

Nouvelle approche de réduction des faux rejets dans un système d'authentification par empreintes digitales.

Thèse

Présentée en vue de l'obtention du

Doctorat de l'Université de Yaoundé I
(Spécialité Informatique)

par :

KOUAMO NGONGANG Stephane Orozco

Composition du jury



UNIVERSITE DE YAOUNDE I
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

BP : 812 YAOUNDE
Email : deptinfo@uy1.uninet.cm



UNIVERSITY OF YAOUNDE I
FACULTY OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

BP : 812 YAOUNDE
Email : deptinfo@uy1.uninet.cm

ATTESTATION DE CORRECTION DE LA THESE

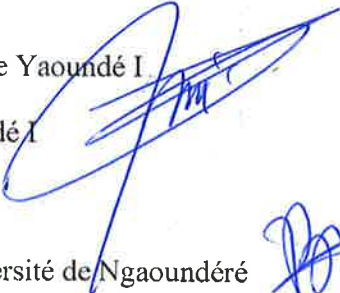
Je soussigné, **Maurice TCHUENTE**, Professeur, Président du jury de la soutenance de thèse de Doctorat/PhD en informatique soutenue par Monsieur **KOUAMO NGONGANG Stéphane Orozco** Matricule **98Y311** atteste que la thèse intitulée « **Nouvelle approche de réduction des faux rejets dans un système d'authentification par empreintes digitales** », présentée par le candidat a été corrigée conformément aux recommandations des membres du jury.

En foi de quoi la présente attestation lui est établie et délivrée pour servir et valoir ce que de droit.

Président : TCHUENTE Maurice, Professeur, Université de Yaoundé I

Rapporteurs :

FOUDA NDJODO Marcel, Professeur, Université de Yaoundé I


(H) TANGHA Claude, Professeur, Université de Yaoundé I

Membres :

YENKE Blaise Omer, Maître de Conférences, Université de Ngaoundéré

KOUAMOU Georges-Edouard, Maître de Conférences, Université de Yaoundé I

DJOTIO NDIE Thomas, Maître de Conférences, Université de Yaoundé I

NDOUNDAM René, Maître de Conférences, Université de Yaoundé I

Dédicace

A mon Père, Monsieur KOUAMO Pierre⁺

Papa,

Tu m'as appris le sens de la famille et des amis.

Tu m'as appris le sens du travail et de la persévérance.

Tu m'as appris à être humble comme tu l'as été toute ta vie durant.

Tu m'as toujours donné les conditions d'études dans la limite de tes moyens et un cadre familiale favorable. Pour moi, tu as été le meilleur père qu'un enfant puisse rêver d'avoir et, tu m'as donné bien plus que ce que j'aurais pu imaginé.

Merci papa pour tout, trouve en ce travail un des fruits de tous les efforts et sacrifices que tu n'as jamais cessé de consentir pour moi et mes Frères.

Le sort a voulu que tu ne sois pas présent physiquement au moment de la soutenance de cette thèse que tu as tant souhaité, mais je suis sûr d'une chose, c'est que, de la haut tu me regarde et tu es fier de moi.

Je te dédie ce travail à toi mon papa.

Que Dieu t'accueille auprès de lui.

Remerciements

Au Professeur Claude TANGHA⁺

Pour avoir su raffermir ma passion pour la recherche en Informatique, pour l'intérêt qu'il a porté à ces travaux, pour avoir toujours été disponible, d'oreille attentive et de conseils pointus pendant tout ce travail.

Aujourd'hui, vous n'êtes plus là Professeur, mais je sais que de la haut vous êtes heureux que ce travail acharné auquel vous avez pleinement contribué, aboutisse enfin.

Puisse la terre de nos ancêtres vous soit légère.

A tous les membres du jury

Au Professeur Marcel FOU DA NDJODO

Pour l'intérêt qu'il a bien voulu porter sur ce travail, pour ses remarques et conseils ô combien importants pour la finalisation de ce travail.

