol{d}_i^{(t)^T}\right)\boldsymbol{E}\boldsymbol{x}_i^T$$

où $\boldsymbol{x}_i^T$ est la i^{me} colonne de $\boldsymbol{X}^T$. Comparé à MOD, ces algorithmes d'apprentissage sont coûteux en temps de calcul. Des tests sur des données synthétiques sont présentés dans [162], [97], [213], [163].

L'union de Bases Orthonormées

Dans [183], le dictionnaire considéré est composé d'une union de bases orthonormées

$$\boldsymbol{D} = [\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_L]$$

où $\boldsymbol{D}_j \in \mathbb{R}^{n \times n}$, $j = 1, 2, \ldots, L$ sont des matrices orthonormées. Les coefficients de la décomposition $\boldsymbol{X}$ peuvent être décomposés en L morceaux, chacun se référant à une base orthonormée différente :

$$\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_L]^T$$

où $\boldsymbol{X}_i$ correspond à la matrice contenant les coefficients de la décomposition relative au dictionnaire $\boldsymbol{D}_i$. La méthode de poursuite utilisée dans [183] est fondée sur l'algorithme de relaxation par blocs [252]. Cela revient à effectuer des itérations de seuillage pour chaque $\boldsymbol{X}_i$ selon $\boldsymbol{D}_i$ tandis que les autres morceaux de $\boldsymbol{X}$ restent fixes.

Une fois les coefficients de la décomposition calculés, la mise à jour de $\boldsymbol{D}_i$ est effectuée en calculant dans un premier temps la matrice de résidus :

$$\boldsymbol{E}_i = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N] = \boldsymbol{Y} - \sum_{j \neq i} \boldsymbol{D}_j \boldsymbol{X}_j$$

Dans un second temps, une décomposition en valeurs singulières ($\boldsymbol{E}_i \boldsymbol{X}_i^T = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{V}^T$) et la mise à jour de la $i^{ème}$ base orthonormée ($\boldsymbol{D}_i = \boldsymbol{U} \boldsymbol{V}^T$) sont réalisées.

L'approche K–SVD

L'approche K–SVD, utilisée dans cette thèse, est une généralisation de la technique de clustering des K–means. L'algorithme des K–means [118] essaie de réaliser un mapping de vecteurs $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^N$ sur un ensemble de codes $\boldsymbol{C}$ comprenant K codes ($N \gg K$) par plus proche voisin. Il peut être formalisé ainsi :

$$\min_{\boldsymbol{D}, \boldsymbol{X}} \{\|\boldsymbol{Y} - \boldsymbol{C}\boldsymbol{X}\|_F^2\} \quad \text{sachant que } \forall i, \boldsymbol{x}_i = \boldsymbol{e}_k \text{ pour un certain } k$$

où $\boldsymbol{e}_k$ représente un vecteur dont toutes les entrées sont nulles sauf la $k^{ème}$ qui est 1 (c'est un cas extrême de parcimonie).

Dans l'approche K-SVD, le signal peut être décomposé en une combinaison linéaire d'atomes, on obtient donc la formalisation suivante :

$$\min_{\boldsymbol{D}, \boldsymbol{X}} \{\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2\} \quad \text{sachant que } \forall i, \|\boldsymbol{x}_i\|_0 \leq T_0$$

où T_0 représente le nombre maximal d'atomes permis pour la décomposition.

L'apprentissage se décompose en deux étapes :

(1) calcul de la décomposition du signal sur le dictionnaire via une approche de poursuite (et en autorisant au plus T_0 coefficients non nuls) ;

(2) mise à jour des atomes selon la décomposition obtenue en (1), ainsi que des coefficients obtenus en (1) pour représenter le signal utilisant les atomes correspondants.

Ici, les vecteurs parcimonieux $\boldsymbol{X}$ ne sont donc pas fixes lors de l'étape de mise à jour du dictionnaire.

La mise à jour des atomes $\boldsymbol{d}_k$ peut se réduire à trouver l'approximation au rang un de la matrice de résidus :

$$\boldsymbol{E}_k = \boldsymbol{Y} - \sum_{j \neq k} \boldsymbol{d}_j \boldsymbol{x}^j$$

où $\boldsymbol{x}^j$ est la $j^{ème}$ ligne de la matrice de coefficients $\boldsymbol{X}$. Cette approximation est réalisée via le calcul des valeurs singulières (SVD). L'algorithme 6 décrit les principales étapes de calcul.

Algorithme 6: Algorithme K–SVD.

Entrées : Ensemble d'apprentissage $\boldsymbol{Y} = \{\boldsymbol{y}_i \in \mathbb{R}^N\}$, $\boldsymbol{Y} \in \mathbb{R}^{N \times L}$
Sorties : Dictionnaire $\boldsymbol{D} \in \mathbb{R}^{N \times M}$
Initialiser les colonnes $\boldsymbol{d}_m$ de Φ aléatoirement, avec $\|\boldsymbol{d}_m\|_2 = 1$
répéter

Étape de décomposition parcimonieuse :
Utiliser un algorithme de décomposition pour calculer $\boldsymbol{x}_i$

$$\min_{\boldsymbol{x}_i} \|\boldsymbol{x}_i\|_0 \quad \text{sachant que } \boldsymbol{y}_i = \sum_{m=1}^{M} (x_m)_i \boldsymbol{d}_m \quad \forall i = 1, 2, \ldots, L$$

Étape de mise à jour du dictionnaire :
pour $m = 1, 2, \ldots, M$ **faire**

- Définir le groupe d'échantillons utilisant $\boldsymbol{d}_m$,

$$\omega_m = \{i | 1 \leq i \leq M, \boldsymbol{x}_i(m) \neq 0\}$$

- Calculer $\boldsymbol{E}_m = \boldsymbol{Y} - \sum_{j \neq m} \boldsymbol{D}_j \boldsymbol{X}_j^T$
- Restreindre $\boldsymbol{E}_m$ aux colonnes correspondant aux éléments de ω_m pour obtenir $\boldsymbol{E}_m^R$
- Appliquer une décomposition en valeurs singulières (SVD) $\boldsymbol{E}_m^R = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^T$
- Mise à jour $\boldsymbol{d}_m = \boldsymbol{u}_1$, $\boldsymbol{x}_R^m = \boldsymbol{\Lambda}(0,0)\boldsymbol{v}_0$

jusqu'à *Convergence*
retourner $\boldsymbol{D}$

À noter que l'apprentissage de dictionnaires invariants a été étudié dans [195]. Les auteurs y proposent un algorithme pour l'apprentissage de dictionnaires invariants aux translations. La méthode pour l'apprentissage est cependant plus complexe.

Une variante de l'algorithme K–SVD a été proposée dans [196] pour réaliser un apprentissage *online* du dictionnaire. Cette méthode permet ainsi d'apprendre des dictionnaires à partir de flux vidéo (par exemple) où tous les échantillons d'apprentissage ne sont pas forcément disponibles au début. Elle permet en outre de traiter des problèmes d'apprentissage où le nombre d'échantillons est très grand (de l'ordre du million). En effet, les méthodes « classiques » nécessitent souvent le calcul de l'inverse d'une matrice (ou d'une pseudo inverse), ce qui devient coûteux lorsque les matrices deviennent trop grandes.

4.5 Classification via des représentations parcimonieuses

4.5.1 Pourquoi la parcimonie ?

Le problème d'identification consiste à trouver l'identité d'une personne parmi une base de données de personnes, la base de données contenant l'empreinte biométrique de la personne test. La démarche classique consiste à calculer l'empreinte biométrique de la personne, puis à la comparer aux empreintes biométriques de la base de données. Sous l'hypothèse que la personne p a été initialement enrôlée, les deux empreintes biométriques de p doivent correspondre.

Soit la matrice $\boldsymbol{D} \in \mathbb{R}^{N \times M}$ dont les colonnes $\boldsymbol{d}_k \in \mathbb{R}^N$ correspondent aux empreintes biométriques des M personnes enrôlées. Avec un système biométrique idéal, une empreinte biométrique test $\boldsymbol{y}$ doit satisfaire :

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}$$

avec $\boldsymbol{x} \in \mathbb{R}^M$ un vecteur dont toutes les entrées sont nulles sauf l'entrée correspondant à l'identité de $\boldsymbol{y}$ qui vaut 1. Pour un tel système idéal, le vecteur $\boldsymbol{x}$ serait ainsi très parcimonieux.

Dans la réalité, les empreintes biométriques sont obtenues via des algorithmes d'extraction de caractéristiques. Ceux–ci ne sont pas forcément robustes à tous les changements que peut subir une image, ainsi deux empreintes biométriques d'une même personne donnent bien souvent (pour ne pas dire toujours) deux vecteurs de caractéristiques différents (variance intra–classe). Pire, deux images de personnes différentes peuvent donner deux vecteurs de caractéristiques proches (variance inter–classe).

L'identification d'une personne p à partir d'une image I_p se résume à trois étapes :

– Calcul de l'empreinte biométrique $\boldsymbol{y}$ de I_p ;
– Recherche parmi les empreintes de la base d'enrôlés $\boldsymbol{D}$ l'empreinte $\boldsymbol{d}_f$ correspondant le plus à $\boldsymbol{y}$;
– Retour de l'identité correspondant à $\boldsymbol{d}_f$.

Ainsi, la recherche de l'identité peut se formaliser ainsi :

$$\boldsymbol{d}_f = \min_k \|\boldsymbol{d}_k - \boldsymbol{y}\|_2^2 \quad \forall k$$

avec $\|\boldsymbol{d}_k - \boldsymbol{y}\|_2^2$ étant la mesure de dissimilarité de deux empreintes biométriques.

L'introduction de la parcimonie lors de l'identification d'une personne permet de considérer le problème différemment. Ainsi, plutôt que de mesurer la dissimilarité entre deux empreintes biométriques, il est préférable de mesurer la similarité entre deux empreintes, en injectant un *a priori* parcimonieux. En effet, un vecteur de similarité représentant la similarité entre une empreinte et une base d'empreintes et qui comporterait beaucoup de zéros permettrait d'écarter d'office les empreintes (et donc les identités) associées (voir section 4.5.2).

La phase d'extraction de caractéristiques peut également bénéficier de l'apport de la parcimonie. En effet, la recherche de caractéristiques parcimonieuses peut renforcer l'aspect discriminant des caractéristiques extraites, et ainsi, être plus robustes aux modifications dûes à la variance intra–classe. Une approche d'extraction de caractéristiques fondée sur un apprentissage de dictionnaire et une décomposition parcimonieuse des parties des images de visage a été mise en œuvre lors de cette thèse, elle sera présentée en détail ultérieurement (Chapitre 5, section 5.3).

4.5.2 Approche « *Sparse Representation–based Classification* »

Dans cette section, nous résumons l'approche SRC (pour « *Sparse Representation–based Classification* ») initialement présentée par Wright *et al.* dans [298]. Cette approche repose sur une décomposition parcimonieuse d'une image test (ou de son vecteur caractéristique initialement calculé) sur les images (ou les vecteurs caractéristiques extraits des images) de la base de données des personnes déjà enrôlées.

L'idée est qu'en minimisant la norme l^1 lors de la décomposition, de nombreuses identités vont être rejetées *de facto* et la classification n'en sera que plus pertinente. L'approche est schématiquement présentée à la figure 4.4.

Étant donné le vecteur caractéristique $\boldsymbol{y} \in \mathbb{R}^n$ d'une image test, et $\boldsymbol{D} \in \mathbb{R}^{n \times M}$ la matrice dont les colonnes sont les vecteurs caractéristiques des personnes enrôlées, l'approche SRC peut alors se décomposer en 3 étapes :

– Décomposition parcimonieuse de $\boldsymbol{y}$ sur $\boldsymbol{D}$ (la matrice $\boldsymbol{D}$ peut alors être vue comme un dictionnaire dont les atomes sont les vecteurs caractéristiques de la base de données)

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left(\|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_2 + \lambda \|\boldsymbol{x}\|_1 \right)$$

(note : le vecteur $\boldsymbol{x}$ contient beaucoup de zéros) ;
– Calcul du vecteur de résidus $\boldsymbol{r} \in \mathbb{R}^M$ (à partir de $\hat{\boldsymbol{x}}$) correspondant à la dissimilarité entre $\boldsymbol{y}$ et les colonnes de $\boldsymbol{D}$;
– Déduction de l'identité correspondant à $\boldsymbol{y}$ en sélectionnant l'identité correspondant au minimum du vecteur de résidus $\boldsymbol{r}$.

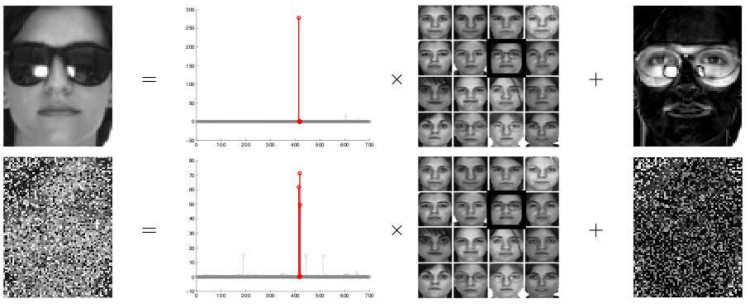


FIGURE 4.4 – Fonctionnement de l'approche SRC (extrait de [298]) où le vecteur caractéristique d'une image serait l'image elle même. Une image est décomposée en une somme d'une image de la base (fois le coefficient associé issu de la décomposition parcimonieuse) et d'un résidu.

La matrice $\boldsymbol{D}$ peut être vue comme un dictionnaire définissant un nouvel espace dans lequel les vecteurs caractéristiques extraits des personnes enrôlées formeraient une base.

Les différentes étapes de l'approche sont résumées à l'algorithme 7 et schématisées à la figure 4.5.

L'étape critique de l'algorithme est la minimisation l^1 de $\boldsymbol{y}$ sur $\boldsymbol{D}$. Cette étape est dans [298] traitée à l'aide d'un algorithme primal–dual. Dans cette thèse, nous avons adopté l'approche fondée sur un seuillage doux itératif.

Algorithme 7: Algorithme SRC.

Entrées : La matrice de la galerie $\boldsymbol{D} \in \mathbb{R}^{N \times M}$, l'empreinte biométrique test $\boldsymbol{y} \in \mathbb{R}^N$

Sorties : L'identité de $\boldsymbol{y}$

- Normaliser les colonnes de $\boldsymbol{D}$ de sorte que $\|d_k\|_2 = 1 \quad \forall k$
- *Décomposer $\boldsymbol{y}$ sur $\boldsymbol{D}$ selon :* $\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} (\|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_2 + \lambda \|\boldsymbol{x}\|_1)$
- *Calcul des résidus* $\boldsymbol{r} = \{r_k\}, k = 1, \ldots, M$ *:* $r_k = \|\boldsymbol{y} - \boldsymbol{D}_k \hat{\boldsymbol{x}}_k\|_2$
- *Déduction de l'identité de $\boldsymbol{y}$:* $identité(\boldsymbol{y}) = \arg\min_k(\boldsymbol{r})$

retourner $identité(\boldsymbol{y})$

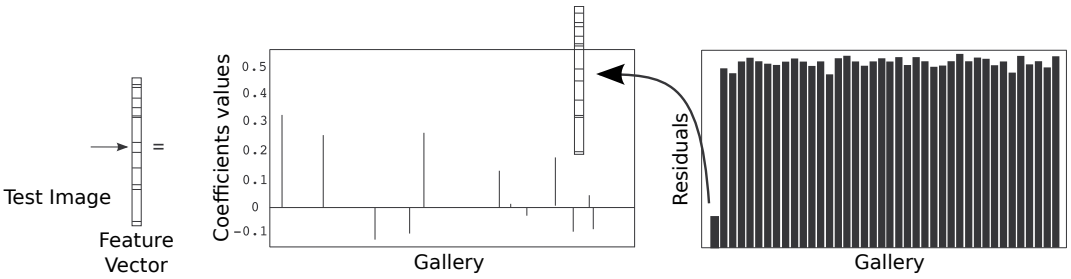


FIGURE 4.5 – Vue schématique de l'approche SRC.

L'approche SRC donne de bons résultats de classification dans le cadre de l'identification [298], [160]. Dans [298], il est notamment montré que le choix de l'extracteur de caractéristiques n'est plus aussi critique qu'avec une classification basée sur les plus proches voisins. En effet, de bons résultats sont obtenus avec des caractéristiques extraites simples (Figure 4.6).

Cet algorithme de classification présente cependant plusieurs défauts :

– la décomposition l^1 d'un vecteur sur le « dictionnaire » nécessite l'utilisation de méthodes de décomposition parcimonieuses (comme celles présentées à la Section 4.3) qui peuvent s'avérer relativement lentes. Dans nos tests, le processus d'identification d'un vecteur de caractéristiques est de l'ordre de 2 secondes. Ces temps d'identification sont évidemment largement dépendants du nombre de personnes dans la base ainsi que de la dimension des vecteurs. Ils restent cependant supérieurs à d'autres approches plus classiques de classification comme les plus proches voisins.
– l'approche ne fonctionne que pour l'identification, étant donné qu'elle nécessite la décomposition du vecteur caractéristique test sur une base. Elle ne peut donc pas être utilisée pour l'authentification.

– dans la version actuelle, l'approche n'est pas ou peu robuste aux rotations, changements de pose, ou mises à l'échelle. Cet aspect peut cependant être contrebalancé par le choix d'un extracteur de caractéristiques robustes à ces modifications de l'image.

FIGURE 4.6 – Caractéristiques utilisées dans [298]. Dans l'ordre : l'image originale, *eigenfaces*, *fisherfaces*, *sous–échantillonnage de taille* 12×10, *random projections*.

4.6 Conclusion

Dans ce chapitre, nous avons présenté une introduction à la théorie de la parcimonie. Un vecteur est dit parcimonieux lorsqu'il contient beaucoup de coefficients nuls. La décomposition parcimonieuse d'un signal en une combinaison linéaire d'éléments de base (ou atomes, regroupés au sein d'un dictionnaire) est un problème difficile et le domaine de recherche s'y rattachant est en constante évolution.

Les principales approches de décomposition d'un signal de façon parcimonieuse ont été présentées ainsi que les dictionnaires canoniques. La problématique de l'apprentissage de dictionnaires permettant d'obtenir de meilleures décompositions parcimonieuses a également été présentée, ainsi que les principaux algorithmes permettant de réaliser de tels apprentissages.

Enfin, une technique récente de classification fondée sur une minimisation l^1 a été détaillée. Celle–ci s'applique avec efficacité pour le problème de l'identification d'une personne dans un contexte biométrique.

Toutes ces techniques ont été appliquées dans la dernière partie de ce manuscrit. Des tests ont été réalisés pour lesquels l'extraction de caractéristiques des images de visages a consisté en une décomposition parcimonieuse de parties de visages sur un dictionnaire préalablement appris. La technique de classification SRC a de plus été mise en œuvre pour une identification d'images pour les modalités visible et infrarouge.

Troisième partie

Résultats expérimentaux

Chapitre 5

Résultats expérimentaux unimodaux

Dans ce chapitre, nous détaillons les principales expériences que nous avons menées durant la thèse ainsi que les résultats unimodaux (sur une seule modalité à la fois) obtenus via les trois principales approches testées :

- les réseaux de neurones convolutionnels,
- les réseaux de neurones convolutionnels pour lesquels un préapprentissage a été réalisé,
- et enfin l'approche basée uniquement sur la parcimonie.

Les deux principaux modes de test d'un système biométrique sont l'*authentification* (ou *vérification*) et l'*identification*. L'*authentification* consiste pour le système à vérifier l'identité clamée par une personne se présentant devant le système, c'est une comparaison $1-1$. L'*identification* consiste à retrouver l'identité d'une personne parmi une base de personnes préalablement enrôlées, c'est une comparaison $1-n$. L'utilisation d'un système biométrique en *authentification* nécessite le calcul d'un score de similarité entre deux empreintes biométriques, la décision étant ensuite prise en fonction de ce score et d'un seuil. L'utilisation d'un système en *identification* nécessite le calcul de n scores de similarité. Le calcul du *rang* pour une image test, c'est–à–dire le moment à partir duquel la bonne identité est retrouvée, permet d'évaluer qualitativement le système biométrique.

Toutes les expériences menées sont en *identification* , c'est–à–dire que le système doit retrouver l'identité de la personne testée. Celles–ci ont été réalisées sous l'hypothèse forte que la personne testée a préalablement été enrôlée. Nous avons ainsi pu nous affranchir de la notion de seuil inhérente à l'*authentification*.

Dans la suite, nous utilisons le terme *galerie* pour désigner les images utilisées lors de l'enrôlement des personnes, et le terme *probe* pour désigner les images utilisées pour tester la méthode d'identification.

Bases de données utilisées Les expériences ont été menées sur certaines bases spécifiques (cf. chapitre 1) :

- La base de données AT&T (anciennement ORL)[9] est composée de 40 personnes, chacune d'elles ayant 10 images. Les images présentent des variations d'illumination, de pose et des artefacts (lunettes, barbes, ...).
- La base de données FERET [10] est composée de près de 1200 personnes pour un total de plus de 14000 images. Celles–ci présentent de nombreuses variations de luminosité, de pose ou encore d'artefacts.
- La base principalement utilisée pour les tests est la la base de données de l'université de Notre–Dame (collection $X1$) [11]. Composée de 590 personnes, cette base de données présente à nos yeux deux avantages cruciaux :
  - elle propose des visages dans les modalités visible et infrarouge (grandes longueurs d'ondes) capturés au même instant,
  - elle dispose d'un protocole bien défini permettant une comparaison précise avec des résultats publiés précédemment sur cette base.

La base Notre–Dame est divisée en deux parties : la première partie, appelée *Ensemble d'apprentissage* (**TrS** dans la suite), est composée de 159 personnes, chacune ayant une seule image visible et son équivalent infrarouge. La deuxième partie, appelée *Ensemble de test* (**TeS** dans la suite), est composée de 82 personnes pour un total de 2292 images visibles et 2292 images infrarouges.

Alors que **TrS** ne contient ni expressions faciales, ni variations de pose ou de luminosité, **TeS** contient de nombreuses images contenant des variations de luminosité, d'expression faciale, de pose et de distribution thermique.

Deux expérimentations appelées *Same–session* et *Time–lapse* ont été conçues pour tester respectivement les problèmes de luminosité et la reconnaissance à travers le temps. Pour chacune de ces expérimentations, des fichiers livrés avec la base permettent la construction des galeries et des probes.

Pour l'expérimentation *Same–session*, quatre fichiers nommés *F{A,B}L{F,M}* permettent de construire les listes d'images servant de galerie ou de probe (chaque fichier contient une seule image pour chacune des 82 personnes). Ainsi, ces jeux de tests, conduisent à 12 sous–expérimentations (par exemple le fichier *FALM* peut être utilisé pour lister les images composant la galerie, et le fichier *FBLF* peut être utilisé pour lister les images composant le probe).

La terminologie de ces fichiers dénotent les particularités des images associées et donc les caractéristiques que l'on souhaite tester. Celle–ci est :

- *FA* où les visages ont une expression faciale neutre,
- *FB* où les visages ont une expression faciale souriante,
- *LF* où les visages sont capturés avec l'illumination *Feret style Lighting*,
- *LM* où les visages sont capturés avec l'illumination *Mugshot Lighting*.

Ces jeux de tests permettent donc de tester différentes combinaisons, comme par exemple :

- un enrôlement où les visages ont une expression faciale neutre, et les visages de tests ont une expression faciale souriante (galerie : FALF, probe : FBLF ou FBLM),
- les visages d'enrôlement et de test ont la même expression faciale, mais la luminosité est changée (galerie : FALF, probe : FALM)
- ...

Pour l'expérience *Time–lapse*, quatre fichiers de galerie (nommés *G_F{A,B}L{F,M}*) permettent de construire quatre galeries différentes (chaque fichier contient une seule image pour chacune des 82 personnes). Quatre fichiers de probe (nommés *P_F{A,B}L{F,M}*) permettent de construire quatre probes différents (chaque fichier contient plusieurs images pour chacunes des 82 personnes). Ces jeux de tests conduisent ainsi à 16 sous–expérimentations. En effet, les fichiers *G_FALF* et *P_FALF* ne listant pas les mêmes images, la sous–expérimentation avec ces deux fichiers comme galerie et probe est valide.

Outre toutes les possibilités offertes par ces différentes listes d'images, les expérimentations *Same–session* et *Time-lapse* permettent de différentier les résultats selon que les images constituant la galerie et le probe ont été capturées à quelques secondes d'intervalle (expérimentation *Same–session*) ou à plusieurs jours (voire semaines) d'intervalle (expérimentation *Time–lapse*).

Notons enfin que les galeries créées à partir des fichiers définissant le protocole ne comportent qu'une seule image par personne (scénarios dits *one image to enroll*).

La base de données Notre–Dame a été largement utilisée car elle possède un protocole bien défini, ce qui est un atout certain pour la comparaison à de précédents travaux, pour peu que ceux–ci suivent le protocole de test. Certains travaux [33] [260] [261] utilisent cette base de données et proposent des algorithmes pour la reconnaissance ainsi que pour la fusion des modalités visible et infrarouge. Certaines entorses au protocole (explicitées par les auteurs) sont cependant faites, ce qui ne permet pas la comparaison directe avec les résultats présentés dans ce chapitre.

Dans [33], les fusions au niveau pixels et au niveau caractéristiques issues d'une ACP sont considérées via des algorithmes génétiques. Les auteurs de [260] et [261] considèrent l'utilisation de multiples Machines à Support de Vecteurs (SVM) [260], ou de transformées en ondelettes [261] pour réaliser la fusion des images visible et infrarouge.

5.1 Résultats avec les Réseaux de Neurones Convolutionels

L'approche basée sur les réseaux de neurones convolutionels mise en œuvre lors de cette thèse repose sur le modèle dit *diabolo* [256] où la sortie souhaitée du réseau est identique à l'entrée, avec une couche intermédiaire de faible dimension. Le réseau apprend ainsi une représentation compacte de l'entrée. En appliquant certaines transformations au vecteur d'entrée sans changer la sortie désirée, le réseau est ainsi capable d'apprendre une représentation compacte invariante à ces transformations. Inspiré des travaux de Duffner et Garcia [90], le Réseau de Reconstruction utilisé fonctionne comme un réseau diabolo. Il projette de façon non–linéaire l'entrée sur un sous–espace puis reconstruit l'image d'un visage de *référence* choisi préalablement.

Le réseau de reconstruction (voir Figure 5.1) prend en entrée une image de taille 56×46 (i.e. : la taille de la rétine du réseau) et la passe dans une succession de couches de convolution C_i, subsampling S_i et de neurones complètement connectés F_i de type *Multi–Layer Perceptron* (MLP). La sortie du réseau est une image, de même taille que l'entrée, qui est reconstruite par la dernière couche F_7. Chaque pixel de la sortie est représenté par un neurone, il y a donc $56 \times 46 = 2576$ neurones sur la dernière couche.

Notre architecture, similaire à celle de Duffner et Garcia [90] a été modifiée notamment en augmentant le nombre de neurones par couches. Cette modification a été réalisée car nous voulions utiliser la même architecture pour les deux modalités, et l'architecture proposée dans [90] n'était pas adaptée à la modalité infrarouge étant donnés les trop grandes variations locales des images. Plus précisément, l'architecure est :

- $Input$. Nombre d'images : 1. Taille : 56×46.
- C_1. Nombre de cartes : 15 ; Taille des noyaux : 7×7 ; Taille des cartes : 50×40. Toutes les cartes sont connectées à l'entrée.
- S_2. Nombre de cartes : 15 ; Taille des noyaux : 2×2 ; Taille des cartes : 25×20. Connexions $1 - 1$.
- C_3. Nombre de cartes : 45 ; Taille des noyaux : 6×6 ; Taille des cartes : 20×15. Connexions partielles pour casser la symétrie (Table B.2).
- S_4. Nombre de cartes : 45 ; Taille des noyaux : 4×3 ; Taille des cartes : 5×5. Connexions $1 - 1$.
- C_5. Nombre de cartes : 250 ; Taille des noyaux : 5×5 ; Taille des cartes : 1×1. Couche complètement connectée à S_4.
- F_6. Nombre de cartes : 100 ; Couche complètement connectée à C_5.
- F_7. Nombre de cartes : 2576 ; Couche complètement connectée à F_6.

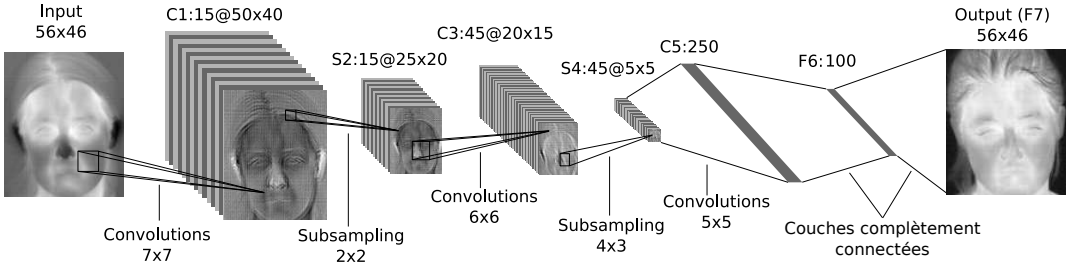


FIGURE 5.1 – Architecture du réseau de reconstruction.

Une telle architecture permet d'apprendre une représentation compacte (vecteur extrait de la couche F_6) de l'entrée. En forçant le réseau à reconstruire toujours la même image pour une personne quelque soit les dégradations de l'image d'entrée, le réseau est « forcé » à apprendre une représentation compacte et invariante à ces dégradations. Le réseau peut ainsi être vu comme la composition de deux parties :

- la première partie composée des couches C_1 à F_6 permettant de réaliser une projection robuste aux dégradations de l'entrée sur un sous–espace (de dimension 100),
- la seconde partie composée de la couche de sortie permettant la reconstruction d'une image particulière pour chaque personne à partir du projeté précédemment obtenu.

Notons que la dernière couche n'est « utile » que lors de l'apprentissage pour « guider » la projection que l'on cherche à obtenir. Une fois l'apprentissage réalisé, la couche de sortie devient « inutile », le résultat de la projection étant le vecteur obtenu à la couche F_6. La reconstruction (couche F_7), outre l'ajout de calculs supplémentaires, n'a plus de sens dès lors que l'image d'entrée correspond à une personne qui n'était pas dans l'ensemble d'apprentissage. En effet, les vecteurs de reconstruction (poids de la couche F_7) correspondent aux vecteurs de reconstruction des personnes de la base d'apprentissage, et ne peuvent ainsi reconstruire toute autre personne.

Le réseau est entraîné à extraire des caractéristiques de bas niveau sur les deux premières couches (couches C_1 et S_2), puis de fusionner ces caractéristiques pour obtenir des caractéristiques de plus haut niveau (couches C_3 et S_4). Les couches suivantes (couches C_5 et F_6) se chargent de la classification des caractéristiques extraites et de la projection finale.

5.1.1 Résultats préliminaires en Visible

Afin de valider l'approche et de tester sa capacité de généralisation, nous avons conduit une expérimentation sur la base de données AT&T. Cette base de données est composée de 10 images pour chacune des 40 personnes. Les images présentent

des variations de pose, d'illumination, d'expression faciale ou encore d'accessoires (lunettes de vue présentes/absentes).

Les images ont été centrées manuellement par rapport aux yeux de sorte que tous les visages aient leurs yeux approximativement au même endroit dans l'image (Figure 5.2). Elles ont été réduites à une taille de 56×46 (la taille de la rétine du réseau). Les valeurs des pixels de l'image ont été normalisées de sorte que leur moyenne soit de 0 et leur variance de 1, ceci pour assurer une meilleure convergence lors de l'apprentissage.

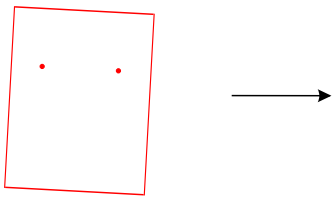


FIGURE 5.2 – Recadrage géométrique des images de visages par rapport aux yeux.

Nous avons partitionné la base en deux ensembles disjoints, le premier appelé SET_1 comprenant 35 personnes, le second appelé SET_2 comprenant les 5 personnes restantes. Ainsi SET_1 contient 350 images tandis que SET_2 en contient 50. Dans la suite, SET_1 est utilisé pour l'apprentissage du réseau, et SET_2 pour les tests. Avec un tel partitionnement, nous pouvons ainsi tester la capacité de généralisation du réseau sur des personnes « inconnues », c'est à dire des personnes n'ayant pas été utilisées pour l'apprentissage.

Apprentissage Pour chaque personne p de SET_1, l'image moyenne I_p a été calculée, et l'image de cette personne la plus proche (au sens de l'erreur quadratique moyenne) de la moyenne a été sélectionnée pour être l'image de référence I_{pt}. Cette image I_{pt} est celle qui va ensuite être utilisée comme « cible » pour p lors de l'apprentissage, i.e. l'image que le réseau va essayer de reconstruire.

Une image par personne (différente de l'image de référence) est ensuite retirée de l'ensemble d'apprentissage. Ces 35 images vont former l'ensemble de validation, pour réaliser la validation croisée lors de l'apprentissage. Cette validation croisée a lieu après chaque passage de l'ensemble d'apprentissage pour stopper l'apprentissage et ainsi éviter, entre autre, l'effet de surapprentissage.

La fonction de coût utilisée lors de l'apprentissage est la fonction classique de régression :

$$E = \|\boldsymbol{o}_p - \boldsymbol{t}_p\|_2^2$$

où $\boldsymbol{o}_p$ et $\boldsymbol{t}_p$ sont les vecteurs de sortie du réseau et les vecteurs cible respectivement pour la personne p.

Protocole de test Les tests sont effectués sur SET_2. Pour une image test I_{test} de la personne p, le protocole de test peut être résumé :

- construction du modèle pour chaque personne de la base,
- comparaison de l'image test I_{test} à chaque modèle,
- sélection du modèle le plus proche de I_{test}.

La construction du modèle pour une personne de la base consiste à projeter toutes ses images (i.e. récupération des vecteurs issus de la couche F_6), puis à moyenner ces vecteurs. Ainsi, pour chaque personne différente de p, le modèle est construit à partir de 10 vecteurs. Étant donné que le vecteur projeté de I_{test} ne peut honnêtement pas concourir à la création du modèle de la personne p, le modèle de p n'est construit qu'à partir des 9 images restantes.

À des fins de comparaisons, nous avons utilisé le même protocole de test avec une approche basée sur une analyse en composantes principales. Les 35 personnes de SET_1 ont été utilisées pour calculer le sous–espace (défini à partir des vecteurs propres représentant plus de 95% de l'énergie totale des valeurs propres associées).

Les correspondances cumulées pour les deux approches sur SET_2 sont résumées au tableau 5.1. Le *rang* pour une image de test représente le nombre de personnes faussement identifiées avant que le système ait trouvé la bonne identité. La méthode des *eigenfaces* n'est ici pas optimale étant donné son caractère linéaire. De meilleurs résultats peuvent être obtenus à l'aide de techniques de réduction de dimension non linéaire telles les méthodes Isomap ou LLE (voir annexe A).

Nous avons appliqué le même protocole en considérant toutes les images de la base comme images potentielles de test. Les résultats sont présentés à la figure 5.3. Les taux de reconnaissance au rang 0 sont supérieurs aux 76% et 58% obtenus précédemment (voir tableau 5.1) étant donné que certaines images de test ont été utilisées pour l'apprentissage.

Afin de tester les capacités de généralisation du réseau sur une autre base, nous avons conduit une expérimentation sur un sous–ensemble de la base FERET composé de 2409 images de 867 personnes. Afin d'éviter de nouveaux apprentissages sur cette base, nous avons réutilisé les poids du réseau issus de l'expérimentation précédente sur la base AT&T. Nous avons également comparé les résultats avec la méthode des Eigenfaces dans les mêmes conditions (en réutilisant les vecteurs propres issus de l'expérimentation précédente).

Rang	Réseau de Reconstruction		Eigenfaces	
0	38	(76%)	29	(58%)
1	45	(90%)	33	(66%)
2	45	(90%)	38	(76%)
3	47	(94%)	40	(80%)
4	47	(94%)	42	(84%)
5	49	(98%)	44	(88%)
6	50	(100%)	44	(88%)

TABLE 5.1 – Correspondances cumulées sur SET_2 pour chaque approche, le taux de reconnaissance est entre parenthèses. La dernière correspondance pour l'approche *Eigenfaces* (ACP) est au rang 23.

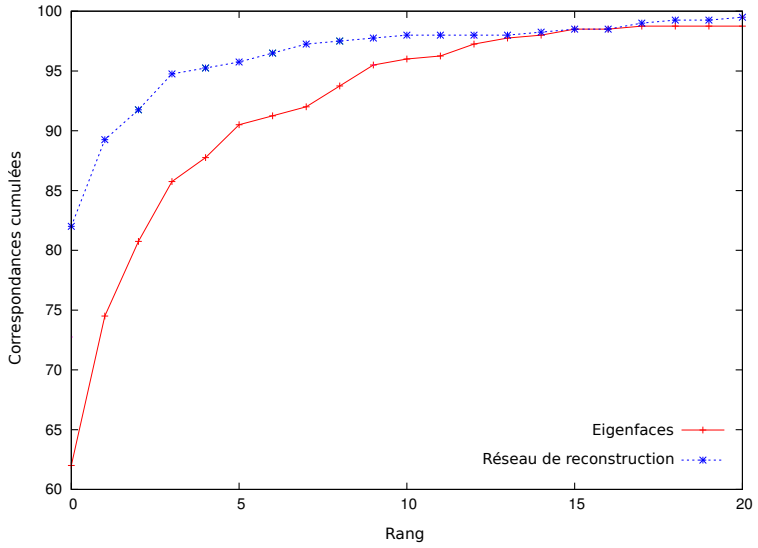


FIGURE 5.3 – Courbe ROC pour les méthodes Eigenfaces et Réseau de reconstruction sur l'ensemble de la base AT&T.

Les taux de reconnaissance au rang 0 sont de $68,3\%$ pour le réseau de reconstruction et de 37% pour la méthode des Eigenfaces. Ce petit test montre que l'approche basée sur les réseaux de neurones convolutionnels a une bonne capacité de généralisation étant donné que nous avons réalisé l'apprentissage sur 35 personnes, et testé sur 867. De plus les deux bases ne présentent pas les mêmes challenges : tandis que la base AT&T présente quelques variations de pose, les visages de la base FERET présentent quelques expressions faciales qui sont absents de la base AT&T, le réseau n'ayant donc pas pu apprendre à y être invariant. Les conditions d'illumination sont de plus différentes entre les deux bases ce qui explique le mau-

vais score de l'approche Eigenfaces, cette méthode étant sensible aux conditions de luminosité. Les courbes ROC pour les deux approche sont présentées à la figure 5.4.

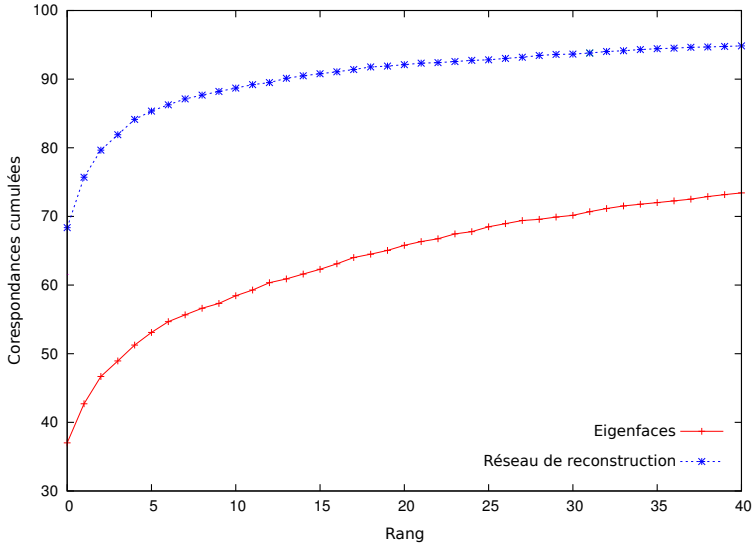


FIGURE 5.4 – Courbe ROC pour les méthodes Eigenfaces et Réseau de reconstruction sur la base FERET (Apprentissage réalisé sur la base AT&T).

5.1.2 Résultats préliminaires en Infrarouge

Afin de tester l'approche basée sur les réseaux de neurones convolutionnels sur la modalité infrarouge, nous avons utilisé une partie de la base Notre–Dame. Nous nous sommes limités à un sous–ensemble de celle–ci contenant 870 images infrarouges de 26 personnes différentes, avec une grande variation du nombre d'images disponibles par personne, allant de 4 à 40. Les images présentent de plus des variations de pose ainsi que des changements de chaleur pour certaines parties du visage (typiquement les oreilles ou le nez).

Les images ont été prétraitées de la même façon que précédemment : centrage manuel par rapport aux yeux, taille réduite à une taille de 56×46, valeurs des pixels normalisées (moyenne nulle, et variance unitaire).

De la même manière que précédemment, la base de données a été divisée en deux parties disjointes SET_1 et SET_2. SET_1 contient les images de 20 personnes (totalisant 736 images), tandis que SET_2 contient les images restantes (134 images de 6 personnes). L'apprentissage a été réalisé avec SET1, et SET2 a été utilisé pour les tests.

Les protocoles d'apprentissage et de tests utilisés sont similaires à ceux utilisés sur le visible (voir plus haut).

Rang	Correspondances cumulées	%
0	115	$85,8\%$
1	129	$96,2\%$
2	132	$98,5\%$
3	134	100%

TABLE 5.2 – Rangs cumulatifs obtenus pour les images de SET_2.

Les résultats en identification sont présentés au tableau 5.2.

Ces résultats préliminaires pour les modalités visible et infrarouge permettent de valider l'approche basée sur le réseau de reconstruction pour ces deux modalités. Ils sont cependant difficilement exploitables et comparables. En effet, ils sont basés sur une séparation aléatoire des bases de données, et ces tests gagneraient à être répétés de nombreuses fois avec des séparations différentes des bases pour être plus généraux. De plus, la création du modèle (i.e. la phase d'enrôlement) repose sur un moyennage des projetés de toutes les images (sauf celle testée), ce qui est incompatible avec de nombreux protocoles de bases de données où la contrainte d'une seule image pour l'enrôlement est imposée. C'est notamment le cas avec le protocole dédié de la base de données Notre–Dame (voir introduction de ce chapitre).

5.1.3 Résultats sur la base de données Notre–Dame

Les tests sur la base Notre–Dame ont été réalisés en trois temps, ce qui conduit aux trois approches détaillées ci–après.

Première expérimentation Dans un premier temps, nous avons utilisé les ensembles détaillés dans l'introduction de ce chapitre pour les deux modalités. Le premier problème avec **TrS** est qu'il n'y a qu'une seule image par personne, nous avons donc créé de nouvelles images en appliquant des transformations aux images originales, telles des rotations, rehaussements de contraste, ou des ajouts de luminosité artificielle à certaines parties de l'image. Nous avons ainsi obtenu $159 \times 12 = 1908$ images que nous avons divisées en deux parties distinctes : la première partie composée de 159 images (une image par personne choisie aléatoirement) pour réaliser une validation croisée lors de l'apprentissage, et le reste pour l'apprentissage. Pour chaque personne, l'image de référence (i.e. l'image à reconstruire) est l'image originale. La validation croisée est réalisée après chaque itération lors de l'apprentissage pour éviter un surapprentissage, ce qui améliore ainsi la capacité de généralisation du réseau.

La moyenne et l'écart–type des différents jeux de tests pour les deux modalités des deux expériences (*Same–session* et *Time–lapse*) sont présentés aux figures 5.5,

5.6, 5.7, 5.8. Il s'agit de courbes ROC (Receiver Operating Characteristic) où l'ordonnée représente le taux de reconnaissance (*true positive rate*) et l'abscisse le taux de faux–acceptés (*false positive rate*). Nous pouvons voir que les résultats pour les deux modalités sont satisfaisants pour l'expérience *Same–session* (Fig.5.5 et 5.7), mais plutôt mauvais en ce qui concerne l'expérience *Time–lapse* (Fig.5.6 et 5.8) où le taux de reconnaissance au rang 0 est d'environ 30%.

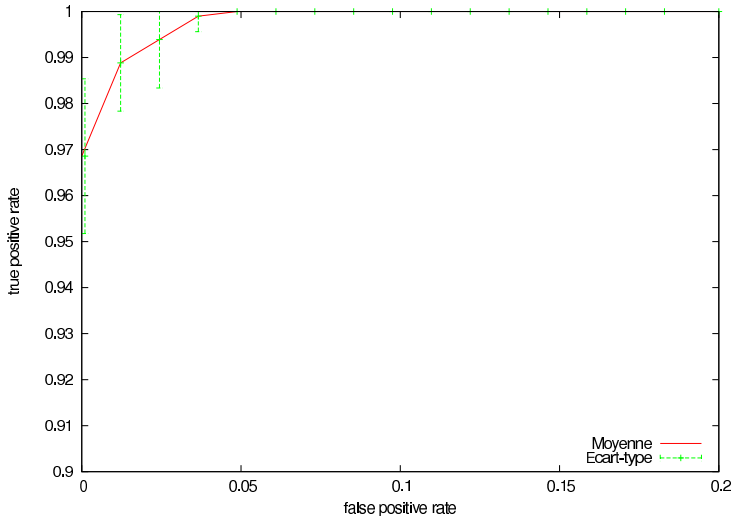


FIGURE 5.5 – Courbe ROC pour l'expérience *Same–session*, Visible, première expérimentation.

La principale raison des mauvais scores pour l'expérience *Time–lapse* est qu'il n'y a qu'une seule image disponible par personne dans **TrS**. En appliquant certaines transformations (rotations, rehaussement de contraste ...) aux images présentées en entrée lors de l'apprentissage, le réseau est capable de les apprendre. Cependant, d'autres variations (comme les expressions faciales) ne sont pas prises en compte (il n'y a pas d'expressions faciales dans **TrS**), le réseau ne peut donc pas apprendre à y être invariant, et comme il y a des expressions faciales dans les galeries et les probes, la reconnaissance échoue.

Deuxième expérimentation Dans cette deuxième expérimentation, nous avons essayé de contourner le problème de la première expérimentation, à savoir le manque d'images de visages présentant des expressions faciales dans l'ensemble d'apprentissage. Nous avons appliqué les mêmes transformations aux 159 images de **TrS** que lors la première expérimentation, et avons ajouté un sous–ensemble de la base FERET composé de 2708 images de visages de 994 personnes. Ce sous–ensemble contient des variations de pose des visages, des variations d'éclairage ainsi que des

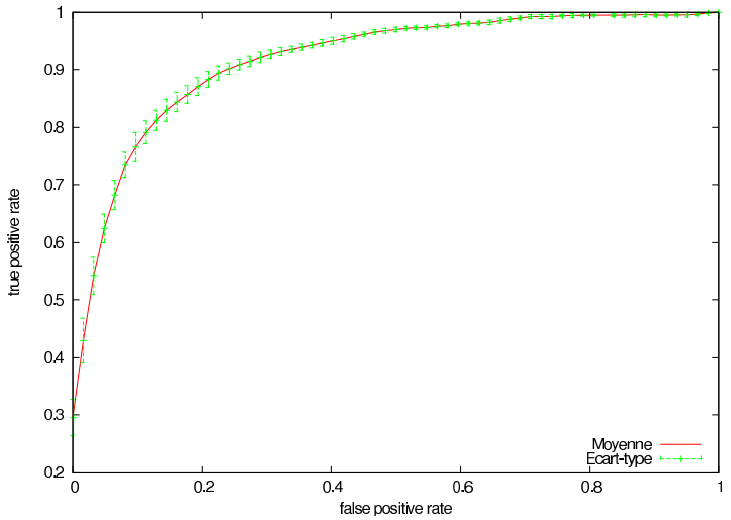


FIGURE 5.6 – Courbe ROC pour l'expérience *Time–lapse*, Visible, première expérimentation.

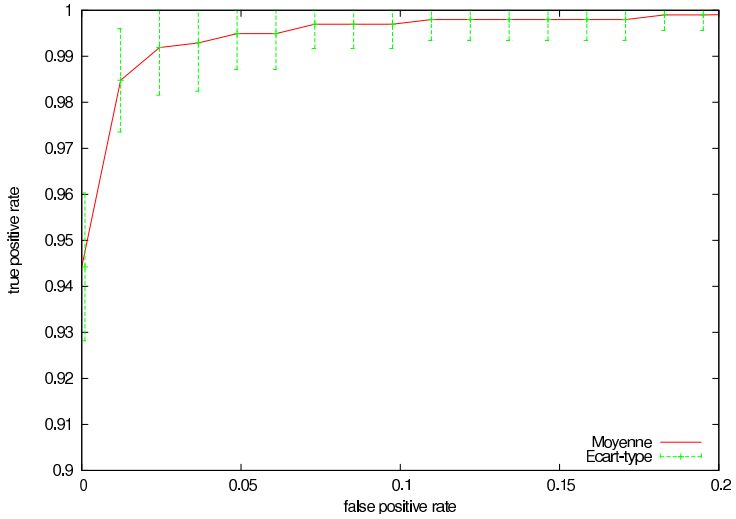


FIGURE 5.7 – Courbe ROC pour l'expérience *Same–session*, IR, première expérimentation.

expressions faciales. La base d'apprentissage est finalement composée de 4608 images de 1153 personnes. Nous en avons extrait 355 images de personnes différentes pour former l'ensemble nécessaire à la validation croisée (comme pour la première expérimentation, voir le paragraphe 5.1.3).

Les résultats obtenus pour l'expérience *Time–lapse* pour la modalité visible sont présentés à la figure 5.9. Les résultats pour l'expérience *Same–session* sont sensi-

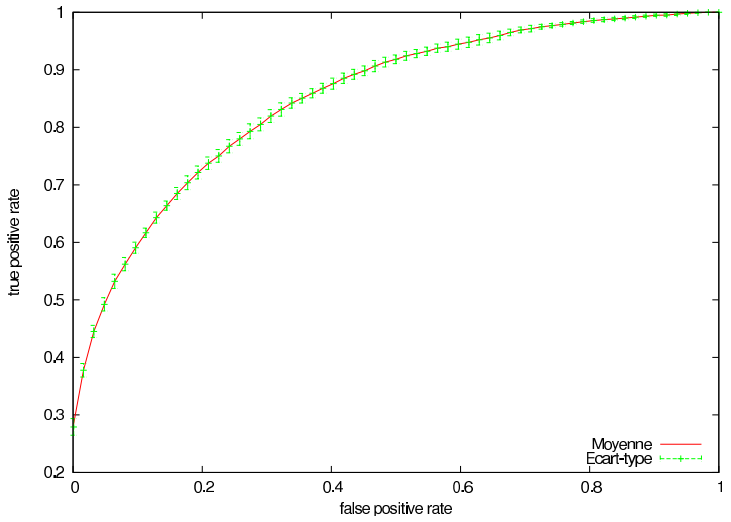


FIGURE 5.8 – Courbe ROC pour l'expérience *Time–lapse*, IR, première expérimentation.

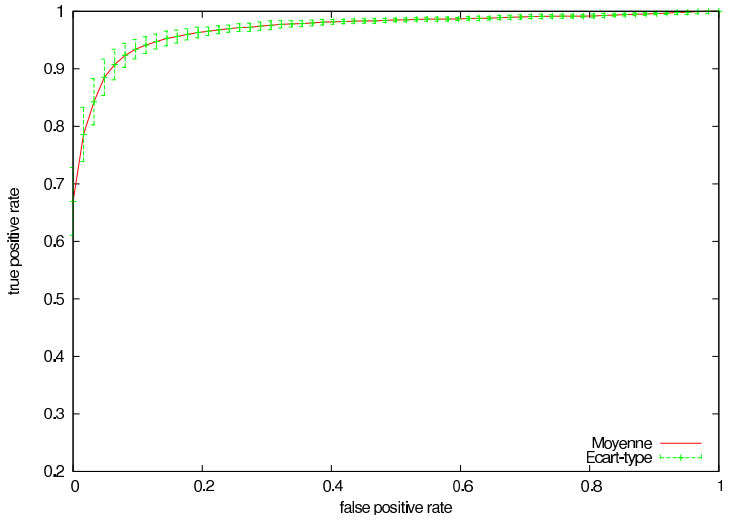


FIGURE 5.9 – Courbe ROC pour l'expérience *Time–lapse*, Visible, deuxième expérimentation

blement identiques que lors de la première expérimentation, ils ne sont donc pas présentés ici.

Comparés à ceux de la première expérimentation, les résultats sont meilleurs (le taux de reconnaissance au rang 0 est compris entre 60% et 76%, il était d'environ 30% pour la première expérimentation), ce qui confirme le manque de variations (et notamment d'expressions) du précédent ensemble d'apprentissage. Le principal pro-

blème avec cette expérimentation est qu'il nous est impossible de réaliser la même chose avec la modalité infrarouge étant donné le manque d'images disponibles pour celle–ci.

Troisième expérimentation En vue d'augmenter le nombre d'images de la base d'apprentissage, et donc le nombre de variations que le réseau peut apprendre, nous avons décidé d'utiliser des personnes de l'ensemble de test (**TeS**) pour la phase d'apprentissage. Nous avons divisé cet ensemble aléatoirement en deux parties disjointes de 41 personnes chacune pour former les ensembles SET_1 et SET_2 composés respectivement de $N1$ et $N2$ images chacun. Les mêmes variations que pour les deux expérimentations précédentes ont été appliquées à **TrS**, et SET_1 a été ajouté à cet ensemble d'apprentissage. Une image par personne a été retirée de cet ensemble pour former l'ensemble de validation, nous avons finalement une base d'apprentissage composée de $159 \times 11 + N1 = 2964$ images de $159 + 41 = 200$ personnes, et une base de validation composée de 200 images (de 200 personnes différentes).

Les listes d'images définissant les probes ont été changées, en effet nous ne voulions pas tester le réseau avec des personnes qui avaient été utilisées lors de l'apprentissage. Ainsi, les 41 personnes de SET_1 ont été enlevés des probes, mais pas des galeries. Les tests consistent ainsi à tester les 41 personnes (de SET_2) parmi les 82 de la base (SET_1+SET_2).

Le tableau 5.3 montre que les résultats obtenus pour l'expérience *Same–session* sont bons, la modalité visible ayant de meilleurs résultats que la modalité infrarouge. Cependant les résultats pour l'expérience *Time–lapse* sont moins bons (Tableau 5.4) que ceux obtenus par Chen *et al.* [63]. Le protocole utilisé ici est de plus plus simple étant donné que le nombre d'images utilisées pour la galerie est inférieur à celui dans [63]. La principale explication est que notre approche fonctionne à basse résolution (les images sont de taille 56×46), tandis que Chen *et al.* utilisent une ACP avec des résolutions d'images plus grandes, ils sont donc capables d'extraire des informations plus pertinentes et précises (les vecteurs propres de l'ACP), et les classes sont finalement mieux séparables.

5.1.4 Importance de l'enrôlement

Les taux relativement mauvais obtenus plus haut pour l'expérience *Time–lapse* sont dus aux galeries. Dans notre approche, la variance intra classe peut être supérieure à la variance inter classe. Dans un scénario où une seule image est utilisée pour l'enrôlement (comme dans les expériences ci–dessus), si l'image utilisée pour l'enrôlement n'est pas bien choisie, les classes peuvent ne pas être clairement séparables, et des faux positifs peuvent apparaître.

Pour montrer cela, nous avons mené des expériences où une image par personne est utilisée pour l'enrôlement et le reste des images pour tester.

Gallerie \ Probe	FALF	FALM	FBLF	FBLM
FALF		1.00 0.90	0.97 0.87	1.00 0.87
FALM	1.00 0.95		0.97 0.87	0.97 0.87
FBLF	0.95 0.97	0.95 0.87		1.00 0.97
FBLM	1.00 0.95	1.00 0.85	1.00 0.92	

TABLE 5.3 – Taux de reconnaissance au rang 0 pour l'expérience *Same–session*, troisième expérimentation. Haut : Visible ; Bas : IR.

Gallerie \ Probe	FALF	FALM	FBLF	FBLM
FALF	0.80 (0.86) 0.41 (0.62)	0.76 (0.85) 0.44 (0.61)	0.68 (0.66) 0.37 (0.55)	0.67 (0.64) 0.38 (0.52)
FALM	0.73 (0.88) 0.42 (0.56)	0.75 (0.59) 0.38 (0.58)	0.68 (0.66) 0.34 (0.51)	0.65 (0.68) 0.38 (0.51)
FBLF	0.72 (0.76) 0.44 (0.55)	0.71 (0.74) 0.37 (0.55)	0.77 (0.79) 0.46 (0.56)	0.78 (0.79) 0.42 (0.58)
FBLM	0.73 (0.76) 0.43 (0.53)	0.71 (0.76) 0.34 (0.53)	0.73 (0.82) 0.41 (0.57)	0.73 (0.82) 0.42 (0.58)

TABLE 5.4 – Taux de reconnaissance au rang 0 pour l'expérience *Time–lapse*, troisième expérimentation. Haut : Visible ; Bas : IR. Entre parenthèses : les résultats obtenus par Chen *et al.* [63].

Le vecteur de poids du réseau de la troisième expérimentation (voir section 5.1.3) a été réutilisé pour calculer les vecteurs projetés des images. Puis une image par personne de SET_2 a été choisie aléatoirement pour former la galerie, le reste formant la probe. Étant donné le caractère aléatoire de la galerie, le processus a été itéré 1000 fois et la moyenne du taux de reconnaissance a été calculée. Le résultat final est présenté à la figure 5.10.

Nous pouvons voir que le taux de reconnaissance moyen au rang 0 pour la modalité visible est d'environ $84.\%$. Cela surpasse les 16 tests de l'expérience *Time–lapse* réalisés lors de l'expérimentation 3 (voir le tableau 5.4). Pour la modalité infrarouge, le taux de reconnaissance moyen au rang 0 est d'environ 41.9%. Cela correspond à environ la moyenne des taux de reconnaissance obtenus lors des 16 tests *Time–lapse* de l'expérimentation 3 (voir le tableau 5.4). De ceci, nous pouvons tirer deux choses : premièrement, la modalité visible surpasse l'infrarouge dans tous

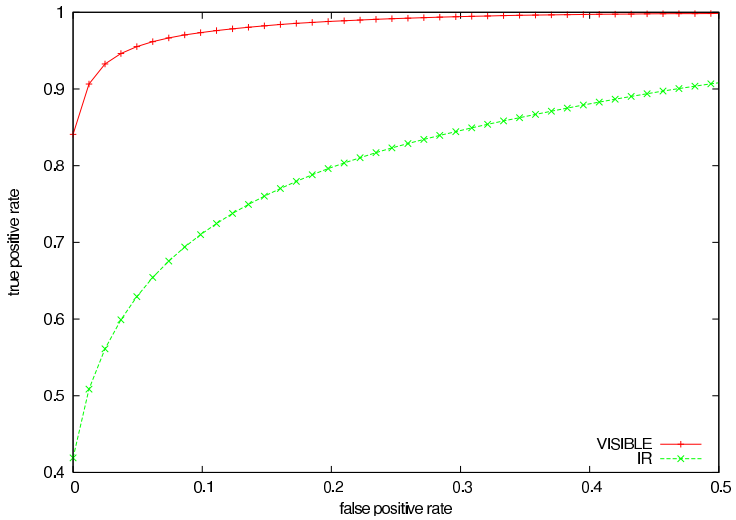


FIGURE 5.10 – Courbe ROC moyenne avec les images d'enrôlement prises aléatoirement.

les cas, et deuxièmement les galeries des expériences *Time–lapse* séparent moins bien les classes que d'autres galeries. Le problème est donc de trouver quelle image est la meilleure pour l'enrôlement d'une personne. Comme nous ne pouvons avoir d'*a priori* sur cette question, une possibilité de contourner le problème est d'enrôler avec plus d'une image.

Nous avons ainsi mené une expérience similaire à celle décrite plus haut, mais cette fois–ci en permettant l'enrôlement avec plusieurs images. Le vecteur caractéristique d'une personne étant alors simplement la moyenne des projetés de ses images d'enrôlement. Le processus a été itéré 1000 fois et la moyenne du taux de reconnaissance est calculée. Pour certaines personnes ne disposant pas d'assez d'images, le maximum d'images disponibles a été pris en compte. Plus formellement :

$$n_{iep} = \begin{cases} \min(\lambda, n_{ip}) - 1 & \text{pour une personne testée} \\ \min(\lambda, n_{ip}) & \text{pour les autres} \end{cases} \tag{5.1}$$

où n_{iep} est le **n**ombre d'**i**mages utilisées pour **e**nrôler une personne **p**, λ est le nombre d'images désiré pour l'enrôlement et n_{ip} le nombre d'*i*mages disponibles pour la personne p. Le terme -1 dans le premier cas apparaît car nous ne voulons pas tester une image ayant été utilisée pour l'enrôlement.

Comme nous pouvons le voir dans le tableau 5.5, le taux de reconnaissance au rang 0 augmente avec le nombre d'images utilisées pour l'enrôlement. Le cas extrême où toutes les images disponibles (sauf celle testée) d'une personne sont utilisées donne un taux de reconnaissance de respectivement 98.4% et 76.4% pour

Modalité	2 images	3 images	4 images	10 images
Visible	91.9	94.5	95.7	97.6
IR	55.4	61.6	65.4	72.9

TABLE 5.5 – Taux de reconnaissance au rang 0 selon le nombre d'images utilisées pour l'enrôlement

les modalités visible et infrarouge. Cependant, ce cas extrême n'est pas réaliste puisqu'il ne prend pas en compte les dates des clichés (l'image testée doit être plus récente que l'image ayant été utilisée pour l'enrôlement).

L'explication de ces résultats est qu'en moyennant les projections de différentes vues d'une même personne, le vecteur signature d'une personne est plus stable aux variations (expressions faciales, luminosité, poses) et les classes deviennent plus facilement séparables. De plus, pour une utilisation opérationnelle, l'utilisation de plusieurs images pour l'enrôlement n'est pas irréaliste, et une mise à jour des vecteurs signatures peut être réalisée facilement au cours du temps.

5.2 Résultats avec un préapprentissage du Réseau de Neurone Convolutionels

Afin de tester l'utilité de préapprentissages non supervisés de couches de convolution d'un réseau de neurones convolutionnels, nous nous sommes appuyés sur l'algorithme PSD proposé dans [155] (Section 3.7). Les tests ont été réalisés sur la base AT&T. Brièvement, l'algorithme est composé d'un encodeur et d'un décodeur, l'encodeur se chargeant de coder de façon non linéaire un patch en un vecteur parcimonieux via une banque de filtres, le décodeur étant utilisé pour reconstruire le patch original à partir du vecteur parcimonieux obtenu (voir Figure 3.13 au chapitre sur les Réseaux de Neurones Convolutionels). La fonctionnelle à minimiser se résume à :

$$L(Y, Z; B, P_f) = \|Y - BZ\|_2^2 + \lambda\|Z\|_1 + \alpha\|Z - F(Y; G, W, D)\|_2^2 \qquad (5.2)$$

où $Y \in \mathbb{R}^n$ est un échantillon d'apprentissage (i.e. une image réarrangée en vecteur), $Z \in \mathbb{R}^m$ le vecteur parcimonieux, $B \in \mathbb{R}^{n \times m}$ la matrice contenant les filtres de reconstruction, et $F(Y; G, W, D)$ est définie par :

$$F(Y; G, W, D) = G \tanh(WY + D)$$

où $W \in \mathbb{R}^{m \times n}$ est la matrice qui encode les filtres de l'encodeur, $D \in \mathbb{R}^m$ est un vecteur de biais, et $G \in \mathbb{R}^{m \times m}$ est une matrice de gains permettant de mettre à l'échelle la sortie de la fonction F. Les paramètres λ et α permettent de donner plus

ou moins d'importance aux différents termes de l'équation. Ces paramètres ont été fixés à 1 lors des tests.

Étant donné le caractère parcimonieux des filtres appris, le nombre de neurones par couches dans le réseau de reconstruction est considérablement augmenté par rapport à l'architecture présentée à la figure 5.1. Les noyaux de convolution de la couche C_1 du réseau de reconstruction étant de taille 7×7, nous avons fixé le nombre de filtres de l'encodeur à 100, ce qui est plus de deux fois la dimension des noyaux, la couche de convolution C_1 du réseau possède donc maintenant 100 neurones.

Afin de réaliser l'apprentissage des filtres correspondant à la première couche de convolution, nous avons extrait 20000 patches de taille 7×7 de la base AT&T avec un écart type suffisant (afin d'éviter les patches trop uniformes).

Après apprentissage, les filtres obtenus pour l'encodeur ressemblent à des détecteurs de contours localisés et orientés (Figure 5.11).

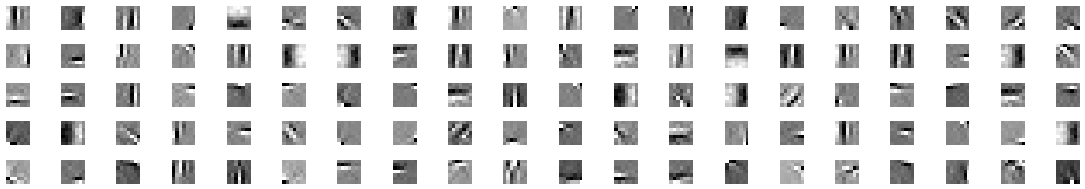


FIGURE 5.11 – Les 100 filtres de l'encodeur obtenus via l'algorithme PSD correspondants à la première couche de convolution.

Une deuxième banque de filtres de taille 6×6 a ensuite été produite via l'algorithme PSD où les patches utilisés pour l'apprentissage ont été obtenus à partir des sorties de la première banque de filtres. Concrètement, de nombreux patches de taille 18×18 ont été extraits des images de la base de données, ces patches ont ensuite été convolués par les noyaux de taille 7×7 obtenus précédemment. Les sorties sont donc des patches de taille 12×12. Ces sorties ont ensuite subi un sous–échantillonnage de taille 2×2 (similaire à la couche S_2 du réseau de neurones convolutionnels) pour ainsi être de taille 6×6. 40000 patches ont ainsi été obtenus, et utilisés comme entrées de l'algorithme PSD. Pour ce second niveau, le nombre de filtres de l'encodeur a été fixé à 150, ce qui est plus de quatre fois la dimension des entrées. Les filtres obtenus sont présentés à la figure 5.12. Ces filtres présentent des aspects locaux moins nets que les filtres obtenus au premier niveau, certaines structures sont cependant apparentes.

Dans la suite, ces banques de filtres vont être utilisées pour initialiser les couches de convolutions C_1 et C_3 du réseau de reconstruction. L'architecture, dont le nombre de neurones par couche a grandement augmenté, est :

– $Input$. Nombre d'images : 1. Taille : 56×46.

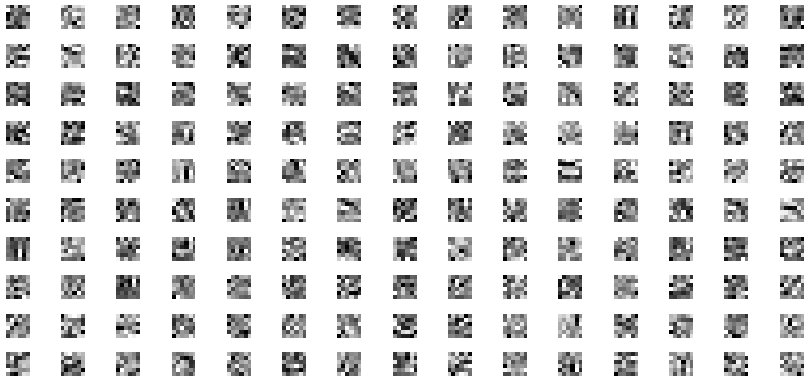


FIGURE 5.12 – Les 150 filtres de l'encodeur obtenus via l'algorithme PSD pour la deuxième couche de convolution.

- C_1. Nombre de cartes : 100 ; Taille des noyaux : 7×7 ; Taille des cartes : 50×40. Toutes les cartes sont connectées à l'entrée.
- S_2. Nombre de cartes : 100 ; Taille des noyaux : 2×2 ; Taille des cartes : 25×20. Connexions $1 - 1$.
- C_3. Nombre de cartes : 150 ; Taille des noyaux : 6×6 ; Taille des cartes : 20×15. Ratio de connexion$= 0.5$.
- S_4. Nombre de cartes : 150 ; Taille des noyaux : 4×3 ; Taille des cartes : 5×5. Connexions $1 - 1$.
- C_5. Nombre de cartes : 200 ; Taille des noyaux : 5×5 ; Taille des cartes : 1×1. Couche complètement connectée à S_4.
- F_6. Nombre de cartes : 100 ; Couche complètement connectée à C_5.
- F_7. Nombre de cartes : 2576 ; Couche complètement connectée à F_6.

Le protocole de test utilisé sur la base AT&T est similaire à celui mis en œuvre lors des résultats préliminaires (Section 5.1.1).

La base de données est divisée en deux parties disjointes, la première contenant 35 personnes (soit 350 images) est utilisée pour l'apprentissage du réseau, la seconde composée des 5 personnes restantes (50 images) est utilisée pour les tests. Le protocole est lui aussi similaire et peut être décomposé en 4 parties :

- Pour une image test I_s de la personne s, la projection P_{I_s} (vecteur extrait de la couche F_6) est calculée,
- pour chaque personne de la base de données, un modèle est créé. Il s'agit du vecteur moyen des projetés des images de cette personne (excepté I_s),
- les distances de P_{I_s} à tous les modèles sont calculées,
- le rang correspondant à I_s est calculé.