Au Professeur Maurice TCHUENTE

Pour son sérieux, les enseignements et astuces reçus tout au long de mes cycles de Master et Doctorat.

Au Professeur Marcelin NKENLIFACK

Pour sa pertinence dans ses remarques aussi importantes que judicieuses.

Au Professeur Laure Pauline FOTSO

Pour son soutien moral, ses conseils pointus et son aide précieuse pour la réalisation de ce travail.

Au Professeur Thomas BOUETOU BOUETOU

Pour l'intérêt qu'il a porté à ces travaux et pour tous ses conseils pertinents.

Au Professeur Thomas DJOTIO NDIE

Pour ses conseils aussi judicieux qu'importants pour la réalisation de ce travail.

Au Professeur Georges Edouard KOUAMOU

Pour ses remarques pertinentes, importantes et judicieuses pour ce travail.

Au Professeur Blaise YENKE

Pour son sérieux et son sens de la répartie.

Au Professeur Daniel NJOPWOUO

Pour son aide précieuse sans laquelle ces travaux n'auraient pas pu être réalisés.

Au Professeur René NDOUNDAM

Pour l'intérêt qu'il a porté à mes travaux et pour tous les enseignements reçus.

Au Docteur Gilbert TINDO

Pour son apport oh ! combien important, son sérieux et sa disponibilité dans le travail.

Au Docteur AMINOU Halidou

Pour sa disponibilité et son sens de la répartie.

Au Docteur Paulin MELATAGIA

Pour son appui précieux durant tout mes cycles de Master et Doctorat.

Au Docteur Nobert TSOPZE

Pour son aide précieuse dans mes travaux de recherche.

Au Docteur Bernabé BATCHAKUI

Pour son soutien tant moral que spirituel dans tous les moments difficiles passés.

**A tous le Corps Enseignant du Département d'Informatique de l'Université
de Yaoundé I**

Pour tous les enseignements reçus

A tous mes camarades du Département d'Informatique

Avec qui tout au long des années nous avons eu à travailler dur pour arriver à nos fins

A toute la famille Moulines :

Professeur Eric MOULINES, Corine BERTHELOT, Vincent MOULINES et Nathalie MOULINES.

A ma famille

Pour le soutien sans faille et la patience qu'ils ont su me montrer :

Ma sœur et son époux,

Mme DJANANG née KOUAMO Germaine et Mr DJANANG TCHALLE Jules⁺

A mon grand frère

Docteur KOUAMO Olaf

A tous mes petits frères

Mr KOUAMO NCHAPMOU Cédric, Mr KOUAMO Hermann, Mr KOUAMO Ebenezer Owens et Mr KOUAMO KAMENI Hervé.

Ce travail est aussi le fruit de vos efforts consentis.

A ma chérie,

Mme KOUAMO Née CHEBOU FEKOU Sonia

A mes neveux,

DJANANG Francesca, DJANANG Nathan, DJANANG Frédérique, NGASSAM Hélène-Chloé, DJANANG Alexandra, KOUAMO Pierre-Kendrick et KOUAMO KAMENI Pierrette Imane.

Sans oublier ma grand-mère (maman KAMENI Helène⁺) qui a tant œuvré pour notre réussite, mes Frères et moi.

Enfin à ma mère, *maman KOUAMO Rosalie.*

Maman, tu as toujours été là toutes les fois où nous avons eu besoin de toi, tu nous as toujours protégé et aimé du mieux que tu pouvais, en nous montrant le bon chemin. Pour toutes ces fois maman, je te dis infiniment merci du fond cœur.

Je remercie toutes les personnes dont j'aurai oublié le nom et qui aurait contribué d'une manière ou d'une autre à la réalisation de ce travail.