Dans ce test, nous avons utilisé la distance l^2 pour calculer toutes les distances. Les phases d'apprentissage et de test ont été réalisé 20 fois avec à chaque fois

une séparation différente de la base de données. Le taux de reconnaissance moyen a ensuite été calculé. Ceci permet de donner aux résultats plus de fiabilité.

Nous avons réalisé 6 expériences différentes afin d'évaluer l'utilité de préapprentissages parcimonieux. Dans la suite, nous notons RR l'expérience où les deux couches de convolution C_1 et C_3 sont initialisées de manière classique (au hasard). UR indique l'expérience où la première couche de convolution C_1 est initialisée avec la première banque de 100 filtres et la seconde couche de convolution est initialisée de manière aléatoire. UU indique l'expérience où les deux couches de convolution C_1 et C_3 sont initialisées avec les banques de 100 et 150 filtres. Enfin chacune de ces expériences est doublée selon que est autorisé ou non un **raffinement** des filtres des couches de convolution lors de l'apprentissage supervisé final.

Les résultats de cette expérience sont présentés au tableau 5.6.

Apprentissage \ Expérience	RR	UR	UU
Non raffiné	84%	88.7%	82%
Écart type	4.67	3.10	3.52
Raffiné	87.5%	90.2%	88.2%
Écart type	3.90	2.54	2.91

TABLE 5.6 – Résultats selon l'initialisation des deux couches de convolution C_1 et C_3. R : couche initialisée aléatoirement, U : couche initialisée avec une banque de filtres.

De cette expérience et des résultats obtenus, nous pouvons dessiner trois conclusions :

– Le résultat le plus surprenant est le bon taux de reconnaissance (84%) obtenu dans le cas où les deux couches de convolution sont initialisées aléatoirement et dont les poids *restent fixes* durant l'apprentissage supervisé final (cas RR non raffiné). Nous pensons que cela est dû au grand nombre de filtres capables de capter suffisamment d'information (même désordonnée) pour une classification. Ce type de résultat a déjà été constaté dans [151]. Une récente étude [253] montre que certains résultats de la littérature à l'aide de réseaux profonds peuvent en grande partie être attribués au choix de l'architecture. Notons cependant que l'écart type pour cette expérience est (évidemment) le plus élevé. Notons également que le cas RR avec un raffinement des poids correspond à l'apprentissage classique des réseaux de convolution.
– Les filtres ayant été raffinés (quelque soit leur initialisation) offrent toujours de meilleurs résultats que lorsqu'ils sont contraints à rester fixes. Notons également que l'écart type des taux de reconnaissance diminue dans les trois cas lorsque les filtres sont raffinés.

– L'initialisation de la deuxième couche de convolution C_3 avec les filtres appris de manière non supervisée (UU) offre de moins bons résultats que l'expérience où seule la première couche est initialisée de cette façon (UR).

L'utilisation de méthodes de préapprentissage pour les couches de convolution des réseaux de neurones convolutionels est une approche relativement récente. La mise en œuvre de telles méthodes n'est pas aisée, mais permet l'amélioration des taux de reconnaissance. La principale difficulté est, à nos yeux, la création des tables de connexion reliant les différentes couches entre elles (spécialement pour la table reliant S_2 à C_3).

L'utilisation de méthodes parcimonieuses pour le préapprentissage nous a conduit dans un troisième temps à l'exploration des méthodes purement parcimonieuses (en dehors du cadre des réseaux de neurones convolutionels) pour la reconnaissance faciale.

5.3 Résultats avec des méthodes parcimonieuses

Dans cette section, nous décrivons les résultats obtenus avec une méthode de décomposition parcimonieuse de visages sur des dictionnaires préalablement appris. Une décomposition parcimonieuse d'un signal (ou d'une image) consiste à décomposer ce signal sur un dictionnaire de sorte que le vecteur de coefficients obtenus comporte beaucoup de zéros. La classification est ensuite effectuée via l'algorithme de classification basée sur la parcimonie (Section 4.5). Les tests ont été effectués sur la base Notre–Dame en suivant le protocole dédié à cette base de données. Les images ont été recadrées par rapport à la position des yeux et redimensionnées à la taille 90×110.

5.3.1 Apprentissage des dictionnaires

Dans le but d'apprendre les dictionnaires utilisés ensuite pour décomposer les patches extraits des images, nous avons extrait 10000 patches de taille 10×10 des images pour la modalité visible du *Train–Set* comprenant suffisamment de variance (afin d'éviter d'avoir des patches trop uniformes). La méthode de décomposition parcimonieuse utilisée est l'algorithme OMP (Section 4.3.2). L'apprentissage du dictionnaire a été effectué via l'algorithme K–SVD (Section 4.4.2). La redondance du dictionnaire a été fixée à 2, ce qui implique que le nombre d'atomes est de 200 ($2 \times 10 \times 10 = 200$). Le nombre maximal d'atomes pour l'algorithme OMP a été fixé à 5, ce qui implique que chaque patch de l'ensemble d'apprentissage est décomposé en une combinaison linéaire de 5 atomes, les coefficients des autres atomes étant 0. Ce sont également 5 atomes par patch d'apprentissage qui vont ainsi être mis à jour lors de l'apprentissage. Le processus itératif de l'apprentissage a été stoppé après

100 itérations. Les atomes obtenus sont présentés à la figure 5.13 et représentent donc le dictionnaire obtenu pour la modalité visible. Les hyperparamètres utilisés ont été fixés expérimentalement. Tandis que certains atomes encodent des motifs à basse fréquence, d'autres sont plus orientés et sélectifs.

Un dictionnaire similaire a été calculé pour la modalité infrarouge, à partir d'images infrarouges du *Train–Set*. Ce dictionnaire n'est cependant pas montré ici, étant donné que les atomes sont *très* ressemblants à ceux obtenus pour la modalité visible. Il se pourrait donc que les images infrarouges contiennent en fait la même diversité morphologique que les images visibles, tout du moins à l'échelle considérée (90×110).

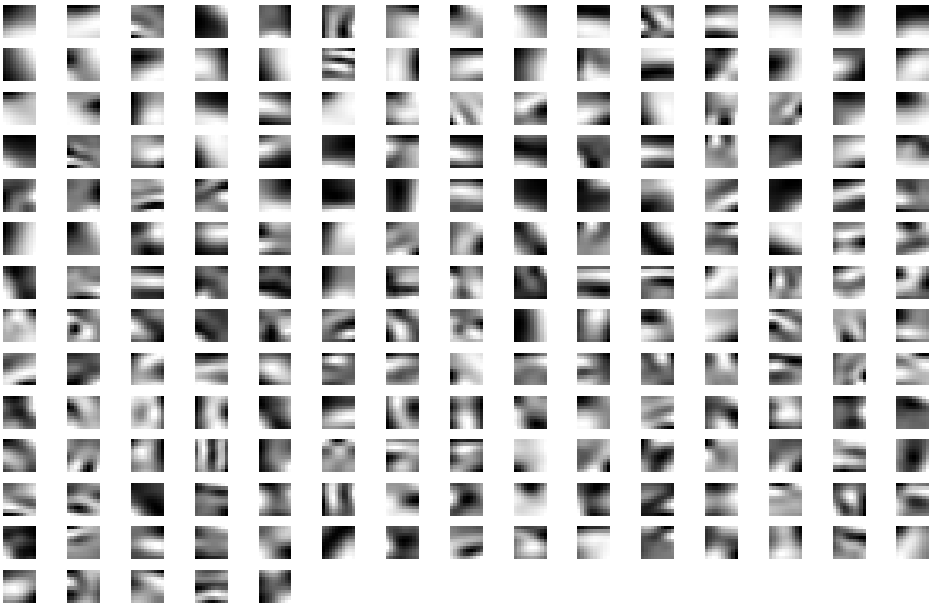


FIGURE 5.13 – Les 200 atomes appris lors de l'apprentissage du dictionnaire.

5.3.2 Création des vecteurs caractéristiques parcimonieux

Une fois les dictionnaires appris, une image de visage est décomposée en patches de taille 10×10 ne se recouvrant pas. Étant donné que les images sont de taille 90×110, chaque image de visage est composée de 99 patches.

Chacun de ces patches est ensuite décomposé sur le dictionnaire (Figure 5.14). Afin d'avoir une approximation rapide du vecteur parcimonieux, nous avons utilisé l'approche d'inversion basée sur un seuillage doux itératif (Section 4.3.1). Le ratio de termes non nuls pour chaque vecteur parcimonieux obtenu est de l'ordre de 2.5% ($\approx 5/200$).

Les 99 vecteurs parcimonieux obtenus sont ensuite concaténés en un seul vecteur représentant le vecteur de caractéristiques de l'image. Ce vecteur est donc de taille $99 \times 200 = 19800$, comprenant approximativement 19300 valeurs nulles, ce qui le rend très parcimonieux.

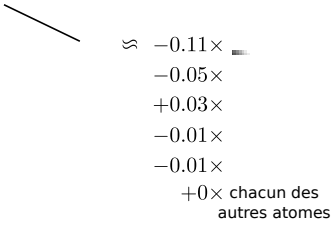


FIGURE 5.14 – Décomposition parcimonieuse de l'image d'un visage sur un dictionnaire.

5.3.3 Résultats de l'identification

L'identification a été réalisée via l'algorithme SRC (Section 4.5). Les résultats pour les expériences *Same–session* et *Time–lapse* sont présentés respectivement aux tableaux 5.7 et 5.8. Ceux–ci pour l'expérience *Same–session*, qui est un test plutôt simple, sont équivalents à ceux présentés avec l'approche neuronale (Section 5.1.3), ou ceux présentés dans [63] basés sur la méthode des eigenfaces. Ils sont cependant significativement meilleurs pour l'expérience *Time–lapse*, notamment pour la modalité visible (Tableau 5.9 comparant les méthodes). La méthode est donc plus robuste aux changements d'illumination (qui affectent les visages de manière globale) qu'aux variations de la distribution de chaleur (qui affectent les visages de manière locale).

5.3.4 Robustesse de la méthode parcimonieuse

Afin de tester la robustesse de l'approche, nous avons appliqué deux types de dégradations aux images de test. Seules les images test ont été dégradées, et non les images utilisées pour l'enrôlement. Pour les deux types de corruption, nous avons utilisé le même protocole que plus haut.

Images bruitées Dans cette expérience, nous avons corrompu les images en y ajoutant un bruit blanc gaussien. L'écart–type de la distribution gaussienne est calculé selon le ratio de la distribution de l'image. Les ratios que nous avons considérés sont de 10%, 20%, 30%, 40% et 50%. Un exemple d'images bruitées est présenté à

Galerie \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF		1.00 0.98	1.00 0.97	0.98 1.00
FA|LM	0.98 0.96		1.00 0.95	0.98 0.96
FB|LF	0.97 1.00	0.97 0.92		1.00 0.97
FB|LM	0.98 0.98	0.98 0.97	1.00 0.98	

TABLE 5.7 – Taux de reconnaissance au rang 0 pour l'expérience *Same–session*. Haut : Visible, bas : IR.

Galerie \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF	0.95 0.83	0.92 0.79	0.87 0.76	0.87 0.77
FA|LM	0.95 0.83	0.93 0.81	0.87 0.77	0.85 0.77
FB|LF	0.86 0.77	0.83 0.74	0.93 0.79	0.91 0.80
FB|LM	0.92 0.79	0.87 0.80	0.88 0.78	0.88 0.82

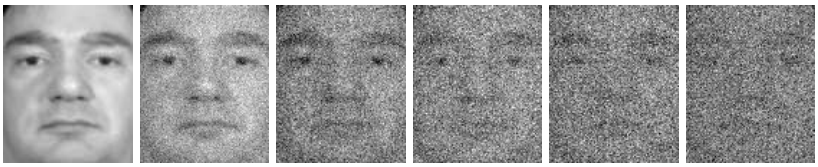
TABLE 5.8 – Taux de reconnaissance au rang 0 pour l'expérience *Time–lapse*. Haut : Visible, bas : IR.

	Same-session			*Time-lapse*		
	[63]	[52]	Méthode proposée	[63]	[52]	Méthode proposée
Visible	97.08 (3.13)	98.41 (1.97)	**98.66** (1.17)	82.66 (7.75)	72.50 (4.01)	**89.31** (3.56)
IR	**97.41** (2.01)	90.5 (4.27)	97.00 (2.08)	77.81 (3.31)	40.06 (3.47)	**78.87** (2.46)

TABLE 5.9 – Comparaison des méthodes pour les deux expériences *Same–session* et *Time–lapse*. Taux de reconnaissance moyens pour les 12 (ou 16) sous–expériences, écart–type entre parenthèses. Meilleur score en gras. [63] recense les résultats avec la méthode fondée sur une ACP, [52] recense les résultats avec l'approche neuronale (également présentés à la section 5.1.3.)

la figure 5.15. Les résultats pour les expériences *Same–session* et *Time–lapse* sont présentés aux figures 5.16 et 5.17. Ces figures montrent les taux moyens de reconnaissance au rang 0 sur les 12 (*Same–session*) ou 16 sous–expériences (*Time–lapse*) en fonction du pourcentage de bruit contenu dans les images. Les écart–types étant de même ordre que ceux obtenus plus haut (Tableau 5.9), ils n'apparaissent pas sur les figures.

Comme nous pouvions nous y attendre, les taux de reconnaissance décroîssent lorsque le taux de bruit contenu dans les images croît. Cette décroissance est cependant quasiment linéaire, et n'est pas significativement différente pour les deux modalités.



(a) Image originale (b) 10% (c) 20% (d) 30% (e) 40% (f) 50%

FIGURE 5.15 – Pourcentage de bruit dans une image de test.

Pixels manquants Dans cette expérience, nous avons corrompu les images de test en leur « enlevant » un certain pourcentage de pixels. La valeur de ces pixels a en fait été mise à 0. Les pourcentages considérés vont de 10% à 90% avec un pas de 10%. Un exemple d'images ainsi corrompues est présenté à la figure 5.18. Les résultats pour les expériences *Same–session* et *Time–lapse* sont présentés aux figures 5.19 et 5.20. Ces figures montrent les taux moyens de reconnaissance au rang 0 sur les 12

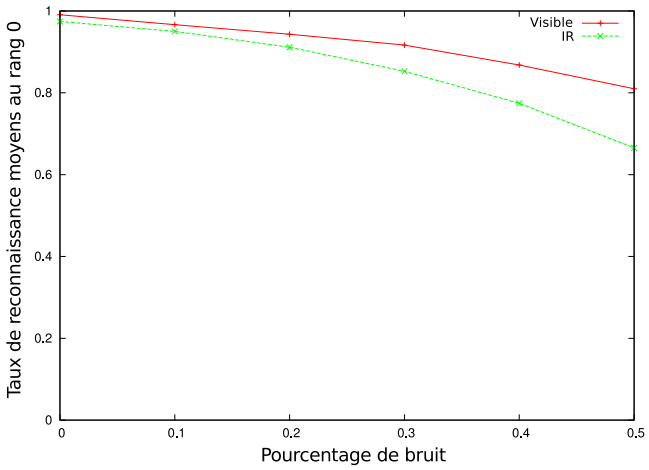


FIGURE 5.16 – Résultats pour l'expérimentation « images bruitées », expérience *Same–session*.

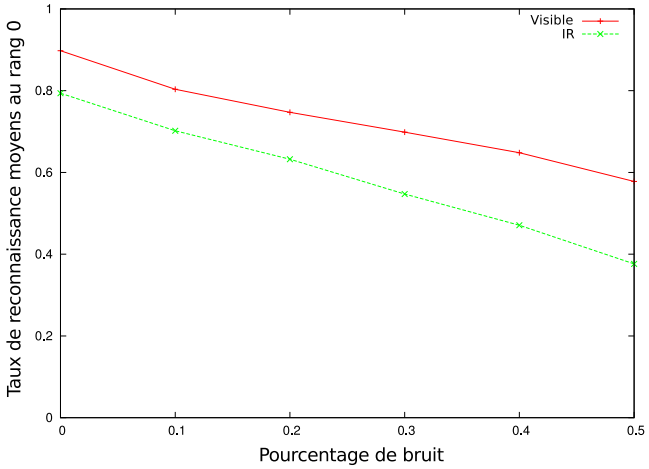
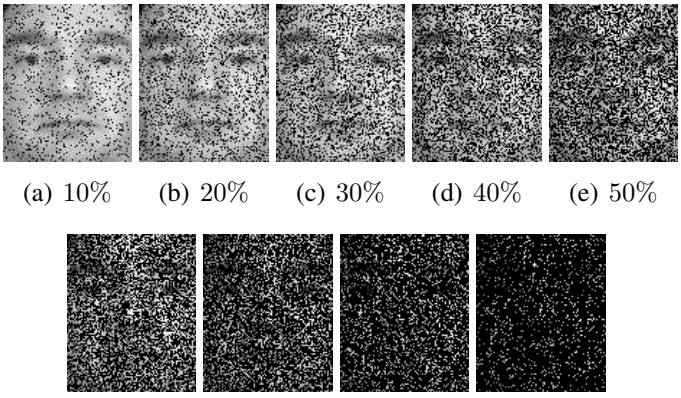


FIGURE 5.17 – Résultats pour l'expérimentation « images bruitées », expérience *Time–lapse*.

(*Same–session*) ou 16 sous–expériences (*Time–lapse*) en fonction du pourcentage de pixels « manquants ».

Nous pouvons voir sur ces figures que la modalité visible résiste bien mieux à ce type de corruption que la modalité infrarouge, pour laquelle les taux de reconnaissance au rang 0 diminuent rapidement.



(a) 10% (b) 20% (c) 30% (d) 40% (e) 50%

(f) 60% (g) 70% (h) 80% (i) 90%

FIGURE 5.18 – Pourcentage de « pixels manquants » dans une image test.

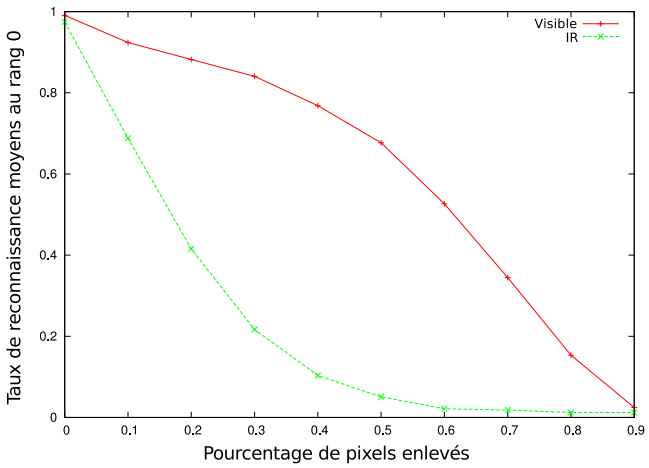


FIGURE 5.19 – Résultats pour l'expérimentation « pixels manquants », expérience *Same–session*.

5.3.5 Variante de la méthode parcimonieuse

Une variante de la méthode basée sur la parcimonie présentée plus haut a été proposée dans [51]. À la différence de la méthode ci–dessus où la classification

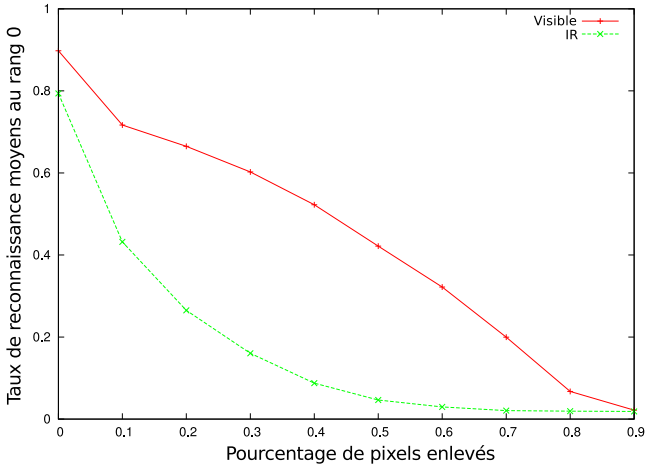


FIGURE 5.20 – Résultats pour l'expérimentation « pixels manquants », expérience *Time–lapse*.

est réalisée sur les vecteurs de caractéristiques parcimonieux entiers (de dimension 19800), la variante considère l'application de l'algorithme SRC sur chacun des vecteurs parcimonieux issus de la décomposition des patches composant l'image.

Cette variante consiste à calculer le vecteur de résidus issus de SRC pour chaque patch, puis à les fusionner. Cette fusion de vecteurs de résidus s'effectue simplement en normalisant chaque vecteur entre 0 et 1 puis en les sommant. Le vecteur de résidu final est ensuite utilisé pour déterminer l'identité de l'image test. Cette variante est schématisée à la figure 5.21.

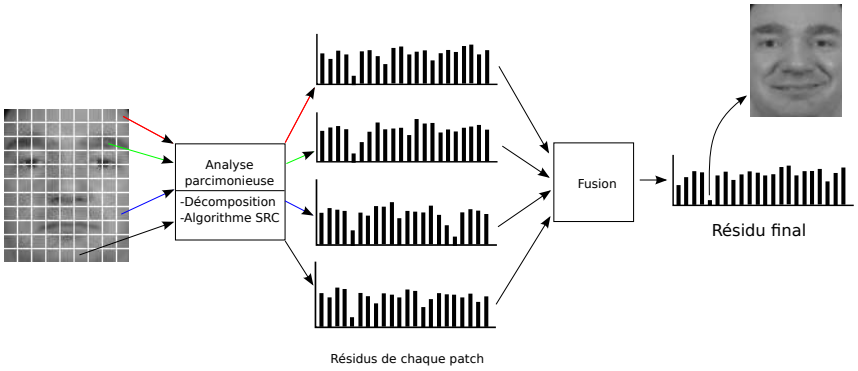


FIGURE 5.21 – Vue schématique de la variante de l'approche parcimonieuse.

Cette variante de la méthode parcimonieuse offre les meilleurs taux de reconnaissance pour l'expérience *Same–session* (Tableau 5.10), même si ceux–ci ne sont pas vraiment significatifs. Les taux de reconnaissance pour l'expérience *Time–lapse* (Tableau 5.11) sont plus contrastés. Ainsi, si les taux de reconnaissance pour la

Galerie \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF		1.00 0.98	1.00 1.00	0.98 0.98
FA|LM	1.00 1.00		1.00 0.95	0.98 0.97
FB|LF	1.00 1.00	0.98 0.93		1.00 0.97
FB|LM	0.98 0.98	0.98 0.98	1.00 1.00	

TABLE 5.10 – Taux de reconnaissance au rang 0 pour l'expérience *Same–session* avec la variante de la méthode parcimonieuse. Haut : Visible, bas : IR.

Galerie \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF	0.96 0.77	0.94 0.76	0.90 0.70	0.89 0.70
FA|LM	0.97 0.77	0.96 0.77	0.87 0.69	0.89 0.72
FB|LF	0.89 0.77	0.88 0.75	0.94 0.79	0.93 0.77
FB|LM	0.93 0.77	0.89 0.75	0.91 0.81	0.92 0.77

TABLE 5.11 – Taux de reconnaissance au rang 0 pour l'expérience *Time–lapse* avec la variante de la méthode parcimonieuse. Haut : Visible, bas : IR.

	Same-session				*Time-lapse*			
	[63]	[52]	(1)	(2)	[63]	[52]	(1)	(2)
Visible	97.08 (3.13)	98.41 (1.97)	98.66 (1.17)	**99.49** (0.60)	82.66 (7.75)	72.50 (4.01)	89.31 (3.56)	**92.12** (1.89)
IR	97.41 (2.01)	90.5 (4.27)	97.00 (2.08)	**98.27** (1.89)	77.81 (3.31)	40.06 (3.47)	**78.87** (2.46)	75.98 (3.37)

TABLE 5.12 – Comparaison des méthodes pour les deux expériences *Same–session* et *Time–lapse*. (1) Méthode parcimonieuse (voir Section 5.3.3), (2) Variante de la méthode parcimonieuse. Taux de reconnaissance moyens pour les 12 (ou 16) sous expériences, écart–type entre parenthèses. Meilleur score en gras.

modalité visible sont améliorés (de l'ordre de 3% par rapport à la méthode parcimonieuse initiale), les taux de reconnaissance pour la modalité infrarouge sont plus faibles (de l'ordre de 3%). La comparaison des taux de reconnaissance de cette variante à d'autres méthodes est résumée au tableau 5.12. Nous pensons que la variante de la méthode favorise la modalité visible étant donné que les variations présentes entre les images d'enrôlement et les images de tests sont plutôt globales pour la modalité visible, et à l'inverse, plutôt locales pour la modalité infrarouge. En effet, les principales différences de variations entre les deux modalités sont la luminosité pour le visible (qui est appliquée de manière globale à l'ensemble du visage) et les changements de chaleur émise par le visage (qui ont un aspect plus local, certaines parties du visage étant modifiées alors que d'autres parties non). De par le caractère plus local de la variante de la méthode parcimonieuse, la modalité visible s'en trouve favorisée, tandis que les résultats pour la modalité infrarouge sont dégradés.

5.4 Conclusion

Dans ce chapitre, nous avons présenté les principaux résultats obtenus via les méthodes mises en œuvre lors de la thèse. Les trois méthodes développées lors de la thèse se sont succédées de manière naturelle. Ainsi, nous nous sommes intéressés dans un premier temps à une approche basée sur des réseaux de neurones convolutionnels. Validée sur les deux modalités, l'extension naturelle a été ensuite d'effectuer des préapprentissages de certaines couches. La manière la plus adéquate à nos yeux pour réaliser de tels préapprentissages passe par l'utilisation de méthodes parcimonieuses. Ce qui nous a finalement conduit à des méthodes ne reposant plus sur les réseaux de neurones convolutionnels, mais à des techniques ne faisant appel qu'à des méthodes parcimonieuses.

Les résultats obtenus sur la base de données Notre–Dame (en suivant le protocole défini avec la base de données) avec le réseau de reconstruction ont montré que les taux de reconnaissance étaient meilleurs pour la modalité visible qu'avec la modalité infrarouge. Ceci semble être le cas pour la plupart des algorithmes qu'il nous ait été donné de voir, et s'est vérifié avec les autres approches développées lors de la thèse.

Un préapprentissage non supervisé des noyaux de convolution ont permis de mettre en lumière l'intérêt de la démarche par rapport aux réseaux entraînés de manière classique (via une descente de gradient). L'utilisation de méthodes parcimonieuses pour ce préapprentissage nécessite cependant de revoir à la hausse le nombre de neurones par couche, ce qui peut rendre une propagation avant plus coûteuse. La mise en œuvre de telles méthodes n'est de plus pas aisée, et le problème

des tables de connexion entre certaines couches reste ouvert (Annexe B).

Finalement, une méthode de décomposition parcimonieuse de patches d'images de visages a été testée, conjointement à une classification basée sur l'algorithme SRC. Cette méthode offre les meilleurs résultats à notre connaissance sur la base de données (à condition de suivre le protocole dédié...). Une variante de cette méthode a permis d'améliorer encore un peu les résultats, notamment pour la modalité visible.

Les temps d'exécution sont un point important pour un système biométrique. Dans le cas d'une identification, ceux–ci dépendent bien évidemment de la taille de la galerie. Dans nos expériences, le réseau de neurones convolutionnels est de ce point de vue très rapide, de l'ordre de 15 projections pas secondes. Étant donnée l'augmentation de neurones lorsqu'un préapprentissage est réalisé, le réseau devient moins rapide, de l'ordre de 5 projections par secondes. Un des avantages des réseaux de neurones cependant est qu'ils sont facilement parallélisables, étant donné que le calcul de la sortie des neurones peut s'effectuer indépendamment. Les temps de calcul en revanche pour l'apprentissage du réseau peuvent être longs, il dépendent bien évidemment de l'architecture du réseau ainsi que du nombre d'images dans l'ensemble d'apprentissage. Dans nos tests, le temps d'apprentissage du réseau est de l'ordre de $4 - 5$ heures.

L'apprentissage du dictionnaire pour la méthode parcimonieuse est de l'ordre d'une demi–heure. Le temps utilisé pour la décomposition parcimonieuse de l'ensemble des patches d'un visage est inférieur à la minute, la classification via l'algorithme SRC permet de classifier un vecteur de caractéristiques en environ 2 secondes. Ces temps d'exécution sont donnés à titre indicatif. La décomposition des patches d'un visage sur le dictionnaire est relativement lente étant donné qu'elle fait appel à un processus itératif pour chacun des patches. L'algorithme SRC est relativement lent dans nos tests, étant donné la grande taille des vecteurs caractéristiques.

Comme montré dans de précédents travaux, les taux de reconnaissance pour l'infrarouge sont inférieurs à ceux obtenus pour le visible. Ces deux modalités peuvent néanmoins offrir des complémentarités, et une fusion de celles–ci devrait permettre d'améliorer les taux de reconnaissance. Le chapitre suivant traite des différentes méthodes de fusion mises en œuvre lors de la thèse, ainsi que des résultats en identification obtenus sur la base de données Notre–Dame.

Chapitre 6

La fusion de modalités

Dans ce chapitre, nous traitons la question de la fusion. Sont détaillés les niveaux de fusion, dont le plus populaire, la fusion au **niveau des scores**. Pour ce dernier, nous rappelons les principales méthodes de normalisation des scores et de combinaisons de ceux–ci.

Nous donnons ensuite les résultats obtenus à différents niveaux de fusion pour les principales approches mises en œuvre lors de la thèse.

6.1 Les types de fusion

La fusion d'éléments biométriques peut se référer à de nombreux scénarios différents (Figure 6.1).

1. **Systèmes multi–algorithmes** : C'est le type de système le plus classique implicitement utilisé par de nombreuses approches. Les caractéristiques sont extraites via différents algorithmes puis fusionnées. La fusion de caractéristiques extraites via un algorithme analysant les textures et un autre la forme d'un caractère biométrique entre dans ce cadre.
2. **Systèmes multi–échantillons** : Un capteur unique peut capturer plusieurs instances du même caractère biométrique dans le but de rendre plus robuste l'extraction des caractéristiques ou d'enrichir le modèle biométrique d'une personne. C'est le cas, par exemple, de plusieurs captures de visage d'une personne sous différents angles. L'utilisation de vidéos entre également dans ce cadre.
3. **Systèmes multi–capteurs** : Plusieurs capteurs permettent de capturer le même caractère biométrique sous différents « angles ». Ainsi la capture d'un visage à l'aide d'une caméra classique et d'une caméra infrarouge entre dans

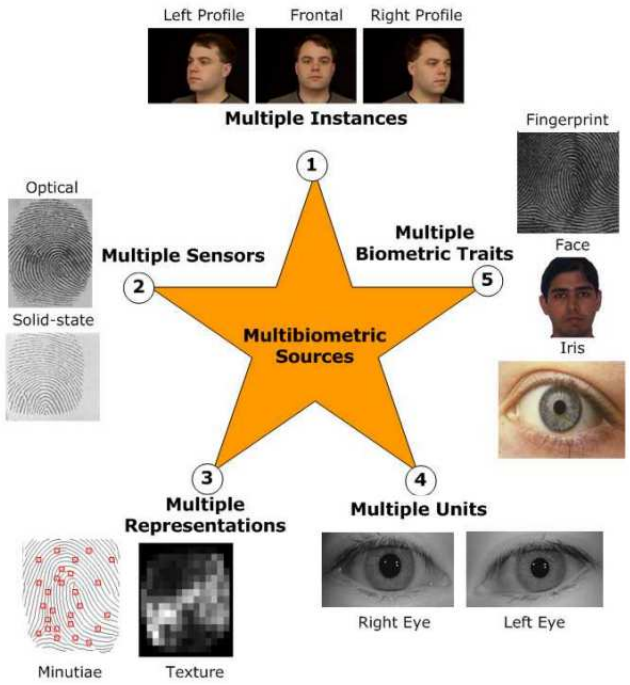


FIGURE 6.1 – Sources de différents types de fusion de traits biométriques [215].

ce scénario. Ce type de système permet notamment la fusion au niveau capteur, ce que ne permettent pas d'autres systèmes comme les systèmes multi–caractères.

4. **Systèmes multi–instances** : Ce type de système permet de capturer plusieurs instances du même caractère biométrique. L'acquisition de plusieurs empreintes digitales via le même capteur est l'exemple typique de ce type de système. Ces systèmes n'entraînent pas de surcoût de capteurs, ni le développement de nouveaux algorithmes. À ne pas confondre avec les systèmes multi–échantillons.
5. **Systèmes multi–caractères** : Ce type de système combine différents traits biométriques d'un individu. Les fusions *visage–iris*, ou *visage–empreinte digitale* font partie de ce type d'approche. Ces systèmes nécessitent différents capteurs ainsi que des algorithmes dédiés à chaque caractère biométrique. Ce type de système a comme principale caractéristique que les caractères biométriques considérés peuvent être plus décorrélés que pour les sytèmes multi–capteurs.

La fusion de données issues de visages capturés via une caméra en lumière visible et une autre en lumière infrarouge entre dans le cadre des systèmes multi–capteurs, où il est considéré que les deux captures sont issues de modalités différentes. Même si les deux captures sont sensiblement décorrélées (la chaleur émise

par un visage n'est pas visible en lumière visible), la fermeture des yeux d'un individu est visible sur les deux modalités.

À noter la présence de *systèmes hybrides* combinant plusieurs scénarios. Une revue de nombreux systèmes biométriques multimodaux développés peut être trouvée dans [244].

6.2 Les différents niveaux de fusion

La reconnaissance d'un individu via une certaine modalité suit une chaîne de traitement, de la capture jusqu'à la décision finale. L'introduction de la multimodalité implique une fusion des différentes modalités, cette fusion pouvant intervenir à différents niveaux de la chaîne de traitement.

Deux familles de fusion peuvent être considérées selon qu'elles interviennent avant ou après l'étape de *matching* (étape qui compare deux empreintes biométriques) [251].

6.2.1 Fusion avant le *matching*

Avant le *matching*, la fusion d'informations peut avoir lieu au **niveau capteur** ou au **niveau caractéristiques**.

Niveau capteur (*Sensor level*)

La fusion de données brutes (*raw data*) peut se faire uniquement si les données capturées proviennent de la même caractéristique biométrique. Les données capturées doivent en effet être compatibles pour être fusionnées (il est par exemple impossible à ce niveau de réaliser une fusion *visage–voix*). La fusion au niveau capteur permet d'obtenir de nouvelles données par fusion des données acquises. Ce sont ces nouvelles données qui vont ensuite être utilisées pour réaliser la reconnaissance. La création d'une image $3D$ à partir d'images $2D$ est un exemple de fusion au niveau capteur. D'autres méthodes utilisent des règles simples comme la somme ou le produit réalisé pixels par pixels.Un autre exemple est la réalisation d'une mosaïque à partir de plusieurs images d'empreintes digitales [147] [208]. D'autres méthodes permettant la fusion au niveau capteur ont été proposées dans la littérature. Dans [33], les auteurs proposent l'utilisation d'algorithmes génétiques pour calculer les poids de la fusion. Dans [261] et [260], les coefficients d'ondelettes des deux images sont fusionnés via de multiples Machines à de Vecteurs de Support (SVM). Dans [240], l'utilisation de la méthode *Particle Swarm Optimization* (PSO) permet de pondérer les coefficients issus de décompositions en ondelettes afin de créér une nouvelle image par transformée en ondelettes inverse (Figure 6.2).