Table des matières

Notations	viii
Table des figures	xiii
Liste des tableaux	xiv
Résumé	xv
Abstract	xvi
Publications	1
Introduction Générale	1
1 <i>La Biométrie et les empreintes digitales</i>	7
1.1 <i>Introduction</i>	7
1.1.1 <i>Qu'est ce que la Biométrie ?</i>	7
1.1.2 <i>Les différentes techniques biométriques</i>	9
1.1.3 <i>Les champs d'application de la biométrie</i>	19
1.2 <i>Les empreintes digitales</i>	20
1.2.1 <i>Historique</i>	20
1.2.2 <i>Définitions</i>	22
1.2.3 <i>Caractéristiques d'une empreinte digitale</i>	23
1.2.4 <i>Le relevé des traces digitales</i>	27
1.3 <i>Les capteurs d'empreintes digitales</i>	28
1.3.1 <i>Les capteurs optiques d'empreintes</i>	29
1.3.2 <i>Les capteurs en sillicium</i>	31
1.3.3 <i>Les capteurs électriques-thermique</i>	32
1.3.4 <i>Les capteurs capacitifs</i>	33

1.3.5	<i>Les capteurs de champ-électrique</i>	33
1.3.6	<i>Les capteurs de pression</i>	34
1.3.7	<i>Les capteurs sans contact</i>	35
1.3.8	<i>La scannerisation à partir du papier</i>	35
1.4	<i>Les problèmes liés à la biométrie</i>	36
1.5	<i>Conclusion</i>	37
2	<i>Principes fondamentaux de la reconnaissance d'empreintes digitales</i>	38
2.1	<i>Introduction</i>	38
2.1.1	<i>L'enregistrement des templates dans la base</i>	39
2.1.2	<i>L'identification/authentification</i>	40
2.2	<i>Traitement et analyse de l'empreinte</i>	42
2.2.1	<i>Le contrôle de qualité de l'image d'empreinte</i>	42
2.2.2	<i>La détection des minuties</i>	62
2.3	<i>La comparaison d'empreintes digitales</i>	65
2.3.1	<i>La comparaison basée sur la corrélation</i>	66
2.3.2	<i>La comparaison basée sur les minuties</i>	67
2.3.3	<i>La comparaison basée sur les singularités de lignes d'empreintes digitales</i>	69
2.4	<i>La classification des empreintes digitales</i>	70
2.5	<i>Conclusion</i>	72
3	<i>Introduction aux réseaux de neurones</i>	73
3.1	<i>Introduction</i>	73
3.1.1	<i>Historique</i>	73
3.1.2	<i>Définition</i>	75
3.1.3	<i>Utilités et limites</i>	76
3.1.4	<i>Structure d'un réseau de neurones artificiels</i>	78
3.2	<i>Les étapes de la conception d'un réseau de neurones</i>	83
3.2.1	<i>Choix et préparation des échantillons</i>	83
3.2.2	<i>Elaboration de la structure du réseau</i>	83
3.2.3	<i>Apprentissage</i>	85
3.2.4	<i>Validation et tests</i>	87
3.3	<i>Les différents types de réseaux de neurones</i>	87
3.3.1	<i>Réseau de neurones à apprentissage supervisés</i>	88
3.3.2	<i>Réseau de neurones à apprentissage non supervisé</i>	89
3.4	<i>Les réseaux de neurones pour les pré-traitements</i>	91
3.4.1	<i>Transformation</i>	91

3.4.2	<i>Prédiction</i>	92
3.4.3	<i>Segmentation</i>	92
3.5	<i>Les réseaux de neurones pour la quantification</i>	93
3.5.1	<i>Codage en quantification scalaire et vectorielle</i>	93
3.5.2	<i>Construction du dictionnaire pour la QV</i>	94
3.6	<i>Réseaux de neurones basés sur l'apprentissage de Hebb</i>	95
3.7	<i>Conclusion</i>	96
4	<i>La reconnaissance d'empreintes digitales avec les réseaux de neurones</i>	98
4.1	<i>Introduction</i>	98
4.2	<i>Présentation des méthodes</i>	99
4.2.1	<i>Apprentissage profond</i>	100
4.2.2	<i>Méthode basée sur les réseaux de neurones probabilistes</i>	104
4.2.3	<i>Méthode basée sur le réseau de neurones localement connectés</i>	113
4.3	<i>Présentation de la méthode proposée</i>	120
4.3.1	<i>Structure du réseau de neurones proposée</i>	123
4.3.2	<i>Construction de la base de données</i>	124
4.3.3	<i>Apprentissage</i>	126
4.3.4	<i>Utilisation</i>	128
4.4	<i>Résultats expérimentaux et discussions</i>	130
4.4.1	<i>Choix et préparation des échantillons</i>	130
4.4.2	<i>Apprentissage avec des sous-blocs de taille 8×8</i>	133
4.4.3	<i>Apprentissage avec des sous-blocs de taille 16×16</i>	134
4.5	<i>Conclusion</i>	146
	Conclusion Générale	147
	Bibliographie	150
	Annexes	164
	Index	167

Notations

$M_k(i, j)$	Valeur du k^{ieme} pixel (i, j) dans une image représentée par la matrice M ;
$\nabla f(x_i, y_j)$	Gradient d'une fonction f au point (x_i, y_j) ;
$\nabla z(A, B)$	tableau des différences carrés entre 2 images A et B ;
$P_{G,C}(i, j)$	index de Poincaré d'un pixel (i, j) calculé sur une courbe C dans un champ de vecteurs G ;
$p(\nabla z)$	probabilité de correspondance ;
$cn(i)$	crossing number d'un pixel i ;
(x, y)	exemple à apprendre ;
$d(x, y)$	distorsion ou erreur quadratique entre x et y ;
$F(u, v)$	transformée de fourrier discrète au point (u, v) ;
E	erreur en sortie d'un neurone ;
E_i	erreur en sortie du neurone i ;
E_i^l	erreur en sortie du neurone i de la couche l ;
e_i	état interne du neurone i ;
g_k	k^{ieme} classe de données de la seconde couche de classification ;
h_l'	ensemble de données de la l^{ieme} couche d'identification ;
z'_{kl}	connexion entre le neurone k et le neurone l ;
o_i	i^{ieme} sortie attendue ;
W	vecteur de poids de connexion ;
$w_{i,j}$	connexion entre le neurone i et le neurone j ;
Δw_{ij}	fonction de modification des poids de connexion entre les neurones i et j ;
$\alpha_{ij}(n)$	niveau d'activation courant du neurone (i, j) après n itérations ;
ADALINE	ADaptive LInear NEuron ;
ADN	Acide DésoxyriboNucléique ;

AFIS	Automated Fingerprint Identification System ;
AHC	Adaptive Heuristic Critic ;
ALN	Adaptive Logic Network ;
AMF	Avalanche Matched Filter ;
ANSI	American National Standard Institute ;
ARP	Associative Reward Penalty ;
BD	Base de Données ;
CARI	Colloque Africain sur la Recherche en Informatique et en Mathématiques Appliquées ;
CasCor	Cascade Correlation ;
CCD	Charge Coupled Device ;
CH	Continous Hopfield ;
CMOS	Common Memory Operating System ;
CNN	Cellular Neural Network ;
DFO	DiazaFluOrnone ;
DH	Discrete Hopfield ;
DPI	Dots Per Inch ;
DTC	Dispositif à Transfert de Charge ;
EER	Equal Error Rate ;
FBI	Federal Bureau of Investigation ;
FFT	File Function Transfert ;
FVCNN	Fingerprint Verification for Convolutional Neural Network
HCR	Haut Commissariat aux Réfugiés ;
IBG	International Biometric Group.
ISO	International Standard Organisation ;
KLT	Kahornen Loeve Transformation ;
LED	Diodes Electroluminescentes ;
LOD	Lateral Overflow Drain ;
MIT	Massassuchetts Institute of Technology ;
MLP	Multi Layer Perceptron ;
NN	Neural Network ;
OBS	Optimal Brain Surgeon ;
OCR	Optical Characters Recognition ;
Perc	Backpercolation ;
PSF	Point Spread Function ;