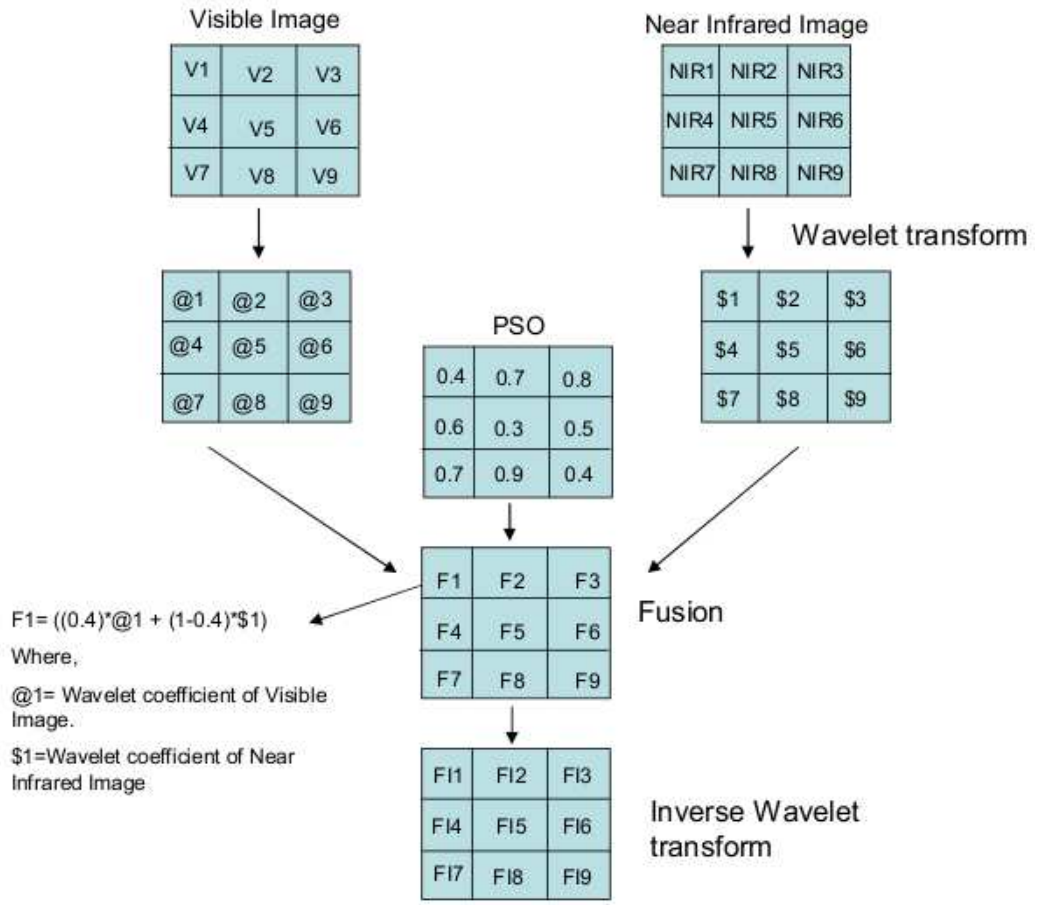


FIGURE 6.2 – Illustration de la fusion pondérée d'images par l'algorithme PSO (tiré de [240]).

Niveau caractéristiques (*Feature level*)

La fusion au niveau caractéristiques consiste à combiner différents vecteurs de caractéristiques (*feature vectors*) issus de différentes modalités ou instances d'une même personne. Une somme pondérée dans le cadre de différentes instances d'un même trait biométrique (caractéristiques **homogènes**) peut ainsi être une façon de calculer un nouveau vecteur de caractéristiques. Dans le cas de caractéristiques **hétérogènes**, une concaténation des vecteurs caractéristiques est souvent le moyen utilisé pour la création du nouveau vecteur. Les vecteurs doivent cependant être compatibles pour une telle fusion, ainsi une concaténation des minuties d'une empreinte digitale (représentée par un graphe) avec des coefficients issus d'une ACP sur des visages n'a pas beaucoup de sens. D'autres méthodes ont été proposées

comme l'utilisation de la méthode *Particle Swarm Optimization* (PSO) dans [238] et [239] pour la sélection des caractéristiques discriminantes. Ce niveau de fusion est difficile à réaliser en pratique étant donné que :

- Les relations entre les différents espaces de caractéristiques doivent être connus à l'avance afin d'éliminer les caractéristiques redondantes. Cela nécessite ainsi l'utilisation d'algorithmes de sélection de caractéristiques.
- La concaténation de deux vecteurs de caractéristiques peut engendrer un vecteur de grande taille et conduire au problème de la malédiction de la dimension [89]. La résolution de ce type de problème peut nécessiter l'utilisation de (trop) nombreux échantillons d'apprentissage, ce qui peut être coûteux dans le cadre biométrique.

6.2.2 Fusion après le *matching*

Les systèmes intégrant les informations de différentes sources pour les combiner après le *matching* peuvent être classés en trois classes principales : ceux réalisant la fusion au **niveau décision**, au **niveau rang** et au **niveau score**.

Niveau décision (*Decision level*)

C'est le niveau de fusion le plus abstrait. Chaque modalité est soumise à sa chaîne de traitement propre et chacun des *matchers* renvoie la décision (accepté/rejeté) pour la modalité associée. Ce sont ces décisions qui sont ensuite fusionnées via des méthodes comme le *majority voting* [174], le *behavior knowledge space* [173], le *weighted voting* [299] ou encore *les règles ET et OU* [126].

Niveau rang (*Rank level*)

Lorsque la sortie de chaque *matcher* biométrique est une liste de résultats triée dans un ordre décroissant de confiance, la fusion peut se faire au niveau *rang*. Différentes méthodes existent pour combiner les rangs assignés par différents *matchers* [136]. Parmi celles–ci, citons la méthode *highest rank method* qui sélectionne le meilleur (minimum) rang de chaque *matcher*, la méthode *Borda count* utilisant la somme des rangs calculés par chaque *matcher* afin d'obtenir les rangs combinés, ou encore la méthode plus générale reposant sur le modèle *Borda count*. Dans cette dernière méthode, les rangs sont dans un premier temps pondérés à l'aide de poids trouvés via une régression logistique, avant d'être dans un second temps additionnés.

Niveau score (*Score level*)

La fusion des scores intervient au niveau des scores produits par chaque *matcher*. Il s'agit de l'approche la plus courante étant donnée sa simplicité d'implémentation et sa plus grande flexibilité. Les données retournées par les *matchers* possèdent en effet une grande richesse d'information (que ce soit une distance à un modèle ou une mesure de dissimilarité). Ce niveau de fusion fait l'objet de la section suivante.

6.3 La fusion au niveau des scores

La sortie d'un *matcher* est classiquement une mesure de similarité entre l'empreinte biométrique testée et l'empreinte biométrique de la galerie dans le cas de l'authentification, ou un vecteur de similarités dans le cas de l'identification.

Dans toute la thèse, nous n'avons considéré que l'identification, les sorties des *matchers* considérés sont donc des vecteurs de similarité correspondant à des distances entre l'empreinte biométrique test et celles contenues dans la base de données.

Afin de s'assurer que ces vecteurs de similarité soient cohérents entre eux, il est nécessaire de les normaliser avant de considérer une fusion des scores.

6.3.1 Normalisation des scores

La normalisation des scores est une étape nécessaire pour répondre aux trois problèmes typiques de la fusion au niveau des scores :

- les scores en sortie de chaque *matcher* peuvent ne pas être homogènes. Par exemple, un premier *matcher* peut retourner en sortie une mesure de distance (dissimilarité) tandis qu'un deuxième peut retourner une mesure de proximité (similarité) ;
- les sorties de chaque *matcher* considéré peuvent ne pas être inclues dans le même intervalle ;
- les scores retournés par différents *matchers* peuvent suivre des distributions statistiques différentes.

La transformation des scores est donc essentielle avant toute combinaison. Il existe différentes techniques de normalisation de scores. Deux caractéristiques doivent être considérées lors du choix du type de normalisation des scores :

- la **robustesse** se référant principalement aux valeurs aberrantes (« *outliers* ») que peut fournir un *matcher*, et
- l'**efficacité** se référant principalement à la proximité de la distribution des scores transformée avec la distribution originale des scores.

Normalisation Min–Max La normalisation **Min–Max** est la technique de normalisation la plus simple, et la plus adaptée lorsque les bornes de la distribution des scores sont connues. Elle consiste simplement à translater les scores minimum et maximum respectivement vers 0 et 1. Lorsque les scores minimum et maximum ne sont pas connus mais qu'ils sont estimables, la technique reste valable mais peut ne pas être robuste à des valeurs supérieures à la borne supérieure estimée ou à des valeurs inférieures à la borne inférieure estimée. La nouvelle valeur d'un score s est obtenue avec cette technique par :

$$s' = \frac{s - \min_i(\mathbf{s_i})}{\max_i(\mathbf{s_i}) - \min_i(\mathbf{s_i})}$$

où $\mathbf{s}$ représente le vecteur de scores. La normalisation **Min–Max** conserve la distribution des scores originale à un facteur d'échelle près.

Normalisation *Decimal scaling* La méthode *Decimal scaling* peut être utilisée lorsque les scores produits par les différents *matchers* évoluent selon une loi logarithmique. Par exemple, si un premier *matcher* produit des scores entre 0 et 1 et un deuxième entre 0 et 100, la normalisation suivante peut être utilisée :

$$s' = \frac{s}{10^n}$$

où $n = log_{10} \max \mathbf{s}$. Les principaux problèmes de cette méthode est qu'elle n'est pas robuste aux valeurs aberrantes, et surtout qu'elle suppose que les scores évoluent d'un facteur logarithmique.

Normalisation Z–Score La technique de normalisation **Z–Score** utilise la moyenne et l'écart–type de la distribution des scores de chaque *matcher*. La moyenne et l'écart–type peuvent être, soit déduits de l'algorithme si l'on a une connaissance *a priori* de celui–ci, soit calculés à partir d'une distribution de scores. La normalisation s'effectue par :

$$s' = \frac{s - \mu}{\sigma}$$

où μ est la moyenne de la distribution calculée ou estimée, et σ l'écart–type. Cette méthode n'est pas robuste aux valeurs aberrantes dans le cas où la moyenne et l'écart-type sont calculés à partir d'une distribution de scores. Cette méthode de normalisation ne conserve pas non plus la distribution originale, à moins que celle–ci ne soit gaussienne. Pour une distribution arbitraire, la moyenne et l'écart–type sont des estimateurs raisonnables mais pas optimaux.

Normalisation MAD Les techniques reposant sur la médiane et l'écart–type absolu (ou MAD pour *Median Absolute Deviation*) sont peu sensibles aux valeurs aberrantes et aux extrémités d'une distribution. La normalisation des scores s'effectue par :

$$s' = \frac{s - \text{median}}{\text{MAD}}$$

où MAD $=$ median$(|s -$ median$(\mathbf{s})|)$. Cependant, lorsque les distributions ne sont pas gaussiennes, la médiane et l'écart–type absolu médian sont des estimateurs faibles. Ainsi, la distribution des scores originale n'est pas conservée lorsque celle–ci est normalisée.

Normalisation QLQ La technique de normalisation QLQ [265] utilise une fonction quadratique–linéaire–quadratique, d'où son nom. Elle s'effectue en deux temps : une normalisation **Min–Max** est d'abord effectuée pour ramener les valeurs entre 0 et 1. La normalisation QLQ prend ensuite comme paramètres le centre c et la largeur w de la zone de recouvrement entre les distributions des imposteurs et des authentiques (Figure 6.3). La zone recouvrant les distributions imposteurs et authentiques

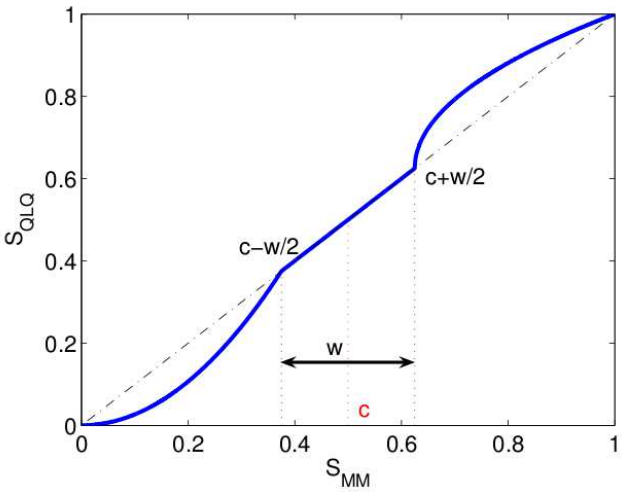


FIGURE 6.3 – Normalisation QLQ.

reste inchangée (fonction linéaire), tandis que les deux autres régions sont transformées à l'aide de fonctions quadratiques par :

$$s' = \begin{cases} \frac{1}{\left(c-\frac{w}{2}\right)} s^2, & \text{si } s \leq \left(c - \frac{w}{2}\right) \\ s, & \text{si } \left(c - \frac{w}{2}\right) < s \leq \left(c + \frac{w}{2}\right) \\ \left(c + \frac{w}{2}\right) + \sqrt{\left(1 - c - \frac{w}{2}\right)\left(s - c - \frac{w}{2}\right)}, & \text{sinon} \end{cases}$$

Normalisation double sigmoïde La technique de normalisation proposée dans [57] utilise une fonction double sigmoïde pour normaliser les scores des différents *matchers* (Figure 6.4). La normalisation des scores s'effectue par :

$$s' = \begin{cases} \frac{1}{1+\exp\left(-2\left(\frac{s-t}{r_1}\right)\right)} & \text{si } s < t \\ \frac{1}{1+\exp\left(-2\left(\frac{s-t}{r_2}\right)\right)} & \text{sinon} \end{cases}$$

où t est un point de référence et r_1 et r_2 sont des paramètres permettant de définir les deux fonctions sigmoïdes. Dans les régions $[t-r_1, t]$ et $[t, t+r_2]$, les deux fonctions sont quasi linéaires. La figure 6.4 montre un exemple de normalisation double sigmoïde transformant des scores compris dans $[0, 300]$ dans le nouvel intervalle $[0, 1]$, avec $t = 200$, $r_1 = 20$ et $r_2 = 30$. Ce type de transformation est généralement util-

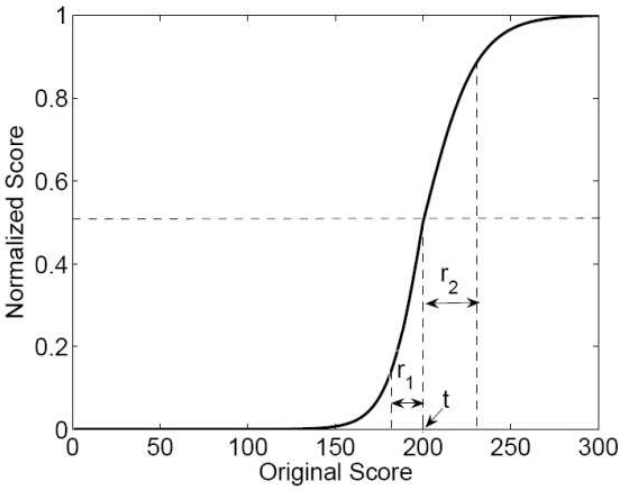


FIGURE 6.4 – Normalisation double sigmoïde.

isé avec le paramètre t choisi dans la zone de recouvrement des imposteurs et des authentiques. Les paramètres r_1 et r_2 sont, eux, choisis comme limites à gauche et à droite des distributions des imposteurs et des authentiques. Ainsi, les scores appartenant à cette zone de recouvrement sont transformés de façon linéaire, tandis que les scores à l'extérieur de cette région sont transformés de façon non–linéaire.

6.3.2 Combinaison des scores

Différentes règles de combinaison de scores formalisées dans [159] ont jeté les bases de la fusion multimodale au niveau scores. Un nouveau score c est produit à partir des scores s_i des M *matchers*.

– **La règle produit** (*Product rule*) : Cette règle définit les nouveaux scores comme étant le produit des scores de chaque *matcher* :

$$c = \prod_{i=1}^{M} s_i$$

Cette règle peut néanmoins être sujette à des valeurs aberrantes si la normalisation utilisée n'est pas robuste. De plus, une normalisation des scores entre 0 et 1 peut poser problème étant donné que les scores de nombreux *matchers* peuvent être rendus nuls si l'un d'eux a été normalisé à 0.

– **La règle somme** (*Sum rule*) : Cette règle définit les nouveaux scores comme étant la somme des scores de chaque *matcher* :

$$c = \sum_{i=1}^{M} s_i$$

Cette règle est généralement plus efficace que la règle produit étant donné qu'elle est plus robuste au bruit ou aux valeurs aberrantes. De plus, un score normalisé à 0 ne va pas pénaliser (ou annihiler l'information des autres scores, comme c'est le cas pour la règle produit).

– **La règle maximum** (*Max rule*) : La règle maximum se contente de définir un nouveau score comme étant le score maximal des scores de chaque *matcher* :

$$c = \max_i \left(s_i\right)$$

– **La règle minimum** (*Min rule*) : La règle minimum se contente de définir un nouveau score comme étant le score minimal des scores de chaque *matcher* :

$$c = \min_i \left(s_i\right)$$

Les deux dernières règles ne sont bien sûr pas du tout robustes aux valeurs aberrantes d'où l'importance du choix de la technique de normalisation des scores.

Les scores issus des différents *matchers* peuvent également être combinés par une somme pondérée. Dans [148], des poids spécifiques à chaque utilisateur sont utilisés pour réaliser une somme pondérée des scores issus de différentes modalités. L'idée de cette technique est que certaines personnes peuvent avoir certains traits biométriques de moins bonne qualité que d'autres personnes. Ainsi, certains ouvriers par exemple peuvent, à force de travaux manuels, présenter des empreintes digitales altérées. Un faible poids pour les empreintes digitales peut, dans ce cas, réduire

les probabilités de faux–rejet. Ce type de méthode requiert cependant un apprentissage spécifique des poids pour chaque utilisateur, et nécessite donc de nombreux échantillons d'apprentissage.

6.4 Résultats de la fusion au niveau capteur

Nous avons conduit une expérience de fusion au niveau capteur en fusionnant les images issues des modalités visible et infrarouge. Étant donné que les images des deux modalités de la base Notre–Dame ont été capturées au même instant, une fusion des deux est possible.

Lors de l'expérience sur ce niveau de fusion, nous avons comparé la variante de la méthode parcimonieuse détaillée au chapitre 5 avec une méthode classique fondée sur une analyse en composantes principales. Les images de chaque modalité ont subi le prétraitement suivant :

- les images ont été coupées afin d'éliminer le fond, puis elles ont été redimensionnées à la taille 90×110. Cette normalisation géométrique a été réalisée selon la distance intra–occulaire, de sorte que les yeux se retrouvent aux mêmes positions dans toutes les images,
- un masque elliptique centré sous les yeux a été appliqué afin d'éliminer certaines parties inutiles de l'image (comme les coins). L'application du masque n'a été réalisée que pour l'approche fondée sur l'ACP,
- les valeurs des pixels ont été normalisées de sorte que la moyenne et l'écart–type de celles–ci soient respectivement de 0 et 1. Notons que seuls les pixels non masqués pour les images où le masque elliptique a été appliqué ont été pris en compte.

La méthode utilisée pour la fusion de deux images se décompose ensuite en deux étapes :

- les valeurs des pixels de l'image de visage infrarouge ont été normalisées entre 0 et 1,
- la nouvelle image I_{fusion} issue de la fusion de l'image visible $I_{visible}$ et de l'image infrarouge I_{ir} a été calculée pixel par pixel selon une loi multiplicative :

$$I_{fusion}(x, y) = i_{visible}(x, y) \times I_{ir}(x, y)$$

Une telle approche de fusion pour l'image permet de rendre les régions de l'image visible correspondant aux régions « froides » dans l'image infrarouge plus sombres que les régions « chaudes ». La texture de l'image visible est ainsi préservée dans les régions « chaudes ». Un exemple de fusion d'images est présenté à la figure 6.5.

FIGURE 6.5 – Exemples de fusion d'images. Les lignes diffèrent selon le prétraitement effectué. Gauche : Visible, Milieu : Infrarouge, Droite : Image fusionnée.

Cette méthode de fusion a été appliquée à toutes les images de la base Notre–Dame, et spécialement au *Train–Set*. C'est en effet à partir de cet ensemble que l'espace défini à partir des principaux vecteurs propres est appris pour la méthode fondée sur l'ACP. C'est également à partir de cet ensemble que le dictionnaire permettant la décomposition parcimonieuse de patches est appris.

Les résultats de la fusion pour les expériences *Same–session* et *Time–lapse* sont présentés respectivement aux tableaux 6.1 et 6.2. Les résultats de reconnaissance pour l'expérience *Same–session*, qui est une expérience relativement aisée, sont similaires pour les deux approches. Les résultats de l'expérience *Time–lapse* montrent que la méthode fondée sur l'ACP fonctionne moins bien que la variante de la méthode parcimonieuse. Nous pensons que cela est dû au fait que les changements d'illumination sont amplifiés par notre approche de fusion multiplicative. La variante de la méthode parcimonieuse donne des résultats décents, moins bons cependant que les résultats pour la modalité visible seule (Tableau 5.11). Ce niveau de fusion offre toutefois une alternative crédible à la fusion populaire au niveau des scores. Il est en effet possible que ce type de fusion fonctionne mieux que la fusion classique au niveau des scores pour certains scénarios. Les cas d'usage d'une telle approche de fusion restent cependant à étudier.

6.5 Résultats de la fusion au niveau caractéristiques

Afin de tester le niveau de fusion au niveau des caractéristiques, nous avons mené des tests sur la base Notre–Dame. Toutes les images ont été prétraitées de manière similaire à l'expérience de fusion au niveau capteur. La variante de la méthode parcimonieuse a été comparée à la méthode des *eigenfaces* dans les mêmes

Galerie \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF		**0.98** **0.98**	0.97 **0.98**	0.96 **0.97**
FA|LM	0.98 **1.00**		0.91 **0.98**	0.95 **1.00**
FB|LF	0.96 **0.98**	0.93 **1.00**		0.97 **1.00**
FB|LM	0.95 **0.97**	0.95 **1.00**	0.98 **1.00**	

TABLE 6.1 – Taux de reconnaissance au rang 0 pour la fusion au **niveau capteur** (image) de l'expérimentation ***Same-session***. Dans chaque cellule, Haut : Approche *eigenfaces*, Bas : Variante de l'approche parcimonieuse. Meilleur score en gras.

Galerie \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF	0.70 **0.91**	0.68 **0.88**	0.55 **0.85**	0.58 **0.85**
FA|LM	0.70 **0.90**	0.67 **0.88**	0.58 **0.84**	0.60 **0.84**
FB|LF	0.61 **0.83**	0.58 **0.82**	0.63 **0.97**	0.65 **0.89**
FB|LM	0.64 **0.88**	0.61 **0.82**	0.67 **0.80**	0.64 **0.90**

TABLE 6.2 – Taux de reconnaissance au rang 0 pour la fusion au **niveau capteur** (image) de l'expérimentation ***Time–lapse***. Dans chaque cellule, Haut : Approche *eigenfaces*, Bas : Variante de l'approche parcimonieuse. Meilleur score en gras.

conditions.

La fusion au niveau des caractéristiques s'est déroulée en 3 étapes :

- calcul des vecteurs caractéristiques $v_{visible}$ et v_{ir} pour chaque modalité,
- concaténation des deux vecteurs caractéristiques en un seul vecteur de caractéristiques $v_{fusion} = \text{concat}(v_{visible}, v_{ir})$,
- réalisation de l'identification sur les vecteurs fusionnés v_{fusion}.

La taille des vecteurs fusionnés est :

- pour la variante de l'approche parcimonieuse, étant donné que les patches sont

décomposés sur un dictionnaire composé de 200 atomes, le vecteur caractéristique d'un patch pour une modalité est de taille 200, le vecteur fusionné pour un patch est donc de taille 400.

– pour la méthode *eigenfaces*, les vecteurs caractéristiques issus de l'ACP sont de taille $m_{visible}$ et m_{ir} pour les modalités visible et infrarouge. Le vecteur fusionné est donc de taille $m = m_{visible}+m_{ir}$. À noter que les deux espaces issus de l'ACP ne sont pas nécessairement de même dimension, on n'a donc pas forcément $m_{visible} = m_{ir}$. De plus, étant donné que les distances sont calculées via la distance de Mahalanobis, la distance entre deux vecteurs fusionnés n'est donc pas nécessairement la somme des distances entre les « sous–vecteurs » (ce qui aurait été le cas avec la distance L_1 par exemple).

Les résultats de la fusion pour les expériences *Same–session* et *Time–lapse* sont présentés respectivement aux tableaux 6.3 et 6.4. Les taux de reconnaissance pour l'expérience *Same–session* sont un peu meilleurs que ceux obtenus lors de la fusion au niveau capteur (Tableau 6.1). Les taux de reconnaissance sont également améliorés pour l'expérience *Time–lapse* pour les deux approches. Les taux de reconnaissance moyens des 16 sous–expériences sont en effet de 72% et 95% respectivement pour la méthode des *eigenfaces* et la variance de la méthode parcimonieuse (alors qu'ils n'étaient que de 63% et 87% lors de la fusion au niveau capteur). À noter que le taux de reconnaissance moyen de 95% pour l'expérience *Time–lapse* est déjà supérieur aux résultats publiés dans l'article de référence sur cette base de données [63], et ce même avec une méthode de fusion simple comme la concaténation.

Gallery \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF		**1.00** **1.00**	**1.00** **1.00**	**1.00** **1.00**
FA|LM	**1.00** **1.00**		0.95 **1.00**	**1.00** **1.00**
FB|LF	0.98 **1.00**	0.96 **1.00**		0.98 **1.00**
FB|LM	**1.00** 0.98	**1.00** **1.00**	**1.00** **1.00**	

TABLE 6.3 – Taux de reconnaissance au rang 0 pour la fusion au **niveau caractéristiques** de l'expérimentation ***Same-session***. Dans chaque cellule, Haut : Approche *eigenfaces*, Bas : Variante de l'approche parcimonieuse. Meilleur score en gras.

Gallery \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF	0.81 **0.98**	0.80 **0.96**	0.65 **0.95**	0.66 **0.92**
FA|LM	0.82 **0.98**	0.79 **0.96**	0.66 **0.95**	0.67 **0.93**
FB|LF	0.66 **0.93**	0.65 **0.92**	0.77 **0.97**	0.76 **0.97**
FB|LM	0.68 **0.96**	0.68 **0.93**	0.79 **0.96**	0.77 **0.95**

TABLE 6.4 – Taux de reconnaissance au rang 0 pour la fusion au **niveau caractéristiques** de l'expérimentation ***Time–lapse***. Dans chaque cellule, Haut : Approche *eigenfaces*, Bas : Variante de l'approche parcimonieuse. Meilleur score en gras.

6.6 Résultats de la fusion au niveau scores

Dans cette section, nous nous penchons sur le niveau de fusion le plus populaire, la fusion au niveau scores. Les scores obtenus en sortie de chaque *matcher* sont, dans nos expériences, des mesures de distances entre l'empreinte biométrique d'une image test et les empreintes biométriques des images de la galerie.

Nous avons proposé deux approches pour la fusion de scores issus de l'approche neuronale. Cette fusion repose sur le calcul de poids différents pour les deux modalités et une somme pondérée des scores.

L'approche fondée sur la parcimonie (ainsi que sa variante) ont également fait l'objet de test de fusion au niveau des scores.

6.6.1 Première pondération des scores issus de l'approche neuronale

Nous présentons ici une première fonction de pondération des scores issus de l'approche fondée sur les réseaux de neurones convolutionnels pour les deux modalités.

Étant donné que nous effectuons les tests en identification, les scores pour une image test consistent en une distribution de distances représentant les distances entre l'empreinte biométrique de l'image test et les empreintes biométriques de la base de données.

L'approche proposée repose sur une mesure de pertinence calculée dynamiquement pour chaque distance de la distribution. L'idée est que la projection issue du

réseau de reconstruction peut être plus ou moins pertinente selon le succès de l'apprentissage, cette projection peut être de « plus ou moins bonne qualité » selon la modalité considérée.

La pondération proposée repose sur le calcul d'une mesure de pertinence pour chaque distance et pour chaque modalité.

Dans un premier temps, les distances issues des *matchers* sont calculées pour chaque modalité. Les deux distributions de distances obtenues sont ensuite normalisées via la méthode de normalisation **Min–max**. Cette méthode de normalisation est utilisable étant donné que les minimums et maximums de chaque distribution sont connus. Les valeurs de chaque distribution sont ramenées entre 0 et 1.

Le calcul de la pertinence s pour une distance donnée d d'une modalité quelconque est ensuite effectuée en 2 étapes :

- calcul pour la distribution de distances considérée de sa moyenne μ et de son écart–type σ,
- calcul de la pertinence s associée à la distance d selon :

$$s = \sigma\sqrt{2\pi}\frac{1}{e^{-\frac{1}{2}\left(\frac{d-\mu}{\sigma}\right)^2}}$$

Un couple d'images test est composé d'une image pour la modalité visible et d'une pour la modalité infrarouge. Ce couple d'images conduit donc à un couple d'empreintes biométriques, conduisant à deux mesures de distances ($d_{visible}$ et d_{ir}. Ces deux distances possèdant leur pertinence propre ($s_{visible}$ et s_{ir}), la distance finale d est calculée comme la somme des distances pour chaque modalité pondérées par leur pertinence respective :

$$d = \frac{d_{visible} \times s_{visible} + d_{ir} \times s_{ir}}{s_{visible} + s_{ir}}$$

Ce calcul est effectué pour chaque couple de distance correspondant aux distances des images tests aux images de la galerie. L'identité est ensuite trouvée en considérant la distribution de ces distances fusionnées.

Le calcul de pertinence est schématisé à la figure 6.6. La fonction de pertinence correspond à l'inverse d'une gaussienne. La justification de ce choix est que cette fonction va donner plus de poids à une distance se trouvant loin de toutes les autres (même si elle est grande/mauvaise), et va donner un poids plus faible aux distances dès que le réseau n'arrive plus à bien séparer les classes.

Les résultats obtenus par fusion des modalités visible et infrarouge pour les expériences *Same–session* et *Time–lapse* sont présentés aux respectivement aux tableaux 6.5 et 6.6. Les résultats pour chaque modalité prise indépendamment correspondant aux résultats présentés aux tableaux 5.3 et 5.4. Notre approche de fusion des scores est toujours supérieure à chaque modalité prise seule, même lorsque les taux d'identification pour une modalité sont plutôt faibles (comme c'est notre cas en infrarouge).

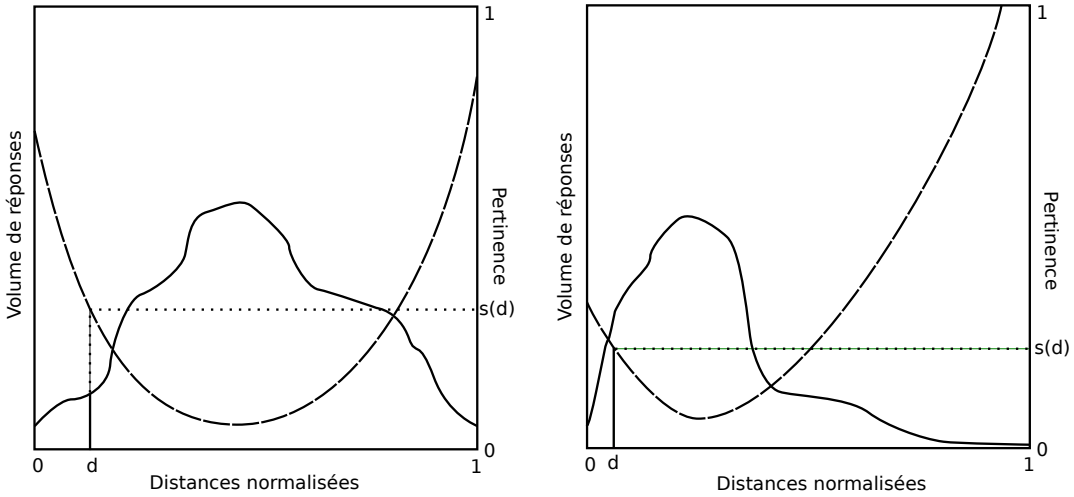


FIGURE 6.6 – Vue schématique du calcul de la pertinence de distances. En trait plein, la distribution normalisée des distances, en long pointillé la fonction de pertinence, et en pointillé court, la valeur $s(d)$ calculée pour une distance d. Dans l'exemple, la distance à gauche a plus de poids bien qu'elle soit supérieure à la distance de droite.

Gallery \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF		1.00 0.90 **1.00**	0.97 0.87 **1.00**	1.00 0.87 **1.00**
FA|LM	1.00 0.95 **1.00**		0.97 0.87 **1.00**	0.97 0.87 **1.00**
FB|LF	0.95 0.97 **1.00**	0.95 0.87 **1.00**		1.00 0.97 **1.00**
FB|LM	1.00 0.95 **1.00**	1.00 0.85 **1.00**	1.00 0.92 **1.00**	

TABLE 6.5 – Taux de reconnaissance au rang 0 pour la fusion au **niveau score** de l'expérimentation ***Same-session*** pour l'approche neuronale avec la première fonction de pondération. Dans chaque cellule, Haut : Visible, Milieu : Infrarouge, Bas : Fusion.

Gallery \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF	0.80 0.41 **0.85**	0.76 0.44 **0.83**	0.68 0.37 **0.75**	0.67 0.38 **0.76**
FA|LM	0.73 0.42 **0.82**	0.75 0.38 **0.80**	0.68 0.34 **0.72**	0.65 0.38 **0.73**
FB|LF	0.72 0.44 **0.82**	0.71 0.37 **0.80**	0.77 0.46 **0.80**	0.78 0.42 **0.88**
FB|LM	0.73 0.43 **0.82**	0.71 0.34 **0.81**	0.73 0.41 **0.80**	0.73 0.42 **0.83**

TABLE 6.6 – Taux de reconnaissance au rang 0 pour la fusion au **niveau score** de l'expérimentation ***Time–lapse*** pour l'approche neuronale avec la première fonction de pondération. Dans chaque cellule, Haut : Visible, Milieu : Infrarouge, Bas : Fusion.

6.6.2 Deuxième pondération

Nous avons proposé une deuxième fonction de pondération pour le calcul de la pertinence des scores obtenus via le réseau de reconstruction. Le même schéma de fusion est appliqué, et la fonction de pertinence considérée est de la forme :

$$s = \left(1 + \frac{1}{2}\tanh\left(\frac{1}{\sigma}\left(d - \mu\right)\right)\right)^{-1}$$

Une vue schématique des fonctions de pertinence est présentée à la figure 6.7.

Cette seconde fonction de pertinence donne beaucoup de poids à une distance qui est vraiment différente des autres pour autant qu'elle soit faible. Comparée à la première fonction de pertinence, une grande distance distante de la moyenne aura un poids faible alors que son poids pour la première fonction était grand. Le reste de la procédure est similaire. Les taux de reconnaissance pour les expériences *Same–session* et *Time–lapse* sont présentés respectivement aux tableaux 6.7 et 6.8.

Les résultats obtenus avec la seconde fonction de pondération donne des résultats légèrement meilleurs qu'avec la première fonction. Cette légère différence tient au fait que les distances fortes obtiennent avec la seconde fonction une pertinence plus faible qu'avec la première fonction.

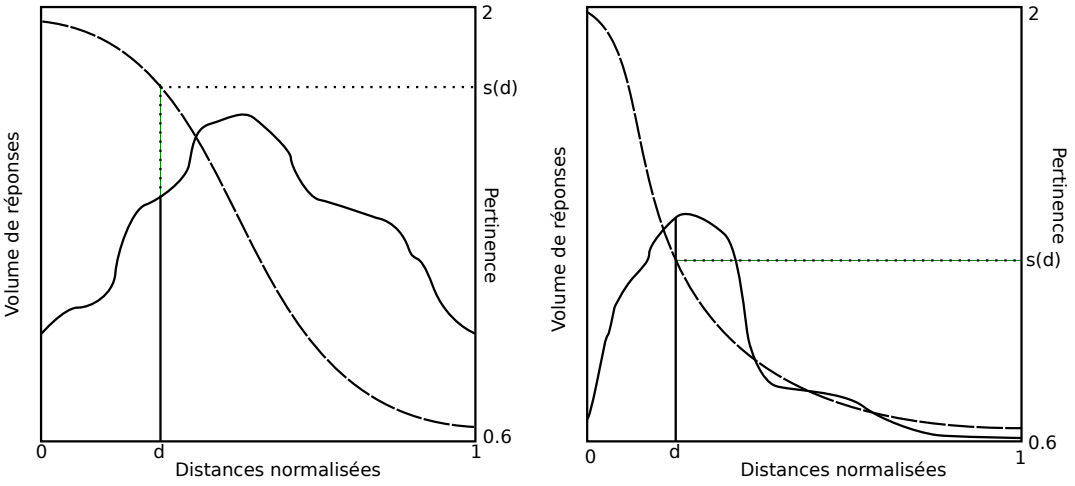


FIGURE 6.7 – Vue schématique du calcul de la pertinence de distances avec la seconde fonction de pertinence considérée.

Gallery \ Probe	FALF	FALM	FBLF	FBLM
FALF		1.00 0.90 **1.00**	0.97 0.87 **1.00**	1.00 0.87 **1.00**
FALM	1.00 0.95 **1.00**		0.97 0.87 **1.00**	0.97 0.87 **1.00**
FBLF	0.95 0.97 **1.00**	0.95 0.87 **1.00**		1.00 0.97 **1.00**
FBLM	1.00 0.95 **1.00**	1.00 0.85 **1.00**	1.00 0.92 **1.00**	

TABLE 6.7 – Taux de reconnaissance au rang 0 pour la fusion au **niveau score** de l'expérimentation ***Same-session*** pour l'approche neuronale avec la deuxième fonction de pondération. Dans chaque cellule, Haut : Visible, Milieu : Infrarouge, Bas : Fusion.

6.6.3 Résultats pour l'approche fondée sur la parcimonie

Dans cette section nous détaillons les résultats obtenus pour la fusion au niveau des scores pour la méthode fondée sur la parcimonie ainsi que sa variante.

Gallery \ Probe	FALF	FALM	FBLF	FBLM
FALF	0.80 0.41 **0.85**	0.76 0.44 **0.83**	0.68 0.37 **0.77**	0.67 0.38 **0.77**
FALM	0.73 0.42 **0.82**	0.75 0.38 **0.81**	0.68 0.34 **0.71**	0.65 0.38 **0.74**
FBLF	0.72 0.44 **0.83**	0.71 0.37 **0.82**	0.77 0.46 **0.82**	0.78 0.42 **0.89**
FBLM	0.73 0.43 **0.85**	0.71 0.34 **0.81**	0.73 0.41 **0.81**	0.73 0.42 **0.83**

TABLE 6.8 – Taux de reconnaissance au rang 0 pour la fusion au **niveau score** de l'expérimentation ***Time–lapse*** pour l'approche neuronale avec la deuxième fonction de pondération. Dans chaque cellule, Haut : Visible, Milieu : Infrarouge, Bas : Fusion.

La méthode de normalisation utilisée est la méthode **Min–max** étant donné que les minimums et maximums des distributions de distances sont connus. La simple règle **somme** est ensuite employée pour fusionner les distances et en obtenir une nouvelle. Les résultats pour les expériences *Same–session* et *Time–lapse* sont présentés respectivement aux tableaux 6.9 et 6.10.

La même approche est utilisée avec la variante de la méthode parcimonieuse. Celle–ci est comparée à la méthode fondée sur l'ACP. Les résultats pour les expérimentations *Same-session* et *Time–lapse* sont présentés respectivement aux tableaux 6.11 et 6.12.

6.7 Résumé des résultats

Dans cette section nous présentons un résumé des principaux résultats obtenus via les méthodes mises en œuvre lors de cette thèse.

Les tableau 6.13 et 6.14 regroupent les taux de reconnaissances moyens pour la fusion respectivement au **niveau capteur** (images) et au **niveau caractéristiques**. La méthode dite des *Eigenfaces* est comparée à la variante de la méthode parcimonieuse pour les deux expérimentations *Same–session* et *Time–lapse*.

Les tableaux 6.15 et 6.16 comparent les résultats obtenus respectivement lors des expériences *Same–session* et *Time–lapse* pour la fusion au **niveau scores**. Les

Gallery \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF		1.00 0.98 **1.00**	1.00 0.97 **1.00**	0.98 1.00 **1.00**
FA|LM	0.98 0.96 **1.00**		1.00 0.95 **1.00**	0.98 0.96 **1.00**
FB|LF	0.97 1.00 **1.00**	0.97 0.92 **1.00**		1.00 0.97 **1.00**
FB|LM	0.98 0.98 **0.98**	0.98 0.97 **1.00**	1.00 0.98 **1.00**	

TABLE 6.9 – Taux de reconnaissance au rang 0 pour la fusion au **niveau score** de l'expérimentation ***Same-session*** pour l'approche parcimonieuse. Haut : Visible, Milieu : IR, Bas : Fusion.

Gallery \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF	0.95 0.83 **0.98**	0.92 0.79 **0.96**	0.87 0.76 **0.94**	0.87 0.77 **0.93**
FA|LM	0.95 0.83 **0.99**	0.93 0.81 **0.97**	0.87 0.77 **0.95**	0.85 0.77 **0.92**
FB|LF	0.86 0.77 **0.93**	0.83 0.74 **0.91**	0.93 0.79 **0.97**	0.91 0.80 **0.95**
FB|LM	0.92 0.79 **0.97**	0.87 0.80 **0.93**	0.88 0.78 **0.95**	0.88 0.82 **0.95**

TABLE 6.10 – Taux de reconnaissance au rang 0 pour la fusion au **niveau score** de l'expérimentation ***Time–lapse*** pour l'approche parcimonieuse. Haut : Visible, Milieu : IR, Bas : Fusion.

méthodes comparées sont : (1) la méthode fondée sur le réseau de reconstruction

Gallery \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF		**1.00** **1.00**	0.98 **1.00**	0.98 **1.00**
FA|LM	**1.00** **1.00**		0.96 **1.00**	**1.00** **1.00**
FB|LF	**1.00** **1.00**	0.96 **1.00**		**1.00** **1.00**
FB|LM	**1.00** 0.98	**1.00** **1.00**	**1.00** **1.00**	

TABLE 6.11 – Taux de reconnaissance au rang 0 pour la fusion au **niveau score** de l'expérimentation ***Same-session***. Haut : Approche *eigenfaces*, Bas : Variante de l'approche parcimonieuse. Meilleur score en gras.

Gallery \ Probe	FA|LF	FA|LM	FB|LF	FB|LM
FA|LF	0.92 **0.98**	0.92 **0.97**	0.75 **0.94**	0.76 **0.94**
FA|LM	0.91 **0.99**	0.91 **0.98**	0.77 **0.96**	0.79 **0.94**
FB|LF	0.81 **0.96**	0.78 **0.94**	0.87 **0.98**	0.87 **0.97**
FB|LM	0.86 **0.98**	0.86 **0.95**	0.86 **0.96**	0.86 **0.97**

TABLE 6.12 – Taux de reconnaissance au rang 0 pour la fusion au **niveau score** de l'expérimentation ***Time–lapse***. Haut : Approche *eigenfaces*, Bas : Variante de l'approche parcimonieuse. Meilleur score en gras.

avec la première fonction de pondération, (2) la même méthode avec la deuxième fonction de pondération, (3) la méthode parcimonieuse, (4) la variante de la méthode parcimonieuse, (5) la méthode des *eigenfaces*, et (6) la méthode présentée dans [63] fondée sur une ACP et une fusion particulière (résultats pour l'expérience *Same–session* non reportés).

	Same-session		*Time-lapse*	
	(1)	(2)	(1)	(2)
Moyenne	0.95	**0.98**	0.63	**0.87**
Écart–type	0.02	0.01	0.04	0.04

TABLE 6.13 – Taux de reconnaissance moyens au rang 0 et écart–type pour les 12 (***Same–session***) et 16 (***Time–lapse***) sous–expérimentations pour la fusion au **niveau capteur**. (1) *Eigenfaces*, (2) variante de la méthode parcimonieuse. En gras, les meilleurs taux de reconnaissance moyens pour chaque expérimentation.

	Same-session		*Time-lapse*	
	(1)	(2)	(1)	(2)
Moyenne	0.98	**0.99**	0.72	**0.95**
Écart–type	0.01	0.01	0.06	0.02

TABLE 6.14 – Taux de reconnaissance moyens au rang 0 et écart–type pour les 12 (***Same–sessions***) et 16 (***Time–lapse***) sous–expérimentations pour la fusion au **niveau caractéristiques**. (1) *Eigenfaces*, (2) variante de la méthode parcimonieuse. En gras, les meilleurs taux de reconnaissance moyens pour chaque expérimentation.

	(1)	(2)	(3)	(4)	(5)
Moyenne	**1.00**	**1.00**	0.99	0.99	0.99
Écart–type	0	0	0.01	0.01	0.01

TABLE 6.15 – Taux de reconnaissance moyens au rang 0 et écart–type pour les 12 sous–expérimentations de l'expérience ***Same–session*** pour la fusion au **niveau score**. (1) Réseau de reconstruction, première fonction de pondération, (2) Réseau de reconstruction, deuxième fonction de pondération, (3) Approche parcimonieuse, (4) Variante de l'approche parcimonieuse, (5) *Eigenfaces*. Meilleurs taux moyens de reconnaissance en gras.

6.8 Conclusion

Dans ce chapitre, nous avons présenté les résultats des différentes méthodes de fusion mises en œuvre avec les méthodes utilisées lors de cette thèse. Tous les tests ont été effectués sur la base de données Notre–Dame, cette base présentant des avantages certains concernant les dates de capture des images des deux modalités.

Deux méthodes de fusion au niveau des **scores** ont été présentées pour l'approche fondée sur les réseaux de neurones convolutionnels. Celles–ci reposent sur

	(1)	(2)	(3)	(4)	(5)	(6)
Moyenne	0.80	0.81	0.95	**0.96**	0.84	0.92
Écart–type	0.04	0.04	0.01	0.01	0.05	0.02

TABLE 6.16 – Taux de reconnaissance moyens au rang 0 et écart–type pour les 16 sous–expérimentations de l'expérience ***Time–lapse*** pour la fusion au **niveau score**. (1) Réseau de reconstruction, première fonction de pondération, (2) Réseau de reconstruction, deuxième fonction de pondération, (3) Approche parcimonieuse, (4) Variante de l'approche parcimonieuse, (5) *Eigenfaces*, (6) Résultats présentés dans [63]. Meilleur taux moyen de reconnaissance en gras.

le calcul pour chaque score d'une mesure de poids. Ce poids est calculé directement à partir de la distribution des scores, permettant ainsi de quantifier la pertinence du vecteur projeté issu de l'architecture neuronale. Cette approche de fusion permet d'améliorer les scores de chaque modalité prise indépendamment. Cette méthode a également été appliquée aux méthodes parcimonieuses développées lors de la thèse. Les résultats se sont révélés cependant être similaires à la méthode plus simple décrite dans ce chapitre (normalisation des scores et somme de ceux–ci).

De tous les résultats présentés dans ce chapitre, nous pouvons tirer trois conclusions :

- les taux de reconnaissance pour la modalité visible sont toujours (ou presque) supérieurs aux taux de reconnaissance obtenus pour la modalité infrarouge,
- la fusion au niveau des caractéristiques permet d'améliorer les taux de reconnaissance, dépassant toujours les taux de chaque modalité prise séparemment,
- les taux de reconnaissance aux niveaux **capteur** (fusion d'images) et **caractéristiques** sont suffisamment corrects pour offrir une alternative crédible à la classique fusion au niveau des **scores**.

Enfin une remarque peut être faite concernant les différents sous–ensembles de la base de données Notre–Dame. Même si ce n'est pas toujours le cas, il semble que les taux de reconnaissance soient globalement supérieurs lorsque les images d'enrôlement et les images tests ont la même expression faciale, les variations d'illumination jouant un rôle moindre.

Plus en détails, les résultats pour l'expérience *Same–session* ne sont pas significativement différents pour toutes les approches reportées dans ce chapitre, ce test étant relativement (trop) simple.

Les résultats pour l'expérience *Time–lapse* sont moins bons que l'état de l'art pour l'approche neuronale. Cela vient du fait que celle–ci repose sur un apprentissage des images dans leur ensemble, et étant donné le manque de variations des images d'apprentissage (définies dans le protocole), les taux de reconnaissance sont

moins bons. Les résultats obtenus via l'approche parcimonieuse (et sa variante) sont en revanche meilleurs. Cela vient du fait que cette méthode ne requiert pas d'apprentissage global des images, mais simplement de dictionnaires. Ces dictionnaires permettent une description locale des visages (via les patches), et le manque de variabilité de l'ensemble d'apprentissage n'est ainsi plus un obstacle. À noter également que tous les tests correspondent à un scénario « *One–image–to–enroll* » où seule une image est utilisée pour l'enrôlement. Ce type de scénario pénalise fortement les approches neuronales où les vecteurs projetés (les empreintes biométriques) sont plus sujets aux variations inter–classe.

À noter que des architectures neuronales permettant directement la fusion à différents niveaux ont été testées lors de la thèse. Ainsi, différentes architectures de réseaux de neurones convolutionnels prenant en entrées non plus une seule image mais un couple d'images visible et infrarouge ont été testées. Pour ces architectures, de nombreux tests ont été effectués pour intégrer la fusion à différents niveaux, selon quelle couche allait faire intervenir la fusion des sorties des neurones issus de l'image visible et de l'image infrarouge. Deux principales architectures ont été testées, chacune déclinée en autant de versions que de possibilités de fusion dans les couches de convolution. La première similaire au réseau de reconstruction unimodal essaie de reconstruire les deux images présentes en entrée (ou des images de la même personne). La seconde n'a en sortie qu'un seul neurone qui prend la valeur 1 si les deux images en entrées appartiennent à la même personne, -1 sinon.

Ces architectures avec plusieurs entrées n'ont cependant pas donné de résultats probants, et leur développement n'a donc pas été poursuivi. Nous pensons que l'apprentissage s'est révélé déficient étant donné le trop grand nombre de données à apprendre ainsi que la trop grande variabilité des images en entrée. L'idée du préapprentissage des premières couches n'a cependant pas été testée avec de telles architectures, il se peut que des préapprentissages séparés pour les deux modalités, puis intégrés au sein de ce type d'architecture permettent de lever les principaux verrous évoqués.

La fusion des modalités visible et infrarouge pour la reconnaissance faciale permet de tirer parti des informations complémentaires offertes par ces deux modalités. Dans le cas de grands changements de luminosité par exemple, la modalité infrarouge peut avantageusement venir compléter (voire complètement remplacer) la modalité visible. De même, dans le cas de grandes variations de chaleur, la modalité infrarouge peut être déficiente, auquel cas la modalité visible peut être la composante prépondérante d'un système fondé sur ces deux modalités.

Les différents niveaux de fusion peuvent ensuite être choisis pour un système multimodal selon le type d'application visé. Par exemple, si la méthode de reconnaissance utilisée par le système est coûteuse (en temps de calcul ou espace mé-

moire), il peut être intéressant de considérer la fusion au niveau capteur, l'algorithme n'aurait ainsi à être appliqué qu'une seule fois (sur les images fusionnées) et non pas deux fois (une fois par modalité).

Enfin, un système biométrique capable de mesurer les conditions extérieures (luminosité et/ou température) pourrait ajuster le niveau de fusion en fonction de ces mesures afin de pallier les problèmes induits par ces conditions externes.

Chapitre 7

Conclusion et perspectives

Dans cette thèse, nous avons décrit le problème de la reconnaissance automatique de visages en lumière visible et infrarouge (grandes longueurs d'ondes). Les principales méthodes de la littérature ont été étudiées, et nous nous sommes plus particulièrement concentré sur une méthode fondée sur les réseaux de neurones convolutionnels ainsi que sur une méthode fondée sur la parcimonie.

Réseaux de neurones convolutionnels Dans un premier temps, une architecture de réseau de neurones convolutionnels a été étudiée. Celle–ci a été préférée parmi plusieurs pour les meilleurs taux de reconnaissance obtenus *a posteriori*. Cette architecture peut se décomposer en deux parties : la première, composée d'une succession de couches de convolution et de subsampling, permet d'extraire l'information de l'image d'entrée, de fusionner ces caractéristiques afin d'obtenir une représentation de plus haut niveau. Cette première partie peut être vue comme un moyen de projeter non–linéairement les images d'entrée du réseau sur un espace de plus faible dimension. La deuxième partie essaie de reconstruire un visage de référence à partir de la représentation compacte issue de la projection. Lors de l'apprentissage, le réseau reconstruit une image référence de visage par personne, rendant ainsi la projection quasi–invariante aux transformations présentées en entrée. L'intérêt des réseaux de convolution est qu'il n'y a pas à choisir *a priori* d'extracteurs de caractéristiques particuliers, les premières couches réalisant cette tâche étant mises à jour lors de l'apprentissage.

Les scénarios d'identification considérés dans cette thèse n'utilisent qu'un nombre limité d'images d'apprentissage ainsi qu'une seule image pour l'enrôlement. Malgré cette limitation, les taux de reconnaissance atteints pour la modalité visible sont comparables à l'état de l'art sur la base de données Notre–Dame. En revanche, les taux de reconnaissance sont faibles pour les visages capturés en lumière infrarouge. Nous pensons que les variations de chaleur, étant plus localisées que des

variations de luminosité pour la modalité visible, induisent une trop grande variation intra–classe qui ne permet pas une bonne classification. Dès que le nombre d'images utilisées pour l'enrôlement augmente, les taux de reconnaissance deviennent similaires à ceux de l'état de l'art.

Une des limitations majeures des approches fondées sur un apprentissage en général, et sur les réseaux de neurones convolutionnels en particulier, est qu'elles nécessitent un grand nombre d'échantillons d'apprentissage afin d'obtenir une extraction de caractéristiques pertinentes ainsi qu'une bonne généralisation. Une manière de surmonter cette limitation consiste à effectuer des préapprentissages de certaines couches du réseau. En particulier, des préapprentissages effectués avec des méthodes parcimonieuses permettent de fixer les poids des couches de convolution, rendant les caractéristiques extraites plus pertinentes. Dans le cas de la reconnaissance faciale, les taux de reconnaissance sont ainsi améliorés. Si les tables de connexion entre couches permettaient à un apprentissage classique de guider l'apprentissage de la partie extraction de caractéristiques, celles–ci peuvent poser problème pour le cas où un préapprentissage est effectué. En effet, une table de connexion est essentiellement utilisée pour caractériser la fusion de caractéristiques, et une telle fusion pour des caractéristiques issues de filtres appris de manière non–supervisée peut ne pas être pertinente.

Cette question des tables de connexion dans le cas de réseaux de neurones convolutionnels avec préapprentissage reste un problème ouvert.

Décompositions parcimonieuses Dans un second temps, une méthode originale fondée sur une décomposition parcimonieuse des patches de l'image d'un visage sur un dictionnaire appris a été développée lors de la thèse. Utilisée conjointement à une méthode de classification reposant sur l'idée de parcimonie, cette méthode obtient des taux de reconnaissance similaires à l'état de l'art pour la modalité infrarouge, mais nettement supérieurs pour la modalité visible, selon le protocole dédié à la base de données Notre–Dame. Des tests ont de plus montré sa bonne robustesse à des dégradations des images tests. Le développement d'une variante de cette méthode a permis d'accroître encore les taux de reconnaissance.

Cette méthode souffre malgré tout de quelques défauts, remédiables cependant. Ainsi, il est nécessaire de normaliser géométriquement les visages précisément. De plus, cette méthode ne devrait pas fonctionner correctement si les visages tests ont subi une rotation hors plan tandis que les visages d'enrôlement sont frontaux. Cette méthode n'est donc pas robuste aux changements de pose. De récents travaux sur l'apprentissage de dictionnaires invariants à des translations permet cependant d'espérer obtenir des méthodes d'apprentissage de dictionnaires qui soient invariants aux rotations. De tels dictionnaires permettraient de s'affranchir de la contrainte de normalisation géométrique.

Fusion Dans un troisième temps, la fusion de modalités a été considérée. Différents niveaux de fusion ont été étudiés. La fusion des images (niveau capteur) ainsi que la fusion au niveau des caractéristiques ont été considérées via l'approche parcimonieuse. La fusion la plus étudiée est la fusion au niveau des scores. Une méthode permettant de pondérer les scores issus des réseaux de neurones convolutionnels a ainsi été développée. Celle–ci permet de prendre en compte la pertinence des projections des réseaux à l'aide d'une fonction de pondération. Les approches parcimonieuses ont également été utilisées comme modules de mise en correspondance avant une fusion des scores des modalités. Les méthodes parcimonieuses ont permis l'utilisation de règles simples de fusion (somme des scores) pour obtenir des taux de reconnaissance élevés.

La fusion de modalités offre une alternative aux systèmes biométriques unimodaux. Partant de l'hypothèse que les modalités offrent des informations complémentaires (ce qui est souvent le cas), la fusion de celles–ci permet globalement d'améliorer la fiabilité d'un système. La fusion de modalités permet en outre de s'affranchir de certaines problématiques inhérentes aux systèmes unimodaux (comme les variations de luminosité pour la modalité visible par exemple). Dans le cadre d'une fusion multi–capteurs, elle nécessite cependant un investissement supplémentaire en capteurs, ce qui peut être coûteux.

L'approche neuronale présente de moins bons résultats de reconnaissance que la méthode fondée sur la parcimonie. Celle–ci est cependant bien plus dédiée à la reconnaissance faciale, et les prétraitements plus contraignants, alors que l'approche neuronale est plus générale. Les réseaux de neurones convolutionnels ont d'ailleurs été utilisés dans le cadre d'un stage effectué au GREYC dans le cadre de l'ANR Biotyful. Ceux–ci ont permis d'améliorer le module de détection des points saillants d'une main dans le cadre d'une reconnaissance d'individus via la paume de leur main (thèse de Julien Doublet [87]).

Perspectives

Réseaux de neurones convolutionnels La construction d'un modèle mathématique permettant d'adapter l'architecture (nombre de neurones par couches, taille des noyaux, connexions entre couches ...) d'un réseau de neurones convolutionnels à un problème donné (dimension du problème, nombre d'échantillons ...) est un sujet nécessitant encore de nombreux travaux. En effet, les architectures utilisées sont construites bien souvent sur une bonne intuition ou empiriquement par essais/erreurs, ce qui est coûteux en temps. L'utilisation de méthodes de préapprentissage permet de résoudre, en partie seulement, ce problème mais la question des tables de connexions décrivant les liens entre neurones se pose alors.

Méthodes parcimonieuses L'utilisation de dictionnaires invariants à certaines transformations (comme la rotation dans le plan évoquée plus haut) devrait permettre de s'affranchir de la nécessité de prétraitement contraignants. L'utilisation de tels dictionnaires nécessiterait cependant la mise au point d'algorithmes de décomposition plus rapide pour une utilisation facilitée.

Classification Le probème de la robustesse d'un système biométrique à diverses altérations (luminosité, rotations, mises à l'échelle ...) peut être attaqué en trois points de la chaîne. La plupart des systèmes biométriques n'en considèrent cependant que deux : améliorer les images avant d'effectuer une extraction de caractéristiques, et/ou extraire des caractéristiques invariantes aux transformations.

Le premier niveau peut consister par exemple en un recadrage géométrique pour éliminer les transformations géométriques subies par l'image, ou en une modification de la dynamique des niveaux de gris pour pallier le problème de la luminosité.

Le deuxième niveau généralement considéré consiste à produire des algorithmes capables d'extraire des caractéristiques invariantes aux dites transformations. La multiplication des transformations à considérer peut cependant rendre les algorithmes très complexes, et donc instables à de nouvelles transformations.

Un troisième niveau pourrait consister à créer de nouvelles images à partir des images d'enrôlement en les modifiant via les transformations auxquelles on souhaite être invariant. Par exemple, à partir d'une image d'enrôlement, créer des images artificielles par rotation, puis les intégrer à la galerie. Une telle approche, combinée à l'algorithme de classification SRC fondé sur la parcimonie, permettrait à une image test ayant subi une rotation d'être mise en correspondance facilement avec son équivalent artificiel, alors qu'elle ne le serait pas forcément avec l'image originale présente dans la galerie.

Une telle approche pour l'invariance par transformations affines serait non seulement simple à réaliser (il suffit de créer autant d'images artificielles que de transformations auxquelles le système doit être robuste), et devrait pouvoir fonctionner avec des algorithmes d'extraction de caractéristiques simples. Un problème se poserait néanmoins avec cette approche : la multiplication des images de la galerie, et donc le temps de calcul nécessaire à la décomposition parcimonieuse d'un vecteur caractéristique sur cette galerie.

Annexe A

Méthodes de réduction de dimension

Dans cette annexe, nous nous attachons à décrire plusieurs méthodes de réduction de dimension. La réduction de dimension consiste à transformer des données représentées dans un espace de grande dimension en une représentation dans une espace de dimension plus faible. Idéalement, la nouvelle représentation a une dimension égale au nombre de paramètres nécessaires pour décrire les données observées [109]. La réduction de dimension est importante dans de nombreux domaines étant donné qu'elle facilite la classification, la visualisation ou encore la compression de données de grande dimension. Elle permet également souvent de limiter l'effet de la malédiction de la dimension et d'autres propriétés non désirées des espaces de grande dimension [152].

Récemment, un grand nombre de méthodes de réduction de dimension ont été proposées [35], [82], [135], [172], [246], [255], [272], [275], [306]. Ces techniques sont capables de traiter des problèmes complexes non–linéaires et ont souvent été proposées comme une alternative aux techniques linéaires classiques telles l'analyse en composantes principales (ACP) ou l'analyse discriminante linéaire (LDA).

De précédentes études ont en effet montré que les approches non–linéaires surpassent les méthodes linéaires sur des jeux de données artificiels hautement non–linéaires, comme le *Swiss roll* (voir Figure A.1).

Cependant, les succès de réduction de dimension avec les méthodes non–linéaires sur des jeux de données naturelles sont plutôt rares. Dans la suite, nous décrivons plusieurs techniques linéaires classiques telles l'Analyse en Composantes Principales (ACP) [138], la factorisation non–négative de matrices (NMF) [48], l'Analyse en Composantes Indépendantes (ICA) [127] et l'Analyse Discriminante Linéaire (LDA) [107], ainsi que dix méthodes non–linéaires (le nom de chaque méthode n'a volontairement pas été traduit) : multidimensional scaling (MDS) [71], [166], Isomap [274], [275], Kernel PCA [204], [255], diffusion maps [172], [214], multilayer autoencoders [78], [135], Locally Linear Embedding (LLE) [246], Laplacian

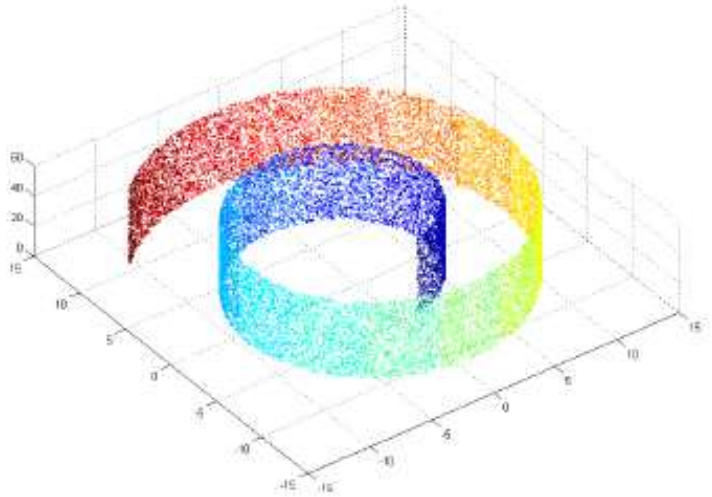


FIGURE A.1 – Swiss Roll.

Eigenmaps [35], Hessian LLE [82], Local Tangent Space Analysis (LTSA) [306], et Locally Linear Coordination (LLC) [272].

D'autres techniques non–linéaires ont été proposées, telles que Principal Curves [61], Generalized Discriminant Analysis [32], Kernel maps [268], Maximum variance unfolding [291], Conformal eigenmaps [257], Locality Preserving Projections [128], Linear Local Tangent Space Alignment [305], Stochastic Proximity Embedding [25], FastMap [102], Geodesic Nullspace Analysis [45]. La plupart d'entre elles sont des variantes des dix méthodes énoncées plus haut, et ne seront donc pas décrites ici.

A.1 La réduction de dimension

Supposons qu'un jeu de données soit décrit par la matrice X de taille $n \times D$ où n est le nombre de vecteurs $\boldsymbol{x_i}$ de dimension D. Ce jeu de données possède une dimension propre (ou intrinsèque) d, où $d < D$ voire $d \ll D$. En termes mathématiques, la dimension intrinsèque signifie que le jeu de données repose sur une variété de dimension d, contenu dans un espace de plus grande dimension D. Une technique de réduction de dimension transforme le jeu de données X en un nouvel ensemble Y de dimension d, en gardant au maximum l'essentiel de l'information de l'ensemble de départ. Généralement, ni la géométrie de la variété, ni d sont connus. Les techniques de réduction de dimension peuvent être classées en plusieurs groupes (voir la figure A.2). La principale critère de classement est l'aspect linéaire ou non des méthodes. Les méthodes linéaires supposent que les données reposent sur une variété linéaire

de l'espace de grande dimension. Les méthodes non–linéaires ne reposent pas sur cette hypothèse et sont capables de caractériser des variétés plus complexes.

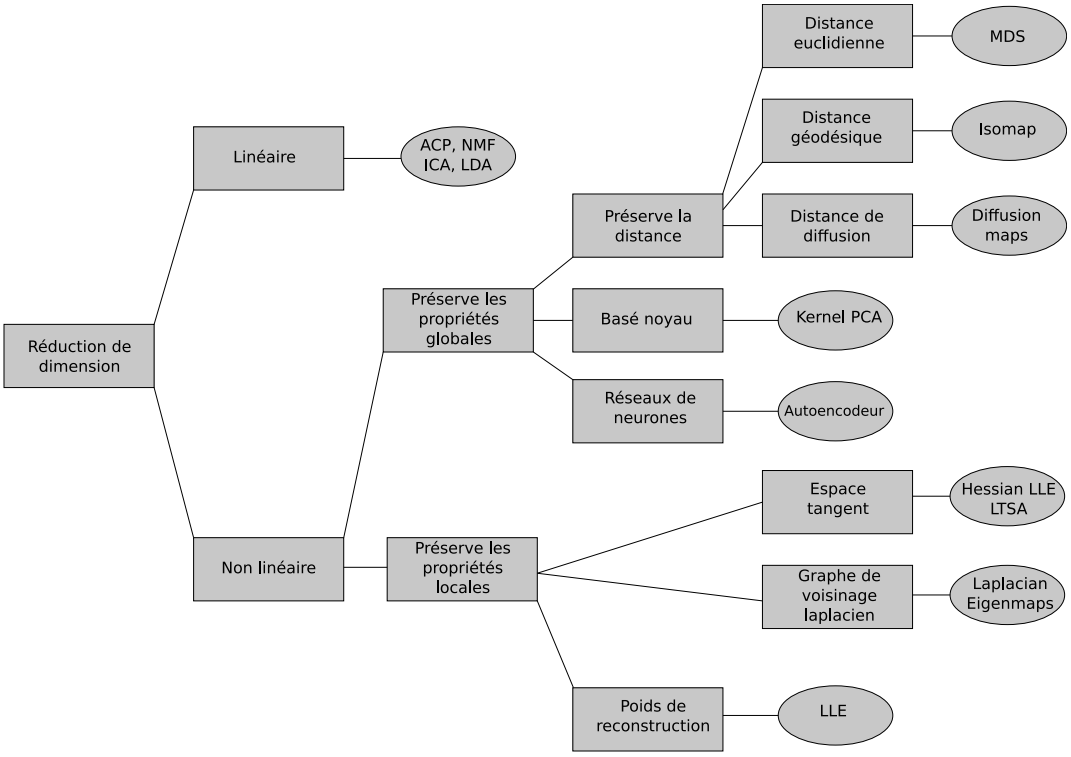


FIGURE A.2 – Taxonomie des techniques de réduction de dimension.

A.2 Méthodes linéaires de réduction de dimension

Nous décrivons ici quatre des méthodes linéaires les plus couramment utilisées : l'Analyse en Composantes Principales (ACP), la Factorisaion de Matrices Non–négatives (NMF), l'Analyse en Composantes Indépendantes (ICA) et l'Analyse Discriminante Linéaire (LDA).

A.2.1 L'Analyse en Composantes Principales

L'Analyse en Composantes Principales (ACP) [153], aussi connue sous le nom de transformée de Karhunen–Loève [193] est une méthode très utilisée en statistique. Introduite par Pearson [228] puis plus tard par Hotelling [138], sa principale idée est de réduire la dimension d'un jeu de données tout en gardant un maximum

d'informations. Cela est réalisé grâce à une projection qui maximise la variance tout en minimisant l'erreur quadratique moyenne de la reconstruction. Pour plus de détails, voir [79], [153], [201], [262]

Dérivation de l'ACP Hotelling définit l'ACP comme une projection orthogonale maximisant la variance dans l'espace projeté. Étant donné n échantillons $\boldsymbol{x_i} \in \mathbb{R}^D$ et $\boldsymbol{u} \in \mathbb{R}^D$ tel que

$$\|\boldsymbol{u}\| = \boldsymbol{u}^T\boldsymbol{u} = 1$$

soit un vecteur orthonormal de projection. Un échantillon $\boldsymbol{x_i}$ est projeté sur $\boldsymbol{u}$ par :

$$a_i = \boldsymbol{u}^T\boldsymbol{x_i}.$$

La variance de l'échantillon peut donc être estimée :

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})$$

où $\bar{x}$ est la moyenne des projetés des échantillons de la base :

$$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x_i}$$

d'où

$$\bar{a} = \boldsymbol{u}^T\bar{\boldsymbol{x}}$$

Ainsi la variance du projeté est donnée par :

$$\begin{aligned} S^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(a_i - \bar{a}) && \text{(A.1)} \\ &= \frac{1}{n-1}\sum_{i=1}^{n}\left(\boldsymbol{u}^T\boldsymbol{x} - \boldsymbol{u}^T\bar{\boldsymbol{x}}\right) && \text{(A.2)} \\ &= \boldsymbol{u}^T\boldsymbol{C}\boldsymbol{u} && \text{(A.3)} \end{aligned}$$

où

$$\boldsymbol{C} \in \mathbb{R}^{D\times D} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\boldsymbol{x_i} - \bar{\boldsymbol{x}}\right)\left(\boldsymbol{x_i} - \bar{\boldsymbol{x}}\right)^T$$

est la matrice de covariance de $X = [\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}] \in \mathbb{R}^{D\times n}$. Le problème de maximisation de la variance dans l'espace projeté peut donc s'écrire :

$$\max \boldsymbol{u}^T\boldsymbol{C}\boldsymbol{u} \qquad \text{avec } \boldsymbol{u}^T\boldsymbol{u} = 1$$

Le calcul de la solution optimale peut être réalisée grâce au multiplicateur de Lagrange :

$$f(\boldsymbol{u}, \lambda) = \boldsymbol{u}^T \boldsymbol{C}\boldsymbol{U} + \lambda \left(1 - \boldsymbol{u}^T \boldsymbol{u}\right)$$

Par dérivation partielle selon $\boldsymbol{u}$

$$\frac{\partial f(\boldsymbol{u}, \lambda)}{\partial \boldsymbol{u}} = 2\boldsymbol{C}\boldsymbol{u} - 2\lambda \boldsymbol{u} = 0$$

on obtient

$$\boldsymbol{C}\boldsymbol{u} = \lambda \boldsymbol{u}$$

Ainsi, le maximum pour le multiplicateur de Lagrance est obtenu si λ est une valeur propre et $\boldsymbol{u}$ un vecteur propre de $\boldsymbol{C}$. Ainsi la variance décrite par le vecteur de projection $\boldsymbol{u}$ est donnée par λ.

D'autres méthodes de dérivation de l'ACP sont données dans [41], [153]. Pour une vue probabiliste de la dérivation de l'ACP, voir [245], [277].

Calcul batch de l'ACP Pour la mise en œuvre de méthodes batch, il est supposé que le jeu de données d'entraînement est disponible en entier. Ainsi nous avons un ensemble de n observations $x_i \in \mathbb{R}^D$ organisés sous forme matricielle $X = [\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}] \in \mathbb{R}^{D \times n}$. L'estimation de la base de projection de l'ACP revient donc à estimer les éléments propres de la matrice de covariance C de X. Le calcul requiert d'abord l'échantillon moyen

$$\bar{\boldsymbol{x}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x_i}$$

puis les échantillons sont normalisés par rapport à la moyenne $\bar{\boldsymbol{x}}$:

$$\hat{\boldsymbol{x_i}} = \boldsymbol{x_i} - \bar{\boldsymbol{x}}$$

pour former la nouvelle matrice $\hat{X} = [\hat{\boldsymbol{x_1}}, \ldots, \hat{\boldsymbol{x_n}}]$. La matrice de covariance $C \in \mathbb{R}^{D \times D}$ est ensuite calculée par :

$$C = \frac{1}{n-1} \hat{X} \hat{X}^T$$

La recherche des éléments propres de C conduit à l'obtention de la base de vecteurs propres $\boldsymbol{u_i} \in \mathbb{R}^D$, pour lesquels, à chacun d'eux, est associée une valeur propre λ_i. Généralement triés par ordre décroissant de valeur propre associée, les premiers vecteurs propres forment alors une base dans laquelle la plupart de l'information du jeu de données d'entraînement est gardée.

ACP pour des données de grande dimension La dimension de la matrice de covariance dépend de la dimension D des vecteurs du jeu de données, qui peut être relativement grande pour certains types de données (typiquement des images). La méthode décrite plus haut devient alors difficile à appliquer, essentiellement à cause de la recheche des éléments propres de la matrice de covariance C. En effet, pour des images de taille 100×100 par exemple, la matrice de covariance C à inverser est de taille 10000×10000. Cependant, il est connu que pour toute matrice X, les produits matriciels XX^T et X^TX partagent les mêmes valeurs propres différentes de zéro. Ainsi, le calcul des éléments propres de $C = XX^T \in \mathbb{R}^{D \times D}$ peut se ramener au calcul des éléments propres de la matrice $M \in \mathbb{R}^{n \times n}$ où $M = X^TX$. Soit $\boldsymbol{e_i}$ les vecteurs propres de M associés aux valeurs propres δ_i. On a donc :

$$X^TX\boldsymbol{e_i} = \delta_i\boldsymbol{e_i}$$

En multipliant à gauche par X les deux côtés de l'équation, on obtient ainsi :

$$X\left(X^TX\boldsymbol{e_i}\right) = X\left(\delta_i\boldsymbol{e_i}\right) \quad \text{(A.4)}$$

$$XX^T\left(X\boldsymbol{e_i}\right) = \delta_i\left(X\boldsymbol{e_i}\right) \quad \text{(A.5)}$$

On voit donc que $X\boldsymbol{e_i}$ est vecteur propre de XX^T et que δ_i est la valeur propre associée, d'où

$$\begin{cases} \boldsymbol{u_i} = X\boldsymbol{e_i} \\ \lambda_i = \delta_i \end{cases}$$

La matrice M étant beaucoup plus petite que la matrice C (typiquement, on passe d'une complexité de l'ordre de la dimension des échantillons à une complexité de l'ordre du nombre d'échantillons d'apprentissage), les calculs sont donc plus efficaces. L'algorithme de l'Analyse en Composantes Principales est résumé à l'Algorithme 8.

Des variantes de l'ACP ont été proposées. Ainsi plusieurs méthodes ont été proposées pour extraire des axes principaux robustes notamment au bruit contenu dans les images d'apprentissage [170] [300], ou des méthodes basées sur une formulation Espérance–Maximisation de l'ACP [245], [263], [277]. Dans le cas où les données d'apprentissage ne sont pas toutes disponibles au départ (cas de vidéos par exemple), des versions incrémentales de l'ACP ont été mises au point [44], [304], [124], [212]. Des méthodes combinant l'aspect incrémental et robuste ont également été

proposées dans [187], [264].

Algorithme 8: Calcul de l'ACP

Entrées : matrice X

Sorties : vecteur moyen $\bar{x}$, base de vecteurs propres U, valeurs propres associées λ_i

Calcul du vecteur moyen :

$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x_i}$

Normalisation des images d'entrées :

$\hat{\boldsymbol{x}}_{\boldsymbol{i}} = \boldsymbol{x_i} - \bar{\boldsymbol{x}}$

$\hat{X} = [\hat{\boldsymbol{x}}_{\mathbf{1}}, \ldots, \hat{\boldsymbol{x}}_{\boldsymbol{n}}]$

si *Données de grande dimension* **alors**

$M = \frac{1}{n-1}\hat{X}^T\hat{X}$

Calcul des éléments propres de M :

$E = [\boldsymbol{e}_{\mathbf{1}}, \ldots, \boldsymbol{e}_{\boldsymbol{n}}]$

$\boldsymbol{\delta} = [\delta_1, \ldots, \delta_n]$

Calcul des éléments finaux :

$\boldsymbol{u_i} = X\boldsymbol{e_i}, \quad U = [\boldsymbol{u}_{\mathbf{1}}, \ldots, \boldsymbol{u}_{\boldsymbol{n}}]$

$\lambda_i = \delta_i, \quad \boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]$

sinon

$C = \frac{1}{n-1}\hat{X}\hat{X}^T$

Calcul des éléments propres de C :

$U = [\boldsymbol{u}_{\mathbf{1}}, \ldots, \boldsymbol{u}_{\boldsymbol{n}}]$

$\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]$

retourner $\bar{\boldsymbol{x}}$,U,$\boldsymbol{\lambda}$

A.2.2 Factorisation de matrice non–négative

La factorisation de matrice non–négative(ou NMF pour *Non–Negative Matrix Factorization*) a été proposée dans [225] et [259]. Introduite dans le cadre de la vision par ordinateur dans [180], cette technique, contrairement à l'ACP, n'autorise pas de valeurs négatives dans les vecteurs de base ni dans les vecteurs de projection. Les vecteurs de base sont donc additifs et représentent des structures locales.

Plus formellement, la méthode peut être décrite ainsi : Étant donnée une matrice $V \in \mathbb{R}^{n\times m}$ positive contenant les images vectorisées, le but est de trouver les matrices non–négatives $W \in \mathbb{R}^{n\times r}$ et $H \in \mathbb{R}^{r\times m}$ qui approximent la matrice V :

$$V \approx WH$$

Les deux matrices W et H doivent être estimées itérativement en considérant le problème d'optimisation suivant :

$$\min \|V - WH\|_2^2 \quad \text{avec } W, H > 0$$

Les règles de mise à jour pour les matrices W et H sont alors :

$$H_{i,j} \leftarrow H_{i,j} \frac{\left(W^T V\right)_{i,j}}{\left(W^T W H\right)_{i,j}} \tag{A.6}$$

$$W_{i,j} \leftarrow W_{i,j} \frac{\left(V H^T\right)_{i,j}}{\left(W H H^T\right)_{i,j}} \tag{A.7}$$

Plus de détails sur la dérivation de la méthode ainsi que sur des descriptions de l'algorithme peuvent être trouvées dans [181] et [294]. De plus, pour améliorer la rapidité de l'algorithme ainsi que pour s'assurer que la solution trouvée soit le minimum global (le problème d'optimisation n'est en effet pas convexe en W ni en H), plusieurs extensions ont été proposées [131], [141]. Elles considérent une contrainte additionnelle de parcimonie et reformulent le problème en un problème convexe.

A.2.3 Analyse en composantes indépendantes

L'Analyse en Composantes indépendantes (ou ICA pour *Independant Component Analysis*) a été introduite par Hérault, Jutten et Ans dans [27], [145] et [146] dans le contexte de la neurophysiologie. Elle devint populaire lors de son utilisation dans le domaine du traitement du signal pour la séparation de sources aveugles dans [68] et [154]. Le but est d'exprimer un ensemble de n variables aléatoires $x_1, \ldots, x_n$ comme une combinaison linéaire de n variables aléatoires statistiquement indépendantes s_j :

$$x_j = a_{j,1} s_1 + \cdots + a_{j,n} s_n \quad \forall\, j$$

ou sous forme matricielle :

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$$

où $\boldsymbol{x} = [x_1, \ldots, x_n]^T$, $\boldsymbol{s} = [s_1, \ldots, s_n]^T$ et $\boldsymbol{A}$ est une matrice contenant les coefficients a_{ij}. Le but de l'Analyse en Composantes Indépendantes est l'estimation des composantes originales s_i, ou de manière équivalente des coefficients a_{ij}. Par définition, les variables aléatoires s_i sont mutuellement indépendantes et la matrice de mélange est donc inversible. Ainsi le problème de l'ICA peut être formulé [58] :

$$\boldsymbol{u} = \boldsymbol{W}\boldsymbol{x} = \boldsymbol{W}\boldsymbol{A}\boldsymbol{s}$$

Plusieurs fonctions objectives ont été proposées, ainsi que des méthodes efficaces de résolution : InfoMax [36] ou FastICA [143]. Pour plus de détails sur la théorie et les applications possibles de l'Analyse en Composantes Indépendantes, voir [144]. Pour l'application de l'ICA à la reconnaissance de visages [30] et [88] proposent deux architectures. Dans la première, les images sont considérées comme une mélange linéaire d'images de base statistiquement indépendantes. Dans la seconde, le but est

de trouver des coefficients statistiquement indépendants représentant l'image d'entrée. Pour ces deux architectures, une Analyse en Composantes Principales est appliquée en prétraitement.

A.2.4 Analyse Discriminante Linéaire

Si les données d'apprentissage sont labélisées, ces informations peuvent être utlisées pour l'apprentissage du sous–espace. Ainsi, pour assurer une classification plus efficace, l'Analyse Discriminante Linéaire de Fisher (LDA pour *Linear Discriminant Analysis*) a pour but de maximiser la distance entre les classes tout en minimisant la variance intraclasse. Plus formellement, soient $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ n échantillons appartenant à une classe parmi c $\{X_1, \ldots, X_c\}$. L'Analyse Discriminante Linéaire calcule une fonction de classification $g(x) = \boldsymbol{W}^T x$, où la matrice $\boldsymbol{W}$ est choisie comme la projection linéaire minimisant la variance intra–classe

$$\boldsymbol{S}_B \;=\; \sum_{j=1}^{c} n_j (\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})^T$$

tandis que la variance inter–classe est maximisée

$$\boldsymbol{S}_W \;=\; \sum_{j=1}^{c} \sum_{x_k \in X_j} (\boldsymbol{x}_k - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_j)^T$$

où $\bar{\boldsymbol{x}}$ est le vecteur moyen de tous les échantillons, $\bar{\boldsymbol{x}}_j$ est le vecteur moyen des échatillons appartenant à la classe j, et n_j est le nombre d'échantillons de la classe j. Le calcul de la projection est ainsi obtenu en maximisant le critère de Fisher :

$$\boldsymbol{W}_{opt} \;=\; \arg\max_{W} \frac{|\boldsymbol{W}^T \boldsymbol{S}_B \boldsymbol{W}|}{|\boldsymbol{W}^T \boldsymbol{S}_W \boldsymbol{W}|}$$

La solution optimale à ce problème d'optimisation est donnée par la résolution du problème généralisé des valeurs propres

$$\boldsymbol{S}_B \boldsymbol{w} \;=\; \boldsymbol{\lambda} \boldsymbol{S}_W \boldsymbol{w}$$

ou en calculant directement les vecteurs propres de $\boldsymbol{S}_W^{-1} \boldsymbol{S}_B$.

Le rang de $\boldsymbol{S}_W^{-1} \boldsymbol{S}_B$ est au plus $c - 1$. Ainsi, pour de nombreuses applications, cette matrice est singulière et le problème des valeurs propres ne peut être résolu. Ce problème est souvent appelé le problème des échantillons de petite taille (*small sample size problem*). Pour surmonter ce problème, plusieurs solutions ont été proposées [165],[161],[309]. De plus, de nombreuses variantes de la LDA ont été introduites telles la classification robuste [103], ou la LDA incrémentale [283].

A.3 Méthodes non–linéaires de réduction de dimension

Nous décrivons ici dix méthodes non–linéaires de réduction de dimension. Les techniques non–linéaires peuvent être catégorisées en trois principaux types : les techniques essayant de préserver les propriétés globales des données d'apprentissage dans l'espace de faible dimension, les techniques s'attachant à préserver les propriétés locales des données d'apprentissage, et les techniques réalisant un alignement global de modèles linéaires.

A.3.1 Méthodes globales

Les méthodes globales de réduction non–linéaire de dimension essaient de préserver les propriétés globales des données d'apprentissage dans le nouvel espace de faible dimension. Sont présentées ici les techniques : MDS, Isomap, Kernel PCA, diffusion maps et les autoencoders multi–couches.

MDS La technique de MultiDimensional Scaling (MDS) [71], [166] représente un ensemble de techniques non–linéaires qui réalisent un mapping de la représentation des données dans l'espace de grande dimension, vers l'espace de faible dimension, tout en préservant autant que possible les distances pair–à–pair des échantillons. La qualité du mapping est exprmée à travers une fonction de stress, une mesure de l'erreur des distances entre les pairs dans les espaces de faible et de grande dimension. Deux exemples importants de fonctions de stress sont la fonction de stress brut et la fonction de coût de Sammon. La fonction de stress brut est définie par :

$$\phi(Y) \ = \ \sum_{i,j} \left(\|x_i - x_j\| - \|y_i - y_j\|\right)^2$$

où $\|x_i - x_j\|$ est la distance euclidienne entre les points x_i et x_j dans l'espace de grande dimension, et $\|y_i - y_j\|$ est la distance euclidienne entre les points y_i et y_j dans l'espace de faible dimension. La fonction de coût de Sammon est donnée par :

$$\phi(Y) \ = \ \frac{1}{\sum_{i,j} \|x_i - x_j\|} \sum_{i,j} \frac{\left(\|x_i - x_j\| - \|y_i - y_j\|\right)^2}{\|x_i - x_j\|}$$

La fonction de coût de Sammon diffère de la précédente en mettant davantage l'accent sur la préservation des distances originellement faibles. La minimisation de la fonction de stress peut être réalisée grâce à différentes méthodes telles la décomposition en éléments propres de la matrice des distances pair–à–pair, la méthode des gradients conjugués, ou encore la méthode dite pseudo–Newton [71]. Des variantes ont également été proposées telles SPE [25], SNE [134] ou la technique FastMap [102].

Isomap La technique MDS a été appliquée avec succès dans beaucoup d'applications, mais elle a l'inconvénient de reposer sur la distance euclidienne et de ne pas prendre en compte la distribution du voisinage des données d'apprentissage. Si les données dans l'espace de grande dimension repose sur une variété courbée (comme le *Swiss Roll* [274]), la technique MDS peut considérer que deux points sont proches alors qu'ils sont bien plus éloignés sur cette variété. La technique de l'Isomap [274] résout ce problème en essayant de préserver la distance géodésique entre des paires de points, la distance géodésique étant la distance mesurée entre deux points en suivant le contour de la variété.

Dans la technique de l'Isomap [274], la distance géodésique entre points x_i est calculée en construisant un graphe de voisinage G, dans lequel chaque noeud représente un point x_i et est connecté à ses k plus proches voisins x_{ij} de l'ensemble de points X. Le plus court chemin géodésique entre deux points représente une bonne estimée de la distance géodésique entre ces deux points et peut être calculée en utilisant l'algorithme de Dijkstra. Toutes les distances entre points peuvent ainsi être calculées pour former une matrice de distances géodésiques. La représentation de l'ensemble des points dans l'espace de faible dimension peut ensuite être calculée en appliquant la technique MDS (voir plus haut) sur cette matrice de distances géodésiques.

Une grosse faiblesse de la technique Isomap est son instabilité topologique [29]. Des connexions erronées peuvent être créées dans le graphe G. De tels « courts–circuits » [182] peuvent ainsi dégrader fortement les performances de l'Isomap.

Plusieurs approches ont été proposées pour pallier ces problèmes. Ainsi la suppression de points présentant de trop grandes variations dans le calcul de Dijkstra [65], ou la suppresssion des plus proches voisins qui violent la linéarité locale du graphe de voisinage [254] permettent dans une certaine mesure de corriger le défaut des « courts–circuits ».

D'autres défauts comme la présence de trous dans la variété [182] ou sa non convexité [275] peuvent mettre en défaut la technique. Cependant, celle–ci a été appliquée avec succès dans plusieurs applications comme la visualisation de données médicales [267].

Kernel PCA L'Analyse en Composantes Principales à Noyaux (ou KPCA pour *Kernel Principal Component Analysis*) est la reformulation non–linéaire de la technique linéaire classique qu'est l'Analyse en Composantes Principales en utilisant des fonctions à noyaux [255]. Depuis plusieurs années, la reformulation de techniques classiques à l'aide de « l'astuce du noyau » a permis l'émergence de nombreuses techniques comme les machines à support de vecteurs (ou SVM pour *Support Vector Machine*) [258]. L'ACP à noyaux calcule les principaux vecteurs propres de la matrice de noyaux plutôt que la matrice de covariance. Cette reformulation de l'ACP classique peut être vue comme une réalisation de l'ACP sur l'espace de

grande dimension transformée par la fonction noyau associée. L'ACP à noyaux permet ainsi de construire des mappings non–linéaires.

L'ACP à noyaux calcule d'abord la matrice de noyaux K des points x_i dont les entrées sont définis par :

$$k_{ij} \;=\; \kappa(x_i, x_j)$$

où κ est la fonction noyau [258]. Ensuite, la matrice de noyaux K est centrée :

$$k_{ij} \;=\; k_{ij} - \frac{1}{n}\sum_l k_{il} - \frac{1}{n}\sum_l k_{jl} + \frac{1}{n^2}\sum_{lm} k_{lm}$$

Cette opération correspond à la soustraction de la moyenne des vecteurs caractéristiques dans l'ACP linéaire classique.

Les d principaux vecteurs propres v_i de la matrice de noyaux centrée sont ensuite calculés. Il peut être montré que les vecteurs propres α_i de la matrice de covariance (dans l'espace de grande dimension) sont des versions mises à l'échelle des vecteurs propres v_i de la matrice de noyaux

$$\alpha_i \;=\; \frac{1}{\sqrt{\lambda_i}} v_i$$

La technique KPCA est une méthode basée sur les noyaux et ses performances dépendent alors grandement du choix de la fonction noyau κ. Les noyaux classiquement utilisés sont le noyau linéaire (cela revient alors à effectuer une ACP classique), le noyau polynomial ou encore le noyau gaussien [258].

L'Analyse en Composantes Principales à Noyaux a été appliquée avec succès à plusieurs problèmes comme la reconnaissance de la parole [189], ou la détection de nouveaux éléments d'un ensemble [137]. Un gros défaut de l'Analyse en Composantes Principales à noyau est que la taille de la matrice de noyaux est le carré du nombre d'échantillons de l'ensemble d'apprentissage ce qui peut rapidement être prohibitif. Une approche permettant de résoudre ce problème peut être trouvée dans [276].

Diffusion maps La technique des diffusion maps [172], [214] est basée sur la définition d'une marche aléatoire sur le graphe de données. En réalisant une marche aléatoire sur le graphe un certain nombre de pas, une mesure de la proximité des points peut être obtenue, pouvant définir ainsi une distance de diffusion. La technique des diffusion maps essaie de préserver autant que possible cette distance de diffusion dans l'espace de faible dimension.

Dans le cadre des diffusion maps, un graphe est construit à partir des données. Les poids des arêtes sont calculées grâce à un noyau gaussien. La matrice des poids résultante W peut ainsi être exprimée par

$$w_{ij} \;=\; e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

où σ représente la variance de la fonction gaussienne. La matrice est ensuite normalisée de sorte que la somme des lignes soit égale à 1. La matrice P résultante peut ainsi être définie par :

$$p_{ij}^{(1)} = \frac{w_{ij}}{\sum_k w_{ik}}$$

La matrice $P^{(1)}$ représente la probabilité de transition d'un point à un autre en un pas de temps. La matrice de probabilité pour t pas de temps est donnée par $\left(P^{(1)}\right)^{(t)}$. En utilisant les probabilités de marche aléatoires $p_{ij}^{(t)}$, la distance de diffusion peut être définie par :

$$D^{(t)}(x_i, x_j) = \sum_k \frac{\left(p_{ik}^{(t)} - p_{jk}^{(t)}\right)^2}{\psi^{(0)}(x_k)}$$

avec $\psi^{(0)}(x_i)$ un terme attribuant plus de poids aux parties du graphe plus dense. $\psi^{(0)}(x_i)$ est défini par $\psi^{(0)}(x_i) = \frac{m_i}{\sum_j m_j}$ où m_i est le degré du noeud x_i défini par $m_i = \sum_j p_{ij}$. A partir de l'équation plus haut, on peut voir que les paires de points possédant une grande probabilité de transition ont une faible distance de diffusion. L'idée sous–jacente à la distance de diffusion est qu'elle est basée sur plusieurs chemins du graphe, la rendant ainsi plus robuste au bruit. Dans l'espace de faible dimension, la technique de diffusion maps essaie de préserver cette distance de diffusion. Il a été montré [172] que la représentation Y dans l'espace de faible dimension qui préserve le mieux la distance de diffusion est formée par les vecteurs propres du problème

$$P^{(t)}Y = \lambda Y$$

Étant donné que le graphe est complètement connecté, la plus grande valeur propre est 1, et le vecteur propre associé est non pertinent et est donc rejeté. L'espace de faible dimension est ainsi défini par les d principaux vecteurs propres suivants. Les vecteurs propres retenus sont ensuite normalisés par leur valeur propre correspondante, d'où la représentation du nouvel espace défini par

$$Y = \{\lambda_2 v_2, \ldots, \lambda_{d+1} v_{d+1}\}$$

Autoencodeurs multicouches Les autoencodeurs sont des réseaux de neurones possédant un nombre impair de couches cachées [78], [135]. La couche cachée du milieu possède d neurones, tandis que l'entrée et la sortie en possèdent D (voir la figure A.3). Le réseau est entraîné pour minimiser l'erreur quadratique entre l'entrée et la sortie du réseau (idéalement elles sont égales). Une fois l'entraînement réalisé, une entrée x_i de dimension D peut être représentée par l'état des neurones de la couche cachée du milieu, dont la dimension est d. Il a été montré

[168] qu'un apprentissage d'un autoencodeur avec l'utilisation de fonctions d'activation linéaires conduisait à des résultats très similaires à ceux obtenus par une Analyse en Composantes Principales. Cependant, en utilisant des fonctions d'activation non–linéaires de type sigmoïde, l'autoencodeur est capable d'apprendre des mappings non–linéaires. Le gros défaut d'un autoencodeur est que si le nombre de neurones est important, le nombre de connexions (et donc de poids) l'est encore plus et l'apprentissage classique par rétopropagation du gradient peut s'avérer lent, voire même être bloqué dans un minimum local. Pour pallier ce problème, une solution [135] consiste à entraîner les couches séparemment à l'aide de Machines de Boltzmann Restreintes (RBMs pour *Restricted Boltzmann Machines*) [133]. Les RBM sont des types de réseaux de neurones où les états des neurones sont binaires et où les connexions entre neurones d'une même couche ne sont pas autorisées. Une fois que le préapprentissage du réseau est effectué pour chaque couche, un raffinement est effectué par rétropropagation du gradient pour l'ensemble du réseau. Une autre approche pour l'apprentissage d'un autoencodeur a été proposée dans [242] et est basée sur les algorithmes génétiques.

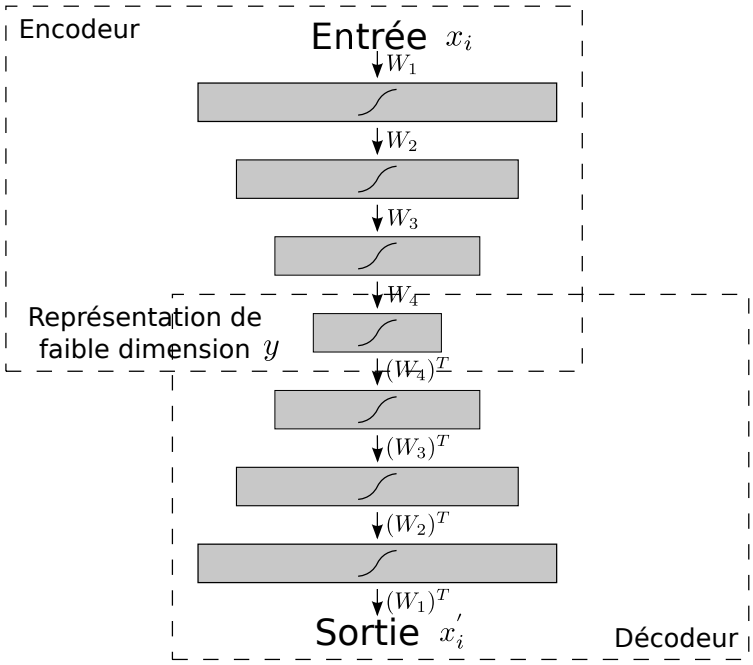


FIGURE A.3 – Modèle d'un autoencodeur multicouches.

A.3.2 Méthodes locales

Les méthodes dites locales de réduction de la dimension essaient de préserver les propriétés dans le voisinage des points. Ce type de technique repose sur la sup-

position qu'en préservant les propriétés locales des données, les propriétés globales de la variété le seront tout autant. La plupart de ces techniques peuvent se ramener à une définition valide dans le cadre de l'ACP à Noyaux à l'aide de noyaux locaux spécifiques [37], [125]. Sont présentés ici les méthodes LLE, Laplacian Eigenmaps, Hessian LLE et LTSA.

Locally Linear Embedding LLE (pour *Locally Linear Embedding*) [246] est une technique locale pour la réduction de dimension similaire à la technique de l'Isomap qui construit une représentation sous forme de graphe de voisinage des points. Cependant, LLE essaie seulement de garder les propriétés locales du graphe créé, rendant cette technique moins sensible aux problèmes de « courts–circuits ». De plus, la préservation des propriétés locales permet de caractériser des variétés non–convexes avec plus de succès que l'Isomap.

Les propriétés locales d'un point x_i sont décrites en considérant un point comme une combinaison linéaire W_i (poids de reconstruction) de ses k plus proches voisins x_{ij}. Ainsi, LLE adapte un hyperplan au point x_i et à ses voisins, supposant de fait un aspect localement linéaire. La supposition localement linéaire implique que les poids de reconstruction W_i du point x_i sont invariants par translation, rotation et mise à l'échelle.

Trouver la représentation Y dans l'espace de faible dimension d revient ensuite à minimiser la fonction de coût :

$$\phi(Y) \;=\; \sum_i \left(y_i - \sum_{j=1}^{k} w_i y_{ij} \right)^2$$

où y_i représente le point x_i dans l'espace de faible dimension.

Il peut être montré que les coordonnées des représentations y_i qui minimisent cette fonction de coût peuvent être trouvées en calculant les vecteurs propres des d plus petites valeurs propres de $(I - W)$ différentes de zéro (I étant la matrice identité de dimension n). En effet la fonction à minimiser peut s'écrire :

$$\phi(Y) \;=\; (Y - WY)^2 \;=\; Y^T (I - W)^T (I - W) Y$$

D'où le fait que les vecteurs propres de $(I - W)^T (I - W)$ correspondant aux plus petites valeurs propres forment la solution qui minimise $\phi(Y)$.

LLE a été appliquée avec succès au problème de super–résolution [60] ainsi qu'à la localisation de sources sonores [91]. Cependant, [267] montre que LLE échoue pour des tâches de visualisation de données biomédicales.

Laplacian Eigenmaps La technique des Laplacian Eigenmaps essaie de trouver une représentation des points dans l'espace de faible dimension en préservant les propriétés locales de la variété [35]. Les propriétés locales sont définies comme la distance entre proches voisins. Les distances entre un point et ses k plus proches voisins sont ensuite minimisées. Les distances sont ainsi pondérées selon la distance du point à ses voisins. Ainsi la distance au plus proche voisin a plus de poids pour le calcul de la fonction de coût que la distance au deuxième plus proche voisin. La minimisation de la fonction de coût est ensuite ramenée à un problème de valeurs propres. L'algorithme construit dans un premier temps un graphe de voisinage G dans lequel chaque point x_i est connecté à ses k plus proches voisins. Pour tous les points x_i et x_j qui sont connectés dans le graphe, le poids de l'arête est calculé grâce à la fonction à gaussiene $w_{ij} = e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}}$. Ces poids sont ensuite organisés sous forme matricielle W. Le calcul des représentations y_i dans l'espace de faible dimension s'effectue ensuite en minimisant la fonction de coût

$$\Phi(Y) = \sum_{ij} (y_i - y_j)^2 w_{ij}$$

Dans cette fonction de coût, un poids important w_{ij} correspond à de faibles distances entre x_i et x_j. Ainsi des points proches dans l'espace de départ sont proches dans l'espace de faible dimension. Le calcul de la matrice de degré M et de la matrice laplacienne L du graphe w permet de redéfinir le problème de minimisation comme un problème de valeurs propres. La matrice de degré M de W est une matrice diagonale dont les entrées sont les somme des lignes de W ($m_{ij} = \sum_j w_{ij}$). La matrice laplacienne du graphe L est donnée par $L = M - W$. Il peut être montré que

$$\Phi(Y) = \sum (y_i - y_j)^2 w_{ij} = 2Y^T LY$$

Ainsi minimiser $\Phi(Y)$ revient à minimiser $Y^T LY$, qui s'apparente à un problème généralisé aux valeurs propres

$$Lv = \lambda Mv$$

pour les d plus petites valeurs propres non nulles. Les d vecteurs propres correspondants forment ainsi l'espace de faible dimension recherché. La technique des Laplacian Eigenmaps a été appliquée avec succès au clustering [293] et à la reconnaissance faciale [129].

Hessian LLE La technique Hessian LLE [82] est une variante de LLE qui minimise la courbure de la variété dans l'espace de départ lors de la réduction de dimension. Cela est réalisé en analysant les valeurs propres de la matrice $\mathcal{H}$ décrivant la courbure de la variété autour des points. Cette courbure est mesurée en moyennant les courbures locales pour chaque point de la variété, grâce à des hyperplans

tangents en chaque point. Il a été montré [82] que les coordonnées dans l'esapce de faible dimension peuvent être trouvées en réalisant une analyse des éléments propres de $\mathcal{H}$. Pour chaque point x_i de la variété, l'hyperplan tangent est calculé en réalisant une ACP sur ses k plus proches voisins, en considérant la distance euclidienne pour leur détermination. Les d principaux vecteurs obtenus par l'ACP forment une base M ($M = \{m_1, \ldots, m_d\}$) permettant de décrire l'espace tangent à x_i (il faut donc que $k \geq d$). Un estimateur de la hessienne locale au point x_i est également calculé en considérant la matrice Z_i contenant en colonne tous les produits vectoriels de M. Une étape d'orthonormalisation de la matrice Z_i est ensuite réalisée. L'estimation de la matrice tangente Hessienne H_i est alors donnée par la transposée des dernières $\frac{d(d+1)}{2}$ colonnes de la matrice Z_i. La matrice $\mathcal{H}$ est ensuite construite à partir des estimées H_i :

$$\mathcal{H}_{lm} = \sum_i \sum_j ((H_i)_{jl} \times (H_i)_{jm})$$

La matrice $\mathcal{H}$ représente l'information de courbure de la variété dans l'espace de grande dimension. Une analyse des éléments propres de $\mathcal{H}$ est alors réalisée pour trouver la représentation dans l'espace de faible dimension qui minimise la courbure de la variété. Celle–ci est représentée par les d vecteurs propres dont les valeurs propres correspondantes sont les plus petites (et non nulles).

LTSA La technique appelée Local Tangent Space Analysis (LTSA) décrit les propriétés locales de l'espace de grande dimension grâce aux espaces tangents pour chaque point de la variété [306]. LTSA part du principe que si la variété est localement linéaire, alors il existe un mapping linéaire d'un point de l'espace de grande dimension vers son espace tangent local, ainsi qu'un mapping linéaire du point correspondant dans l'espace de faible dimension vers le même espace tangent local [306]. LTSA commence par calculer les bases des espaces tangents locaux Θ_i à chaque point x_i. Cette étape est réalisée par une ACP sur les k plus proches voisins du point x_i. Une propriété de l'espace tangent à un point est qu'il existe un mapping linéaire L_i des coordonnées θ_{ij} de cet espace tangent vers les représentations y_{ij} de l'espace de faible dimension. Ainsi la minimisation suivante est réalisée

$$\min_{Y_i, L_i} \sum_i \|Y_i J_k - L_i \Theta_i\|^2$$

où J_k est la matrice centrée de taille k [258]. Il peut être montré que la solution de cette minimisation est formée par les vecteurs propres d'une matrice d'alignement B correspondant aux d plus petites valeurs propres différentes de zéro de B. Les valeurs de la matrice d'alignement B sont obtenues en sommant itérativement (pour toutes les matrices V_i en initialisant $b_{ij} = 0, \forall ij$)

$$B_{N_i,N_i} = B_{N_i,N_i} + J_k(I - V_i V_i^T)J_k$$

où N_i est la matrice contenant les indices des plus proches voisins du point x_i. La représentation Y dans l'espace de faible dimension est ensuite obtenue en calculant les vecteurs propres correspondants aux d plus petites valeurs propres non nulles de la matrice symétrique $\frac{1}{2}(B + B^T)$.

Annexe B

Tables de connexions

Les tables de connexions se réfèrent aux connexions établies entre différentes couches dans un réseau de neurones convolutionnels.

Dans un réseau de neurones classiques, les couches sont complètement connectées entre elles, ce qui signifie que chaque neurone n_j d'une couche j a comme entrées tous les neurones n_i de la la couche i. Dans ce cas, le neurone n_j possède autant de poids qu'il y a de neurones sur la couche i, soit $Card(i)$. En pratique, cela conduit à une matrice de poids pour la couche j de dimension $Card(i) \times Card(j)$.

Pour une réseau de neurones convolutionnels, il est préférable de ne pas connecter tous les neurones d'une couche à sa couche précédente. En pratique, une couche de convolution de cardinalité N_c est suivie d'une couche de subsampling de cardinalité N_s, avec la contrainte $N_c = N_s$. Ainsi, chaque neurone de subsampling n'est connecté qu'à un seul neurone de convolution, c'est–à–dire que chaque neurone de subsampling n'a comme entrée le résultat que d'un seul neurone de convolution, on parle alors de connexion $1 - 1$.

Une table de connexion importante est celle reliant la première couche de subsampling S_2 à la seconde couche de convolution C_3. Classiquement, $Card(C_3) > Card(S_2)$, et chaque neurone de la couche C_3 a comme entrées les sorties de plusieurs neurones de la couche S_2. Par exemple, pour l'architecture LeNet5 [72] (Figure B.1) comportant 6 neurones de subsampling sur la couche S_2 et 16 sur la couche C_3, la table de connexion est présentée au tableau B.1. Une telle table de connexion a pour effet de casser la symétrie du réseau et force par ricochet, lors de la rétropropagation arrière, les neurones des couches C_1 à extraire des caractéristiques différentes.

Similairement à la table de connexion du réseau LeNet5, la table de connexion utilisée pour l'architecture de reconstruction de visages (voir Figure B.2) est présentée au tableau B.2.

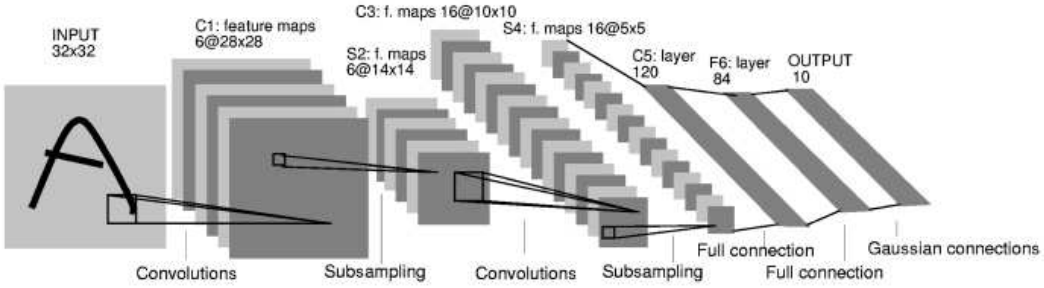


FIGURE B.1 – Architecture du réseau LeNet5 (tiré de [72]).

S_2 \ C_3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	x				x	x	x			x	x	x	x		x	x
2	x	x				x	x	x			x	x	x	x		x
3	x	x	x				x	x	x			x		x	x	x
4		x	x	x			x	x	x	x			x		x	x
5			x	x	x			x	x	x	x		x	x		x
6				x	x	x			x	x	x	x		x	x	x

TABLE B.1 – Table de connexions entre les couches S_2 et C_3 pour le réseau LeNet5 B.1. Chaque colonne indique quelles cartes de caractéristiques de S_2 sont combinées par les neurones de la couche C_3.

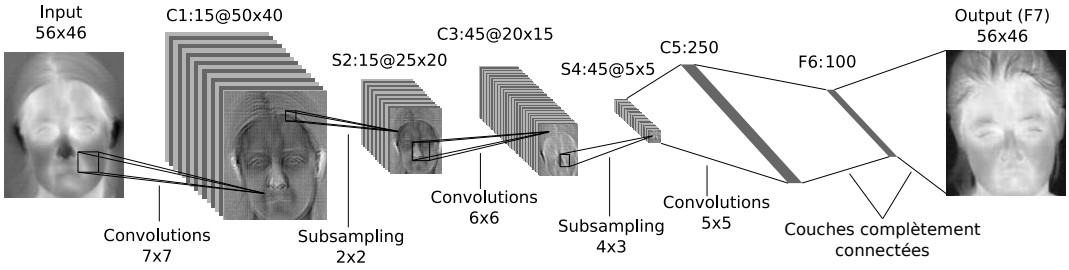


FIGURE B.2 – Architecture du réseau de reconstruction.

C_3 \ S_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	x	x	x												
2		x	x	x											
3			x	x	x										
4				x	x	x									
5					x	x	x								
6						x	x	x							
7							x	x	x						
8								x	x	x					
9									x	x	x				
10										x	x	x			
11											x	x	x		
12												x	x	x	
13													x	x	x
14	x													x	x
15	x	x													x
16	x	x	x	x											
17		x	x	x	x										
18			x	x	x	x									
19				x	x	x	x								
20					x	x	x	x							
21						x	x	x	x						
22							x	x	x	x					
23								x	x	x	x				
24									x	x	x	x			
25										x	x	x	x		
26											x	x	x	x	
27												x	x	x	x
28	x												x	x	x
29	x	x												x	x
30	x	x	x												x
31	x	x		x	x										
32		x	x		x	x									
33			x	x		x	x								
34				x	x		x	x							
35					x	x		x	x						
36						x	x		x	x					
37							x	x		x	x				
38								x	x		x	x			
39									x	x		x	x		
40										x	x		x	x	
41											x	x		x	x
42	x											x	x		x
43	x	x											x	x	
44		x	x											x	x
45	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

TABLE B.2 – Table de connexions entre les couches S_2 et C_3 pour le réseau de reconstruction B.2. Chaque ligne indique quelles cartes de caractéristiques de S_2 sont combinées par les neurones de la couche C_3.

Annexe C

Principales bases de données de visages

Les bases de données de visages peuvent être classées en trois catégories selon l'objectif recherché : reconnaissance, détection de visages ou analyse des expressions faciales.

C.1 Bases de données pour la reconnaissance faciale

De nombreuses bases de données ont été créées pour répondre aux besoins spécifiques du problème de la reconnaissance faciale. Nous détaillons ici les principales bases de données créées pour la problématique de la reconnaissance faciale.

AR Database La base de données AR [12] (Figure C.1) présente 116 personnes (63 hommes et 53 femmes) pour un total de 3288 images de taille 768×576. La capture a consisté en deux sessions séparées de deux semaines. La base de données présente des variations d'expressions faciales, d'illumination et d'occlusion, ces variations étant soigneusement contrôlées.

BANCA Database La base de données BANCA [13] (Figure C.2) présente les acquisitions de 52 personnes (26 hommes et 26 femmes). Les captures ont consisté en 12 sessions sur une période de 3 mois. Les acquisitions ont eu lieu avec différentes caméras (hausse et basse résolution) ainsi qu'avec un microphone. Trois scénarios ont ainsi été étudiés : environnement contrôlé, dégradé et adverse. La base de donnée possède également un protocole bien défini de vérification avec des listes de vrais clients et des listes d'imposteurs.

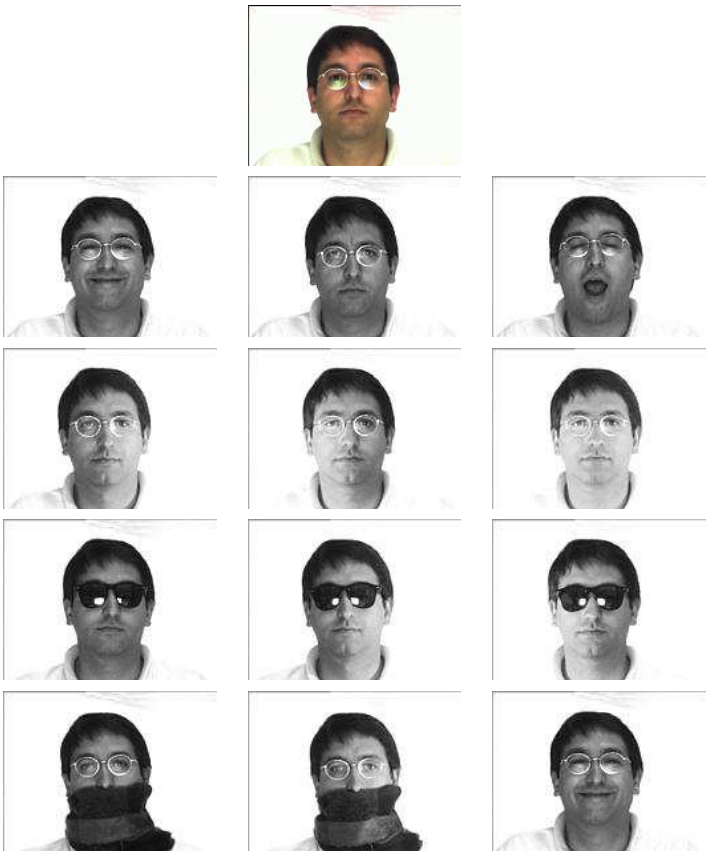


FIGURE C.1 – Échantillons de la base AR.

Equinox Database La base de données Equinox [15] (Figure C.3) présente 91 personnes capturées à l'aide de caméras visible et infrarouge à grande longueurs d'onde (8–12 μm). Les sessions ont consisté en la prise d'une séquence vidéo de 4 secondes pendant lesquelles les personnes prononçaient des voyelles. La base présente des expressions faciales (sourire, froncement de sourcils et surprise), ainsi que trois conditions d'illumination : frontal, latéral gauche et latéral droit. De plus, les personnes portant des lunettes ont également été prises sans.

FERET La base de données FERET [10] (Figure C.4) présente 1199 personnes pour un total de plus de 14000 images couleur (ou non) de taille 512×768 (dans la nouvelle version de la base). La base présente des variations d'expressions faciales, de pose, d'illumination ainsi qu'un certain délai entre différentes captures d'une même personne. Le protocole bien défini ainsi que le très grand nombre d'images disponibles en ont fait une des bases de données les plus populaires en reconnaissance de visages.

FIGURE C.2 – Échantillons de la base BANCA.

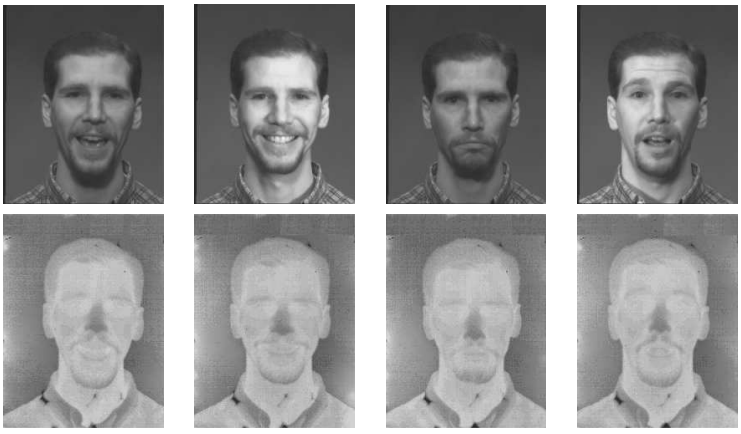


FIGURE C.3 – Échantillons de la base Equinox.

PIE Database La base de données PIE (Pose, Illumination and Expression) [14] (Figure C.5) est composée de 68 personnes pour un total de 41368 images de taille 640×486. La base présente de grandes variations de pose, d'illumination ainsi que des expressions faciales.

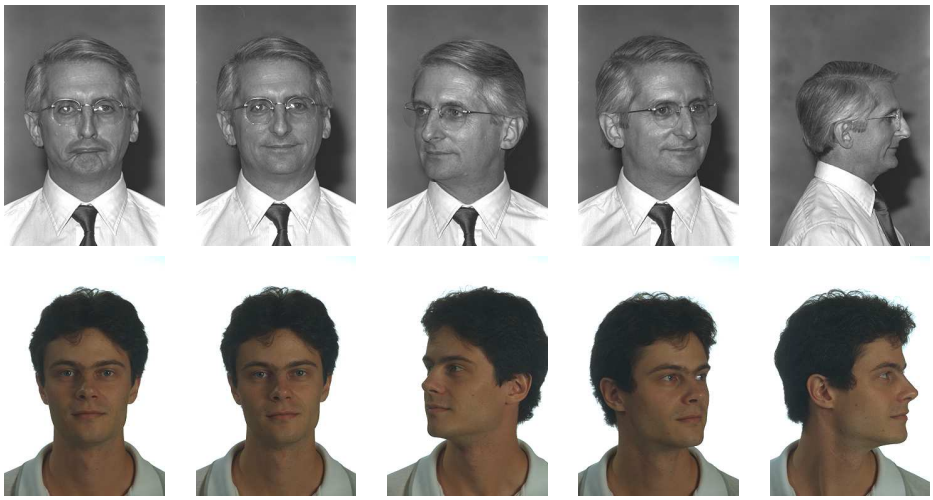


FIGURE C.4 – Échantillons de la base Feret.

FIGURE C.5 – Échantillons de la base PIE.

YaleB La base de données YaleB [16] (Figure C.6) est composée de 10 personnes prises sous 9 poses et 64 illuminations différentes. Pour chaque pose, une capture avec l'illumination ambiante est également présente, ce qui fait un total de 5850 images de taille 640×480.

ATT La base de données AT&T (anciennement ORL) [9] (Figure C.7) est composée de 40 personnes pour un total de 400 images de taille 92×110. Les visages présentent des variations d'illumination, de pose et d'artefacts (lunettes, barbes, ...).

LFW La base de données « Labeled Faces in the Wild » [17] (Figure C.8) est composée de plus de 13000 images provenant de news du web. 1680 personnes présentent au moins 2 images différentes. La seule contrainte pour la réalisation de la base est la détection des visages par le détecteur de Viola et Jones. Une version de

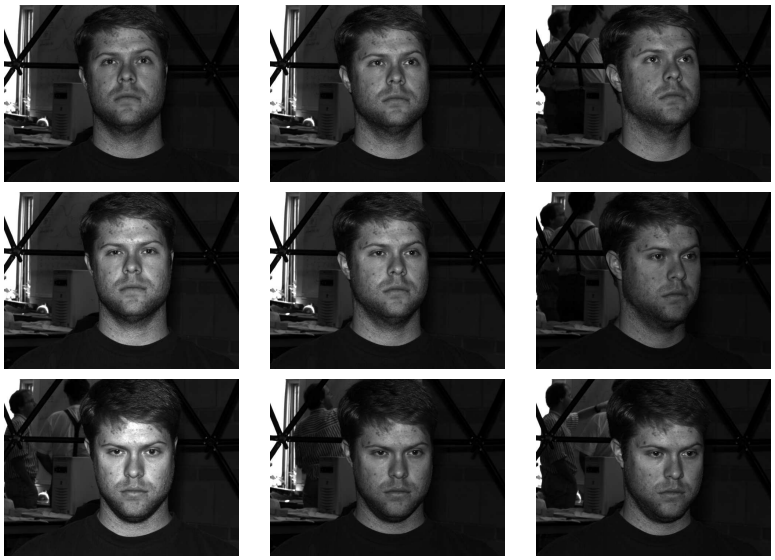


FIGURE C.6 – Échantillons de la base YaleB pour une luminosité d'angle et d'azimut de 0.

FIGURE C.7 – Échantillons de la base AT&T.

cette base existe où tous les visages sont recadrés géométriquement. Un protocole d'authentification est disponible avec la base afin de permettre la comparaison de

résultats.

FIGURE C.8 – Échantillons de la base Labeled Faces in the Wild.

Notre–Dame Database La base de données de l'université de Notre–Dame [11] (Figure C.9) est une collection d'ensembles d'images 2D visibles, infrarouges ou d'images 3D de visages. Elle comporte également des données relatives aux iris ou aux oreilles. Elle contient notamment le sous–ensemble $X1$ présentant des images visibles et infrarouges (grandes longueurs d'ondes) prises au même instant.

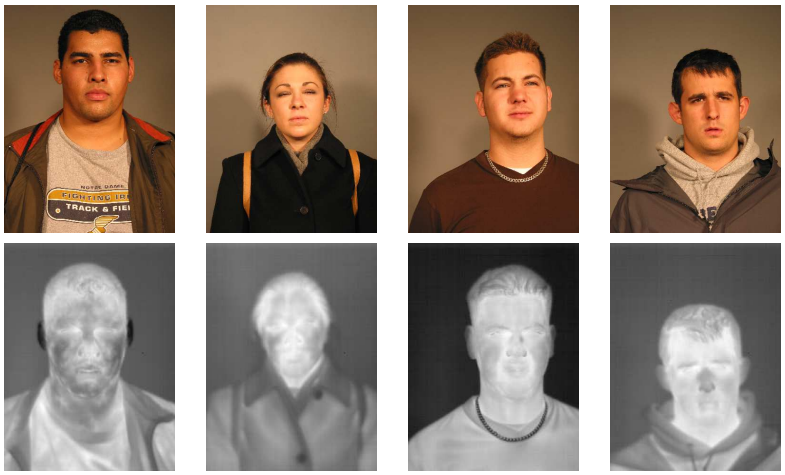


FIGURE C.9 – Échantillons de la base Notre–Dame (Collection $X1$).

Plastic Surgery Face Database La base de données « Plastic Surgery » [21] (Figure C.10) est composée de 900 personnes ayant subi une opération de chirurgie plastique. Pour chacune d'entre elles, une image avant et après l'opération est disponible. Les différentes chirurgies appliquées sont nombreuses comme la chirurgie du nez, l'effacement des rides (chirurgies locales) ou encore le lifting général (chirurgie globale).

FIGURE C.10 – Échantillons de la base Plastic Surgery.

C.2 Bases de données pour la détection de visages

Les algorithmes de détection de visage ont souvent besoin d'être entraînés sur des images de visages ainsi que sur des images ne contenant pas de visages afin de construire une représentation du visage humain. Bien que les bases de données citées plus haut soient souvent utilisées, des bases de données spécifiques ont été créées pour répondre aux problèmes tels que la multidétection, ou la détection dans des environnements non contrôlés. Nous détaillons ici les principales bases de données créées pour la problématique de la détection de visages.

Combined MIT/CMU L'ensemble d'images appelé *Combined MIT/CMU* [18] (Figure C.11) inclut 180 images organisées en deux sous–ensembles. Le premier contient 130 images contenant 507 visages frontaux. Les images proviennent de nombreuses sources : Internet, journaux et magazines, scans, ... Le second ensemble est composé de 50 images contenant 223 visages et a été créé pour tester les algorithmes sur des images de visages ayant subi une rotation dans le plan.

CMU Test Set II Cet ensemble d'images [18] (Figure C.12) a été collecté pour tester les algorithmes de détection sur des visages ayant subi une rotation hors plan. Il est composé de 208 images contenant 441 visages dont 347 sont de profil. Les images proviennent d'Internet.

BioID La base de données BioID [19] (Figure C.13) contient 1521 images de visages de 23 personnes différentes. Les images sont de taille 384×288 et ont été capturées à des endroits différents conférant ainsi aux visages de grandes variations de luminosité, d'expression ou encore de taille. Les emplacements des yeux sont disponibles pour tous les visages.

FIGURE C.11 – Échantillons de la base MIT/CMU.

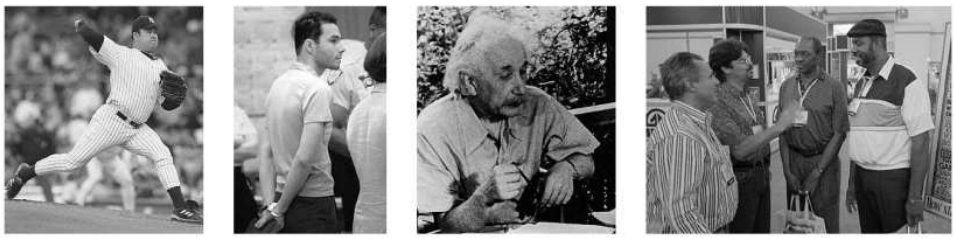


FIGURE C.12 – Échantillons de la base CMU Test Set II.

FIGURE C.13 – Échantillons de la base BioID.

Un grand nombre d'algorithmes de détection de visages construisent la représentation d'un visage à partir d'une base de données. Ainsi s'il est relativement facile de se procurer des données de visages labélisées pour créer une représentation d'un visage, caractériser la classe des non–visages est beaucoup plus difficile, voire impossible. Une base de données populaire contenant des images de non–visages (autres que des images du Web) a été créée par l'université de Washington [20].

C.3 Bases de données pour la reconnaissance d'expressions faciales

La reconnaissance automatique d'expressions faciales est un domaine pour lequel beaucoup de travaux sont réalisés. La constitution de bases de données dédiées diffère cependant selon le type d'expressions faciales supportées. Ainsi un premier groupe de bases de données se concentre sur ce qui est désigné par Ekman et Priesen [94] comme les six émotions de base (joie, tristesse, peur, dégoût, surprise, colère). Un second groupe essaie d'extraire plus finement les caractéristiques de l'expression du visage, où 44 unités d'action sont répertoriées (appelé FACS pour *Facial Action Coding System*) reposant chacune sur un ensemble spécifique de muscle du visage. La constitution de bases de données de reconnaissance d'expressions faciales n'est pas aisée étant donnée que la plupart du temps, les personnes doivent exécuter une expression demandée, et qu'ainsi elle peut être forcée, et n'est donc pas forcément très naturelle. Notons cependant la base de données vidéo de l'université du Texas [224] dans laquelle de nombreuses émotions sont spontanées.

Japanese Female Facial Expression (JAFFE) Database La base de données JAFFE [24] (Figure C.14) est composée de 213 images de femmes japonaises. Les personnes ont dû mimer les expressions demandées. Les expressions des images ont ensuite été notées par 60 autres femmes japonaises sur une échelle de 1 à 5.

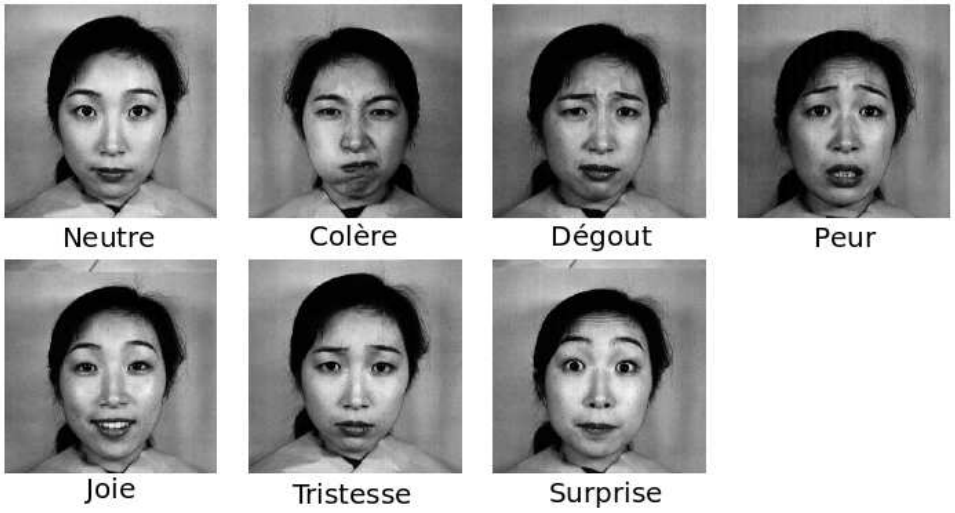


FIGURE C.14 – Échantillons de la base JAFFE.

Bases de données de l'université du Maryland La base de données de l'université du Maryland [22] (Figure C.15) est composée de séquences d'images de 40 personnes d'origines et de cultures différentes. La base contient 70 séquences pour un total de 145 expressions.

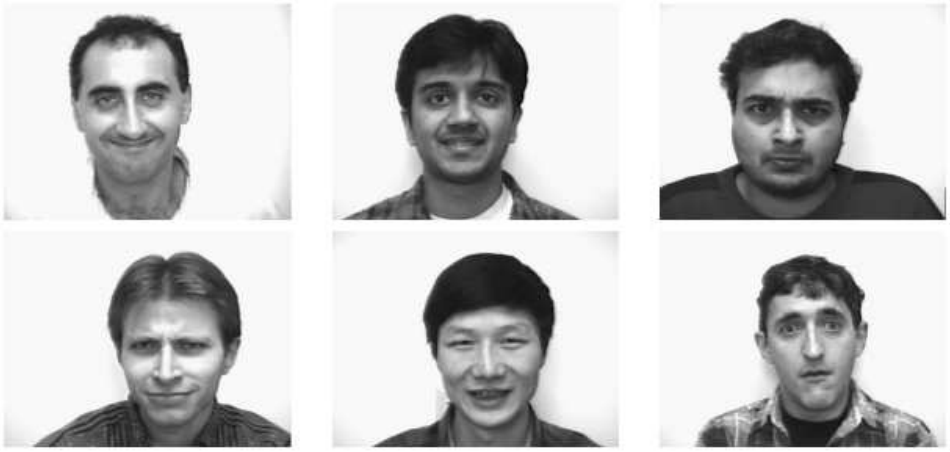


FIGURE C.15 – Échantillons de la base de données de l'Université du Maryland.

Cohn–Kanade AU–Coded Facial Expression Database La base de données de Cohn–Kanade (Figure C.16) est disponible depuis l'université de Carnegie Mellon [23]. Elle est composée de 100 personnes pour 23 expressions faciales différentes, codées à l'aide du système FACS (pour « Facial Action Coding System »). Notons que la base de données présente en plus des expressions faciales des variations de luminosité ou encore des rotations du visage.

FIGURE C.16 – Échantillons de la base de Cohn–Kanade.

Publications et Séminaires

Conférences Internationales avec actes

[1] P. Buyssens and M. Revenu. Fusion levels of visible and infrared modalities for face identification. In *IEEE International Conference on Biometrics : Theory, Applications and Systems (BTAS)*, Washington, September 2010.

[2] P. Buyssens and M. Revenu. IR and visible identification via sparse representation. In *IEEE International Conference on Biometrics : Theory, Applications and Systems (BTAS)*, Washington, September 2010.

[3] P. Buyssens and M. Revenu. Learning sparse face features : Application to face verification. In *International Conference on Pattern Recognition (ICPR)*, Istanbul, August 2010.

[4] P. Buyssens, M. Revenu, and O. Lepetit. Fusion of IR and visible light modalities for face recognition. In *IEEE International Conference on Biometrics : Theory, Applications and Systems (BTAS)*, Washington, September 2009.

Conférences Nationales avec actes

[5] P. Buyssens and M. Revenu. Fusion des modalités visible et infrarouge pour la reconnaissance faciale. In *Reconnaissance de Formes et Intelligence Artificielle (RFIA)*, Caen, 2010.

[6] P. Buyssens, M. Revenu, and O. Lepetit. Réseau de neurones convolutionnels pour la reconnaissance faciale infrarouge. In *Colloque GRETSI*, Dijon, 2009.

Séminaires

[7] P. Buyssens. Utilisation de réseaux de neurones convolutionels pour la reconnaissance faciale multimodale. Télécom & Management SudParis, Evry (ex INT), November 2008.

[8] P. Buyssens. Utilisation de réseaux de neurones convolutionnels pour la reconnaissance faciale visible/infrarouge. Séminaire $15 + 15$, Orange Labs, April 2009.

Bibliographie

[9] http ://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[10] http ://www.itl.nist.gov/iad/humanid/feret/.

[11] http ://www.nd.edu/ cvrl/Data_Sets.html.

[12] http ://www2.ece.ohio-state.edu/ aleix/ARdatabase.html.

[13] http ://www.ee.surrey.ac.uk/CVSSP/banca/.

[14] http ://www.ri.cmu.edu/research_project_detail.htmlproject_id=418&menu_id=261.

[15] http ://www.equinoxsensors.com/products/HID.html.

[16] http ://cvc.yale.edu/projects/yalefacesB/yalefacesB.html.

[17] vis-www.cs.umass.edu/lfw/.

[18] http ://www.ri.cmu.edu/projects/project_419.html.

[19] http ://www.humanscan.de/support/downloads/facedb.php.

[20] http ://www.wuarchive.wustl.edu/ aminet/pix/.

[21] http ://iiitd.edu.in/iab/Image_Analysis_and_Biometrics_Group/Resources.html.

[22] http ://www.umiacs.umd.edu/users/yaser/DATA/index.html.

[23] http ://vasc.ri.cmu.edu/idb/html/face/facial_expression/index.html.

[24] http ://www.kasrl.org/jaffe.html.

[25] D. K. Agrafiotis. Stochastic proximity embedding. *Journal of Computational Chemistry*, 24(10) :1215–1221, 2003.

[26] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns : Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12) :2037–2041, December 2006.

[27] B. Ans, J. Hérault, and C. Jutten. Adaptive neural architectures : detection of primitives. In *COGNITIVA*, pages pages 593–597, 1985.

[28] A. Antoniadis and J. Fan. Regularization of wavelet approximations. *Journal of the American Statistical Association*, 96(455) :939–963, September 2001.

[29] M. Balasubramanian and E. L. Schwartz. The isomap algorithm and topological stability. *Science*, 295(5552) :7, January 2002.

[30] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski. Face recognition by independent component analysis. *Transactions on Neural Networks*, August 15 2002.

[31] R. Battiti. Accelerated backpropagation learning : Two optimization methods. *Complex Systems*, 3 :331–342, 1989.

[32] G. Baudat and Fatiha Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10) :2385–2404, 2000.

[33] G. N. Bebis, A. Gyaourova, S. Singh, and I. T. Pavlidis. Face recognition by fusing thermal infrared and visible imagery. *Image and Vision Computing*, 24(7) :727–742, July 2006.

[34] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *Journal on Imaging Sciences*, 2(1) :183–202, 2009.

[35] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Neural Information Processing Systems*, pages 585–591. MIT Press, 2001.

[36] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7 :1129–1159, 1995.

[37] Y. Bengio, O. Delalleau, N. Le Roux, J. F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16(10) :2197–2219, 2004.

[38] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In Bernhard Schölkopf, John C. Platt, and Thomas Hoffman, editors, *Neural Information Processing Systems*, pages 153–160. MIT Press, 2006.

[39] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards ai. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large-Scale Kernel Machines*. MIT Press, 2007.

[40] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[41] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[42] S. Biswas, K. W. Bowyer, and Patrick J. Flynn. Multidimensional scaling for matching low–resolution facial images. In *Biometrics : Theory, Applications and Systems*, 2010.

[43] A. Björck. *Numerical Methods for Least Squares Problems*. SIAM Publications, 1996.

[44] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision*, page I : 707 ff., 2002.

[45] M. Brand. From subspaces to submanifolds. Technical report, Mitsubishi Electric Research Laboratories, September 2004.

[46] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1) :34–81, 2009.

[47] R. Brunelli and T. Poggio. Face recognition : Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10) :1042–1052, 1993.

[48] I. Buciu. Non-negative matrix factorization, A new tool for feature extraction : Theory and applications. *International Journal of Computers, Communications & Control*, III(S.) :67–74, May 2008.

[49] I. Buciu and I. Pitas. Application of non-negative and local non negative matrix factorization to facial expression recognition. In *International Conference on Pattern Recognition*, pages I : 288–291, 2004.

[50] P. Buddharaju, I. T. Pavlidis, and I. A. Kakadiaris. Face recognition in the thermal infrared spectrum. In *OTCBVS*, page 133, 2004.

[51] P. Buyssens and M. Revenu. Fusion levels of visible and infrared modalities for face identification. In *Biometrics : Theory, Applications and Systems*, Washington, September 2010.

[52] P. Buyssens, M. Revenu, and O. Lepetit. Fusion of IR and visible light modalities for face recognition. In *Biometrics : Theory, Applications and Systems*, Washington, September 2009.

[53] E. J. Candès, L. Demanet, D. L. Donoho, and L. Ying. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5(3) :861–899, January 2006.

[54] E. J. Candès and D. L. Donoho. Curvelets and curvilinear integrals. *Journal of Approximation Theory*, 113(1) :59–90, 2001.

[55] E. J. Candès. Ridgelets and the representation of mutilated Sobolev functions. *SIAM Journal on Mathematical Analysis*, 33(2) :347–368, 2001.

[56] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51, 2005.

[57] R. Cappelli, D. Maio, and D. Maltoni. Combining fingerprint classifiers. In *Multiple Classifier Systems*, pages 351–361, 2000.

[58] J. F. Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters*, March 23 1999.

[59] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana. Discriminative common vectors for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1) :4–13, 2005.

[60] H. Chang, D. Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 275–282, 2004.

[61] K. Chang and J. Ghosh. Principal curve classifier – A nonlinear approach to pattern classification. In *IEEE International Conference on Neural Networks*, volume I, pages I–695–I–700, Anchorage, AK, July 1998. IEEE. UT Austin.

[62] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. Technical report, Department of Statistics, Stanford University, February 1996.

[63] X. Chen, P. J. Flynn, and K. W. Bowyer. IR and visible light face recognition. *Computer Vision and Image Understanding*, 99(3) :332–358, September 2005.

[64] Y. Chen, S. Dass, and A. K. Jain. Fingerprint quality indices for predicting authentication performance. In *Proceedings of Fifth International Conference on Audio and Video-Based Biometric Person Authentication (AVBPA)*, 2005.

[65] H. Choi and S. Choi. Robust kernel isomap. *Pattern Recognition*, 40(3) :853–862, March 2007.

[66] A. Cohen, R. Devore, P. Petrushev, and H. Xu. Nonlinear approximation and the space BV(R^2). *AMER. J. MATH*, October 12 1998.

[67] P. L. Combettes and J. C. Pesquet. Proximal thresholding algorithm for minimization over orthonormal bases. *SIAM Journal on Optimization*, 18(4) :1351–1376, 2008.

[68] P. Comon. Independent component analysis — a new concept ? *Signal Processing*, 36(3) :287–314, 1994.

[69] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6) :681–685, June 2001.

[70] T. F. Cootes and C. J. Taylor. Constrained active appearance models. In *International Conference on Computer Vision*, pages 748–754, 2001.

[71] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1994.

[72] Y. Le Cun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 86(11) :2278–2324, November 1998.

[73] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, November 02 2003.

[74] G. Davis, S. Mallat, and Z. Zhang. Adaptive time-frequency decompositions. *Optical Engineering*, 33(7) :2183–2191, July 1994.

[75] K. Delac and M. Grgic. A survey of biometric recognition methods. *46th International Symposium Electronics in Marine*, 2004.

[76] D. Petrovska Delacretaz, G. Chollet, and B. Dorizzi. *Guide to Biometric Reference Systems and Performance Evaluation*. Springer, 2009.

[77] L. Demanet and L. Ying. Wave atoms and sparsity of oscillatory patterns. *Applied and Computational Harmonic Analysis*, 2006.

[78] D. DeMers and G. Cottrell. Non-linear dimensionality reduction. In *Neural Information Processing Systems*, pages 580–587, 1993.

[79] K. I. Diamantaras and S. Y. Kung. *Principal Component Neural Networks : Theory and Applications*. Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control. Wiley, New York, 1996.

[80] M. N. Do and M. Vetterli. The contourlet transform : An efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 14(12) :2091–2106, December 2005.

[81] D. L. Donoho and M. Elad. Maximal sparsity representation via l^1 minimization. *Proceedings of National Academy of Sciences*, November 25 2003.

[82] D. L. Donoho and C. Grimes. Hessian eigenmaps : New locally linear embedding techniques for high-dimensional data. Technical report, July 02 2003.

[83] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7) :2845–2862, 2001.

[84] D. L. Donoho, I. M. Johnston, J. C. Hoch, and A. S. Stern. Maximum entropy and the nearly black object. *Journal of the Royal Statistical Society. Series B*, 54 :41–81, 1992.

[85] D. L. Donoho and B. F. Logan. Signal recovery and the large sieve. *SIAM Journal on Applied Mathematics*, 52(2) :577–591, April 1992.

[86] D. L. Donoho and P. B. Stark. Uncertainty principles and signal recovery. *SIAM Journal of Applied Mathematics.*, 49(3) :906–931, June 1989.

[87] J. Doublet. Étude de nouveaux caractères biométriques de la main dans un contexte télécom. Thèse de doctorat, Université de Caen, December 2009.

[88] B. A. Draper, K. Baek, M. S. Bartlett, and J. R. Beveridge. Recognizing faces with PCA and ICA. *Computer Vision and Image Understanding*, 91(1–2) :115–137, July/August 2003.

[89] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[90] S. Duffner and C. Garcia. Face recognition using non-linear image reconstruction. In *i-LIDS : Bag and Vehicle Detection Challenge*, pages 459–464, 2007.

[91] R. Duraiswami and V. C. Raykar. The manifolds of spatial hearing. In *Internation Conference on Acoustics, Speech and Signal Processing*, pages 285–288, 2005.

[92] Hossein Ebrahimpour-Komleh. Fractal techniques for face recognition, 2006.

[93] G. J. Edwards and C. J. Taylorand T. F. Cootes. Interpreting face images using active appearance models. In *Automatic Face and Gesture Recognition*, pages 300–305, 1998.

[94] P. Ekman and W. V. Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2) :124–129, 1971.

[95] K. Engan, S. O. Aase, and J. H. Husøy. Frame based signal compression using method of optimal directions (MOD). In *International Symposium on Circuits and Systems*, pages IV–1–IV–4, Orlando, USA, June 1999.

[96] K. Engan, S. O. Aase, and J. H. Husøy. Multi-frame compression : Theory and design. *Signal Processing*, 80(10) :2121–2140, October 2000.

[97] K. Engan, B. Rao, and K. Kreutz-Delgado. Frame design using FOCUSS with method of optimized directions (MOD). In *Nordic Signal Processing Symposium*, pages 65–69, Oslo, Norway, September 1999.

[98] Meng Joo Er, Shiqian Wu, Juwei Lu, and Hock Lye Toh. Face recognition with radial basis function (RBF) neural networks. *IEEE-EC*, 13 :697–710, May 2002.

[99] M. J. Fadili. Une exploration des problèmes inverses par les représentations parcimonieuses et l'optimisation non lisse. Thèse d'Habilitation à Diriger des Recherches, Université de Caen, March 2010.

[100] M. J. Fadili and E. T. Bullmore. Penalized partially linear models using orthonormal wavelet bases with an application to fMRI time series. In *International Symposium on Biomedical Imaging*, pages 1171–1174. IEEE, 2004.

[101] M. J. Fadili and J. L. Starck. Monotone operator splitting for optimization problems in sparse recovery. In *International Conference on Image Processing*, pages 1461–1464. IEEE, 2009.

[102] C. Faloutsos and K. I. Lin. FastMap : a fast algorithm for indexing, datamining and visualization of traditional and multimedia datasets. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 24(2) :163–174, June 1995.

[103] S. Fidler, D. Skocaj, and A. Leonardis. Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3) :337–350, March 2006.

[104] D. J. Field. Scale-invariance and self-similar 'wavelet' transforms : An analysis of natural scenes and mammalian visual systems. In M. Farge, J. C. R. Hunt, and J. C. Vassilicos, editors, *Wavelets, Fractals, and Fourier Transforms*, pages 151–194. Clarendon Press, 1993.

[105] D. J. Field. What is the goal of sensory coding ? *Neural Computation*, 6(4) :559–601, 1994.

[106] M. A. T. Figueiredo and R. D. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12(8) :906–916, August 2003.

[107] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 :179–188, 1936.

[108] M. K. Fleming and G. W. Cottrell. Categorization of faces using unsupervised feature extraction. In *IEEE International Joint Conference on Neural Networks*, volume II, pages II–65–II–70, San Diego, 1990. IEEE. UCSD.

[109] K. Fukunaga. *Introduction to Statistical Pattern Recognition, Second Edition*. Academic Press, Boston, MA, 1990.

[110] K. Fukushima. Neocognitron : A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36 :193–202, 1980.

[111] K. Fukushima. A neural network model for selective attention in visual pattern recognition. *Biological Cybernetics*, 55 :5–15, 1986.

[112] K. Fukushima. Analysis of the process of visual pattern recognition by the neocognitron. *Neural Networks*, 2(6) :413–420, 1989.

[113] K. Fukushima and T. Imagawa. Recognition and segmentation of connected characters with selective attention. *Neural Networks*, 6(1) :33–41, 1993.

[114] Y. Gao and M. K. H. Leung. Face recognition using line edge map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6) :764–779, 2002.

[115] C. Garcia and M. Delakis. A neural architecture for fast and robust face detection. In *International Conference on Pattern Recognition*, pages II : 44–47, 2002.

[116] C. Garcia, G. Zikos, and G. Tziritas. Wavelet packet analysis for face recognition. *Image and Vision Computing*, 18(4) :289–297, March 2000.

[117] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many : Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell*, 23(6) :643–660, 2001.

[118] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer, 1992.

[119] R. Gross, I. Matthews, and S. Baker. Eigen light-fields and face recognition across pose. In *FG*, pages 1–7, 2002.

[120] R. Gross, J. Shi, and J. F. Cohn. Quo vidas face recognition ? In *Workshop on Empirical Evaluation Methods in Computer Vision*, pages xx–yy, 2001.

[121] G. D. Guo, S. Z. Li, and K. L. Chan. Face recognition by support vector machines. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 196–201, 2000.

[122] K. Guo and D. Labate. Optimally sparse multidimensional representation using shearlets. *SIAM Journal on Mathematical Analysis*, 39(1) :298–318, 2007.

[123] G. Hagen, T. Smith, A. Banasuk, R.R. Coifman, and I. Mezic. Validation of low-dimensional models using diffusion maps and harmonic averaging. In *IEEE Conference on Decision and Control*, 2007.

[124] P. M. Hall, D. R. Marshall, and R. R. Martin. Incremental eigenanalysis for classification. In *British Machine Vision Conference*, 1998.

[125] J. Ham, D. Lee, S. Mika, and B. Schölkopf. Kernel view of the dimensionality reduction of manifolds. *International Conference on Machine Learning*, 2004.

[126] Hao, Anderson, and Daugman. Combining crypto with biometrics efficiently. *IEEE Transactions on Computers*, 55, 2006.

[127] S. Harmeling. *Independent component analysis and beyond.* PhD thesis, Universität Potsdam ; Mathematisch-Naturwissenschaftliche Fakultät. Institut für Informatik, 2004.

[128] X. He and P. Niyogi. Locality preserving projections. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *Neural Information Processing Systems*. MIT Press, 2003.

[129] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3) :328–340, 2005.

[130] D. O. Hebb. *The Organization of Behavior : A Neuropsychological Theory*. Wiley, New York, new edition edition, June 1949.

[131] M. Heiler and C. Schnorr. Learning non-negative sparse image codes by convex programming. In *International Conference on Computer Vision*, pages II : 1667–1674, 2005.

[132] B. Heisele, P. Ho, J. Wu, and T. Poggio. Face recognition : component-based versus global approaches. *Computer Vision and Image Understanding*, 91(1–2) :6–21, July/August 2003.

[133] G. E. Hinton, S. Osindero, and Y. Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7) :1527–1554, 2006.

[134] G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Neural Information Processing Systems*, pages 833–840. MIT Press, 2002.

[135] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313, 2006.

[136] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1) :66–75, January 1994.

[137] H. Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40(3) :863–874, March 2007.

[138] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24 :417–441, 498–520, 1933.

[139] K. Hotta. View independent face recognition based on kernel principal component analysis of local parts. In *International Conference on Image Processing*, pages III : 760–763, 2005.

[140] N. Houmani, S. Garcia-Salicetti, and B. Dorizzi. On assessing the robustness of pen coordinates, pen pressure and pen inclination to time variability with personal entropy. In *Biometrics : Theory, Applications and Systems*, 2009.

[141] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5 :1457–1469, November 2004.

[142] D. H. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Physiol*, 160 :106–154, 1962.

[143] A. Hyväerinen. The fixed-point algorithm and maximum likelihood estimation for independent component analysis. *Neural Processing Letters*, 10(1) :1–5, 1999.

[144] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*, pages 1–12. John Wiley & Sons, 2001.

[145] J. Hérault, B. Ans, and C. Jutten. Circuits neuronaux à synapses modifiables : Décodage de messages composites par apprentissage non supervisé. In *Comptes Rendus de l'Académie des Sciences*, pages 299(III–13) :525–528, 1984.

[146] J. Hérault, C. Jutten, and B. Ans. Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé. In *Actes du* $X^{me} colloque GRETSI$, pages 1017–1022, 1985.

[147] A. K. Jain and A. Ross. Fingerprint mosaicking. In *Internation Conference on Acoustics, Speech and Signal Processing*, pages IV : 4064–4067, 2002.

[148] A. K. Jain and A. Ross. Learning user-specific parameters in a multibiometric system. In *International Conference on Image Processing*, pages 57–60, 2002.

[149] A. K. Jain and A. Ross. Multibiometric systems. In *Communication of the ACM, special issue on multimodal interfaces*, 2004.

[150] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. In *IEEE Transactions on Circuits and Systems for Video Technology*, 2004.

[151] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition ? In *International Conference on Computer Vision*. IEEE, 2009.

[152] L. O. Jimenez and D. A. Landgrebe. Supervised classification in high-dimensional space : Geometrical, statistical, and asymptotical properties of multivariate data. *IEEE Transactions on Systems, Man and Cybernetics*, 28(1) :39–54, February 1998.

[153] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[154] C. Jutten and J. Herault. Blind separation of sources, part I : An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24 :1–10, 1991.

[155] K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. Fast inference in sparse coding algorithms with applications to object recognition. Technical report, Computational and Biological Learning Lab, Courant Institute, NYU, 2008.

[156] H. Kim, H. Park, and H. Zha. Distance preserving dimension reduction for manifold learning. In *International Conference on Data Mining*. SIAM, 2007.

[157] T. K. Kim, H. W. Kim, W. J. Hwang, and J. V. Kittler. Independent component analysis in a local facial residue space for face recognition. *Pattern Recognition*, 37(9) :1873–1885, September 2004.

[158] M. Kirby and 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, January 1990.

[159] J. V. Kittler. Combining classifiers : A theoretical framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(1) :18–27, 1998.

[160] B. Klare and A. K. Jain. Heterogeneous face recognition : Matching nir to visible light images. In *International Conference on Pattern Recognition*, Istanbul, August 2010.

[161] H. Kong, E. K. Teah, J. G. Wang, and R. Venkateswarlu. Two dimensional fisher discriminant analysis : Forget about small sample size problem. *Internation Conference on Acoustics, Speech and Signal Processing*, II :761–764, 2005.

[162] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T. W. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15(2) :349–396, 2003.

[163] K. Kreutz-Delgado and B. D. Rao. Focuss–based dictionary learning algorithms. In *Wavelet Applications in Signal and Image Processing*, volume 41, pages 19–53, 2000.

[164] D. J. Kriegman, J. P. Hespanha, and P. N. Belhumeur. Eigenfaces vs. fisherfaces : Recognition using class-specific linear projection. In *European Conference on Computer Vision*, pages I :43–58, 1996.

[165] D. J. Kriegman, J. P. Hespanha, and P. N. Belhumeur. Eigenfaces vs. fisherfaces : Recognition using class-specific linear projection. In *European Conference on Computer Vision*, pages I :43–58, 1996.

[166] J. B. Kruskal. Multidimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. *PSym*, 29 :1–29, 1964.

[167] A. Kumar and Y. Zhou. Human identification using knuclecodes. In *Biometrics : Theory, Applications and Systems*, 2009.

[168] S. Y. Kung and K. I. Diamantaras. A neural network learning algorithm for adaptive principal component extraction (APEX). In *Internation Conference on Acoustics, Speech and Signal Processing*, pages 861–864. Albuquerque, NM, 1990.

[169] S. Kunis and H. Rauhut. Random sampling of sparse trigonometric polynomials, II. orthogonal matching pursuit versus basis pursuit. *Foundations of Computational Mathematics*, 8(6) :737–763, 2008.

[170] F. De la Torre and M. J. Black. Robust principal component analysis for computer vision. In *International Conference on Computer Vision*, pages 362–369, 2001.

[171] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *Computers, IEEE Transactions on*, 42(3) :300–311, March 1993.

[172] S. Lafon and A. B. Lee. Diffusion maps and coarse-graining : A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9) :1393–1403, September 2006.

[173] L. Lam and C. Y. Suen. Optimal combinations of pattern classifiers. *Pattern Recognition Letters*, 16(9) :945–954, 1995.

[174] L. Lam and C. Y. Suen. Application of majority voting to pattern recognition : An analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics*, 27, 1997.

[175] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13(5) :393–401, June 1995.

[176] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition : A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1) :98–113, January 1997.

[177] Y. LeCun. Generalization and network design strategies. In R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels, editors, *Connectionism in Perspective*, Zurich, Switzerland, 1989. Elsevier. an extended version was published as a technical report of the University of Toronto.

[178] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In D. Touretzky, editor, *Neural Information Processing Systems*, volume 2. Morgan Kaufman, 1990.

[179] Y. LeCun, J. S. Denker, S. Solla, R. E. Howard, and L. D. Jackel. Optimal brain damage. In David Touretzky, editor, *Neural Information Processing Systems*, Denver, CO, 1990. Morgan Kaufman.

[180] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755) :788–791, October 1999.

[181] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Neural Information Processing Systems*, pages 556–562. MIT Press, 2000.

[182] J. A. Lee and M. Verleysen. Nonlinear dimensionality reduction of data manifolds with essential loops. *Neurocomputing*, 67 :29–53, 2005.

[183] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya. Learning unions of orthonormal bases with thresholded singular value decomposition. In *Internation Conference on Acoustics, Speech and Signal Processing*, 2005.

[184] M. S. Lewicki, H. Hughes, and B. A. Olshausen. A probabilistic framework for the adaptation and comparison of image codes. *Journal of the Optical Society of America*, November 04 1998.

[185] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12 :337–365, February 2000.

[186] Qi Li, Jieping Ye, and Chandra Kambhamettu. Linear projection methods in face recognition under unconstrained illuminations : A comparative study. In *CVPR (2)*, pages 474–481, 2004.

[187] Y. M. Li. On incremental and robust subspace learning. *Pattern Recognition*, 37(7) :1509–1518, July 2004.

[188] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li. Learning mutli–scale block local binary patterns for face recognition. *International Conference on Biometrics*, (12) :828–837, 2007.

[189] A. Lima, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura. On the use of kernel–pca for feature extraction in speech recognition. In *Transactions on Information Systems*, volume 12, pages 2802–2811, 2004.

[190] S. H. Lin, S. Y. Kung, and L. J. Lin. Face recognition/detection by probabilistic decision-based neural-network. *IEEE Trans. Neural Networks*, 8(1) :114–132, January 1997.

[191] C. Liu and H. Wechsler. A unified bayesian framework for face recognition. In *International Conference on Image Processing*, pages 151–155, 1998.

[192] Chengjun Liu. Gabor-based kernel PCA with fractional power polynomial models for face recognition. *IEEE Trans. Pattern Anal. Mach. Intell*, 26(5) :572–581, 2004.

[193] M. Loève. Fonctions aléatoires du second ordre. *Processus stochastiques et mouvements browniens*, 1948.

[194] Juwei Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *IEEE-NN*, 14 :117–126, January 2003.

[195] B. Mailhé, S. Lesage, R. Gribonval, F. Bimbot, and P. Vandergheynst. Shift-invariant dictionary learning for sparse representations : extending k-svd. In *European Signal Processing Conference*, 2008.

[196] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, *International Conference on Machine Learning*, volume 382 of *ACM International Conference Proceeding Series*, page 87. ACM, 2009.

[197] S. Mallat and G. Peyré. A review of bandlet methods for geometrical image representation. *Numerical Algorithms*, 44(3) :205–234, 2007.

[198] S. Mallat and G. Peyré. Orthogonal bandlet bases for geometric images approximation. *Communications on Pure and Applied Mathematics*, February 10 2009.

[199] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. Technical report, inst-courant-cs, November 1992.

[200] A. M. Martínez and A. C. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2) :228–233, 2001.

[201] T. Melzer, M. Reiter, and H. Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. *Lecture Notes in Computer Science*, 2130 :353–361, 2001.

[202] C. Meszaros. On the sparsity issues of interior point methods for quadratic programming. Technical report, Laboratory of Operations Research and Decision Systems, Hungarian Academy of Sciences, 1998.

[203] F. G. Meyer, R. R. Coifman, and J. Kovacevic. Brushlets : A tool for directional image analysis and image compression. *Applied and Computational Harmonic Analysis*, February 04 1997.

[204] S. Mika, B. Scholkopf, and A. Smola. Kernel PCA and de-noising in feature spaces. *Neural Information Processing Systems*, December 21 1999.

[205] S. Mika and J. Weston. Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing*, May 06 1999.

[206] M. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, Mass, 1969.

[207] B. Moghaddam and A. P. Pentland. Probabilistic visual learning for object detection. In *International Conference on Computer Vision*, pages 786–793, 1995.

[208] Y. Moon, H. Yeung, K. Chan, and S. Chan. Template synthesis and image mosaicking for fingerprint registration : An experimental study. In *Internation Conference on Acoustics, Speech and Signal Processing*, pages 409–412, 2004.

[209] J. J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes Rendus de l'Académie des Sciences, Série A Mathématiques*, 255 :2897–2899, 1962.

[210] J. J. Moreau. Propriétés des applications « prox ». *Comptes Rendus de l'Académie des Sciences, Série A Mathématiques*, 256 :1069–1071, 1963.

[211] J. J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93 :273–299, 1965.

[212] H. Murakami and B. V. K. Vijaya Kumar. Efficient calculation of primary image from a set of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(5) :511–515, September 1982.

[213] J.F. Murray and K. Kreutz-Delgado. An improved focuss–based learning algorithm for solving sparse linear inverse problem. In *IEEE International Conference on Signals, Systems and Computers*, volume 41, pages 19–53, 2001.

[214] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Journal of applied and computational harmonic analysis*, March 22 2005. Comment : submitted to journal of applied and computational harmonic analysis.

[215] K. Nandakumar. *Integration of Multiple Cues in Biometric Systems.* PhD thesis, Michigan State University, May 2005.

[216] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM*, 24(2) :227–234, 1995.

[217] A. Nefian. *A hidden Markov model-based approach for face detection and recognition*. PhD thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 1999.

[218] Y. Nesterov. Gradient methods for minimizing composite objective function. 2007. CORE discussion paper – Université Catholique de Louvain.

[219] M. Nikolova. Local strong homogeneity of a regularized estimator. *SIAM Journal of Applied Mathematics*, 61(2) :633–658, 2000.

[220] Kazunori Okada and Christoph von der Malsburg. Pose-invariant face recognition with parametric linear subspaces. In *Proceedings of* the IEEE International Conference on Automatic Face and Gesture Recognition, pages 71–76, May 2002.

[221] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive-field properties by learning a sparse code for natural images. *Nature*, 381(6583) :607–609, June 1996.

[222] B. A. Olshausen and D. J. Field. Natural image statistics and efficient coding. *Computation in Neural Systems*, 7(2) :333–339, May 1996.

[223] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set : A strategy employed in V1 ? *Vision Research*, 37 :3311–3325, 1997.

[224] A. O'Toole, S. L. Snow, J. Huarms, D. R. Hurst, M.Pappas, J. H. Ayyad, and H. Abdi. A video database of moving faces and people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[225] P. Paatero and U. Tapper. Positive matrix factorization : A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2) :111–126, 1994.

[226] S. Pang and N. Kasabov. Investigating LLE eigenface on pose and face identification. In Jun Wang, Zhang Yi, Jacek M. Zurada, Bao-Liang Lu, and Hujun Yin, editors, *International Symposium on Neural Networks*, volume 3972 of *Lecture Notes in Computer Science*, pages 134–139, 2006.

[227] Y. Pang, Z. Liu, and Y. Sun. Subspace learning based on laplacian eigenmaps and LDA for face recognition. *International Journal on Information Acquisition*, 3(1) :45–51, 2006.

[228] K. Pearson. On lines and planes of closest fit to points in space. *Philosophical Magazine*, 2 :559–572, 1901.

[229] P. S Penev and J. J Atick. Local feature analysis : a general statistical theory for object representation. *Network : Computation in Neural Systems*, 7(3) :477–500, August 1996.

[230] E. Le Pennec and S. G. Mallat. Sparse geometric image representations with bandelets. *IEEE Transactions on Image Processing*, 14(4) :423–438, April 2005.

[231] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[232] V. Perlibakas. Face recognition using principal component analysis and log-gabor filters. *CoRR*, abs/cs/0605025, 2006.

[233] F. Perronnin, J. L. Dugelay, and K. Rose. A probabilistic model of face mapping with local transformations and its application to person recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7) :1157–1171, 2005.

[234] P. Jonathon Phillips, W. Todd Scruggs, Alice J. O'Toole, Patrick J. Flynn, Kevin W. Bowyer, Cathy L. Schott, and Matthew Sharpe. FRVT 2006 and ICE 2006 large-scale experimental results. *Pattern Analysis and Machine Intelligence*, 32(5) :831–846, 2010.

[235] J. R. Price and T. F. Gee. Face recognition using direct, weighted linear discriminant analysis and modular subspaces. *Pattern Recognition*, 38(2) :209–219, February 2005.

[236] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1) :145–151, 1999.

[237] B. Raducanu and F. Dornaika. Dynamic facial expression recognition using laplacian eigenmaps-based manifold learning. In *International Conference on Robotics and Automation*, pages 156–161. IEEE, 2010.

[238] B. Raghavendra, R.and Dorizzi, A. Rao, and G.K Hemantha. Pso versus adaboost for feature selection in multimodal biometrics. In *Biometrics : Theory, Applications and Systems*, 2009.

[239] R. Raghavendra, B. Dorizzi, A. Rao, and G.H. kumar. Designing efficient fusion schemes for multimodal biometric systems using face and palmprint. *Pattern Recognition*, 2010.

[240] R. Raghavendra, Bernadette Dorizzi, Ashok Rao, and G. Hemantha Kumar. Particle swarm optimization based fusion of near infrared and visible images for improved face verification. *Pattern Recognition*, 44(2) :401–411, 2011.

[241] M. A. Ranzato, Y-L. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Neural Information Processing Systems*. MIT Press, 2007.

[242] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain. Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4, 2000.

[243] F. Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65 :386–408, 1958.

[244] A. Ross. Multibiometrics. In Stan Z. Li and Anil K. Jain, editors, *Encyclopedia of Biometrics*, pages 967–973. Springer US, 2009.

[245] S. T. Roweis. EM algorithms for PCA and SPCA. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Neural Information Processing Systems*. The MIT Press, 1997.

[246] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500) :2323–2326, December 2000.

[247] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel distributed processing : Explorations in the microstructure of cognition, Volume 1 : Foundations*. MIT Press, 1986.

[248] Z. Saidane and C. Garcia. Robust binarization for video text recognition. In *International Conference on Document Analysis and Recognition*, pages 874–879, 2007.

[249] R. Salomon. Improved convergence rate of back-propagation with dynamic adaption of the learning rate. *Lecture Notes in Computer Science*, 496 :269, 1991.

[250] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of the 2nd IEEE workshop on Applications of Computer Vision*, Sarasota, Florida, 1994.

[251] C. Sanderson and K. K. Paliwal. Information fusion and person verification using speech and face information. In *Technical Report*, 2002.

[252] S. Sardy, A. Bruce, and P. Tseng. Block coordinate relaxation methods for nonparametric signal denoising with wavelet dictionaries. Technical report, Department of Mathematics, University of Washington, Seattle, WA, October 1998.

[253] A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng. On random weights and unsupervised feature learning. In *Wokshop on deep learning and unsupervised feature learning, Neural Information Processing Systems*, 2010.

[254] A. Saxena, A. Gupta, and A. Mukerjee. Non-linear dimensionality reduction by locally linear isomaps. In Nikhil R. Pal, Nikola Kasabov, Rajani K. Mudi, Srimanta Pal, and Swapan K. Parui, editors, *Neural Information Processing*, volume 3316 of *Lecture Notes in Computer Science*, pages 1038–1043. Springer, 2004.

[255] B. Scholkopf, A. Smola, and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, pages 1299–1319, 1998.

[256] H. Schwenk. The diabolo classifier. *Neural Computation*, 10(8) :2175–2200, 1998.

[257] F. Sha and L. K. Saul. Analysis and extension of spectral methods for nonlinear dimensionality reduction. In Luc De Raedt and Stefan Wrobel, editors, *International Conference on Machine Learning*, volume 119 of *ACM International Conference Proceeding Series*, pages 784–791. ACM, 2005.

[258] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. CUP, June 2004.

[259] J. Shen and G. W. Israël. A receptor model using a specific non-negative transformation technique for ambient aerosol. *Atmospheric Environment*, 23(10) :2289–2298, 1989.

[260] R. Singh, M. Vatsa, and A. Noore. Integrated multilevel image fusion and match score fusion of visible and infrared face images for robust face recognition. *Pattern Recognition*, 41(3) :880–893, March 2008.

[261] Richa Singh, Mayank Vatsa, and Afzel Noore. Hierarchical fusion of multispectral face images for improved recognition performance. *Information Fusion*, 9(2) :200–210, 2008.

[262] D. Skočaj. *Robust Subspace Approaches to Visual Learning and recognition*. Delo ali doktorska disertacija ; peerreviewed, University of Ljubljana, February 2003.

[263] D. Skočaj, H. Bischof, and A. Leonardis. A robust PCA algorithm for building representations from panoramic images. In *European Conference on Computer Vision*, page IV : 761 ff., 2002.

[264] D. Skočaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In *International Conference on Computer Vision*, pages 1494–1501, 2003.

[265] R. Snelick, U. Uludag, A. Mink, M. Indovina, and A. K. Jain. Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3) :450–455, March 2005.

[266] D. A. Socolinsky and A. Selinger. Thermal face recognition in an operational scenario. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1012–1019, 2004.

[267] L. I. Soo, P. de Heras Ciechomski, S. Sarni, and D. Thalmann. Planar arrangement of high-dimensional biomedical data sets by isomap coordinates. *IEEE Symposium on Computer-Based Medical Systems*, 2003.

[268] J. A. K. Suykens. Data visualization and dimensionality reduction using kernel maps with a reference point. *IEEE Transactions on Neural Networks*, 19(9) :1501–1517, 2008.

[269] D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8) :831–836, 1996.

[270] B. Takàcs. Comparing face images using the modified hausdorff distance. *Pattern Recognition*, 31(12) :1873–1881, December 1998.

[271] Teewoon Tan and Hong Yan. Analysis of the contractivity factor in fractal based face recognition. In *ICIP (3)*, pages 637–641, 1999.

[272] Y. W. Teh and S. T. Roweis. Automatic alignment of local representations. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Neural Information Processing Systems*, pages 841–848. MIT Press, 2002.

[273] M. N. Teli. Dimensionality reduction using neural networks. Technical report, April 02 2008.

[274] J. B. Tenenbaum. Mapping a manifold of perceptual observations. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Neural Information Processing Systems*. The MIT Press, 1997.

[275] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500) :2319–2323, December 2000.

[276] M. E. Tipping. Sparse kernel principal component analysis. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Neural Information Processing Systems*, pages 633–639. MIT Press, 2000.

[277] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, Aston St, Birmingham, B4 7ET, UK, September 1997.

[278] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2) :443–482, 1999.

[279] L. Torres, L. Lorente, and J. Vila. Automatic face recognition of video sequences using self–eigenfaces. *International Symposium on Image/video Communication over Fixed and Mobile Networks*, December 11 2000.

[280] B. J. Tromberg, M. Prasad, G. Healey, and Z. H. Pan. Face recognition in hyperspectral images. In *CVPR*, pages I : 334–339, 2003.

[281] J. A. Tropp. Greed is good : algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 2004.

[282] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–590, Hawai, June 1992.

[283] M. Uray, D. Skocaj, P. M. Roth, H. Bischof, and A. Leonardis. Incremental LDA learning by combining reconstructive and discriminative approaches. In *British Machine Vision Conference*, pages xx–yy, 2007.

[284] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis of image ensembles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages II : 93–99, 2003.

[285] P. A. Viola and M. J. Jones. Robust real-time face detection. In *International Conference on Computer Vision*, page 747, 2001.

[286] M. Visani, C. Garcia, and J. M. Jolion. Normalized radial basis function networks and bilinear discriminant analysis for face recognition. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 342–347, 2005.

[287] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon. Accelerating the convergence of the backpropagation method. *Biological Cybernetics*, 59 :257–263, 1988.

[288] J. Wang, C. Zhang, and Z. Kou. An analytical mapping for LLE and its application in multi-pose face synthesis. In *British Machine Vision Conference*, pages xx–yy, 2003.

[289] Q. Wang and J. Li. Combining local and global information for nonlinear dimensionality reduction. *Neurocomputing*, 72(10-12) :2235–2241, 2009.

[290] Y. Wang, Y. Jia, C. Hu, and M. Turk. Non-negative matrix factorization framework for face recognition. *Internation Journal of Pattern Recognition and Artificial Intelligence*, 19(4) :495–511, 2005.

[291] K. Q. Weinberger, B. D. Packer, and L. K. Saul. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In *International Workshop on Artificial Intelligence and Statistics*, pages 381–388, 2005.

[292] P. Weiss. *Algorithmes rapides d'optimisation convexe. Applications à la reconstruction d'images et à la détection de changements.* PhD thesis, ENS Cachan, December 31 2008.

[293] Y. Weiss. Segmentation using eigenvectors : A unifying view. In *International Conference on Computer Vision*, pages 975–982, 1999.

[294] S. Wild, J. Curry, and A. Dougherty. Improving non-negative matrix factorizations through structured initialization. *Pattern Recognition*, 37(11) :2217–2232, November 2004.

[295] L. Wiskott, J.-M. Fellous, N. Kruger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 129–132.

[296] L. Wiskott, J. M. Fellous, N. Kruger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) :775–779, July 1997.

[297] L. Wolf and A. Shashua. Learning over sets using kernel principal angles. *Journal of Machine Learning Research*, 4 :913–931, 2003.

[298] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[299] L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3) :418–435, 1992.

[300] L. Xu and A. L. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks*, 6(1) :131–143, January 1995.

[301] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *Automatic Face and Gesture Recognition*, pages 318–323, 1998.

[302] M. H. Yang. Face recognition using extended isomap. In *International Conference on Image Processing*, pages 117–120, 2002.

[303] H. Yu and J. Yang. A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, 34(10) :2067–2070, October 2001.

[304] H. Zhang, J. Winkeler, Y. F. Wang, B. S. Manjunath, and S. Chandrasekaran. An eigenspace update algorithm for image analysis. In *Technical Report*, 1995.

[305] T. Zhang, J. Yang, D. Zhao, and X. Ge. Linear local tangent space alignment and application to face recognition. *Neurocomputing*, 70(7-9) :1547–1553, 2007.

[306] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal of Scientific Computing*, 26(1) :313–338, 2004.

[307] W. Zhao. *Robust Image-based 3d Face Recognition*. PhD thesis, University of Maryland, 1999.

[308] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition : A literature survey. *CSURV : Computing Surveys*, 35, 2003.

[309] X. S. Zhuang and D. Q. Dai. Inverse fisher discriminate criteria for small sample size problem and its application to face recognition. *Pattern Recognition*, 38(11) :2192–2194, November 2005.