PNN	Probabilistic Neural Network ;
RAM	Random Access Memory ;
RBF	Radial Basis Function ;
RNA	Réseaux de Neurones Artificiels ;
SOM	Self Organization Map ;
TDNN	Time Delay Neural Network ;
TF	Transformée de Fourier ;
TFA	Taux de Fausses Acceptations ;
TFR	Taux de Faux Rejets ;
TFR	Transformée de Fourier Rapide ;
US	United State ;

Table des figures

1.1	Revenus annuels de la biométrie en millions de Dollars US de 2009 à 2019 (Copyright ©2019 International Biometric Group)	8
1.2	Part des marchés des différentes techniques biométriques	10
1.3	Exemples de quelques caractéristiques biométriques : (a) Empreinte digitale, (b) visage, (c) main, (d) oreille, (e) démarche, (f) iris de l'œil, (g) empreinte vocale, (h) signature, (i) ADN, (j) rétine.	10
1.4	Principales classes d'empreintes digitales selon Galton-Henry [58]	23
1.5	Caractéristiques d'une empreinte digitale [109]	24
1.6	Les différents types de minutie	25
1.7	relevé des traces d'empreintes digitales	27
1.8	Capteur optique CCD	29
1.9	Capteur en silicium © <i>Eikon</i>	31
1.10	Capteur électrique-thermique	32
1.11	Capteur capacitif	33
1.12	Capteur de pression	34
1.13	Relation entre le TFR et le TFA	36
2.1	Principe générale d'un système de reconnaissance d'empreintes digitales	39
2.2	Illustration du processus d'enregistrement/enrôlement	40
2.3	Illustration du processus d'authentification	41
2.4	Illustration du processus d'identification	41
2.5	(a) les lignes ont une orientation verticale, (b) les lignes ont une orientation oblique à gauche, (c) les lignes ont une orientation horizontale.	43
2.6	(a) Image de hauteur 6 et de largeur 10, (b) la matrice 6×10 des pixels de l'image (a)	44
2.7	L'axe oblique indique l'orientation des lignes au voisinage du point (i, j)	44
2.8	Exemple de calcul d'orientation d'une minutie	46
2.9	L'index de Poincaré calculé sur une courbe C ; $P_{G,C} = -263^\circ$	47

2.10	Détection d'une région singulière par l'algorithme de Poincaré	49
2.11	Image binarisée avec des tailles fixes.	50
2.12	Squelettisation d'une empreinte digitale	51
2.13	Fenêtre d'érosion conditionnelle	52
2.14	Principe du filtrage sur la FFT	55
2.15	Application du filtrage spatial sur un bloc d'image	57
2.16	Application du filtre de Gabor	59
2.17	Application du filtre fréquentiel auto-adaptatif	61
2.18	Calcul du crossing number dans un voisinage de 8 pixels (cadre rouge)	64
2.19	Matching d'empreintes deux à deux	66
2.20	La comparaison de minuties de deux empreintes	67
2.21	Le matching de minuties	69
2.22	Les trois classes d'empreintes digitales selon Galton	71
2.23	Les cinq classes les plus utilisées.	71
3.1	Vue simplifiée d'un réseau de neurone	76
3.2	Vue simplifiée d'un neurone biologique	78
3.3	Structure d'un neurone formel	79
3.4	La fonction logistique : $Y = \frac{2}{(1+e^{(-2 \times X)-1})}$	82
3.5	La tangente hyperbolique : $Y = \frac{2}{(1+e^{(-2 \times X)-1})}$	82
3.6	La fonction de Gauss : $Y = e^{(\frac{-X^2}{2})}$	82
3.7	Une fonction à seuil : $Y = 0$ si $X \leq 0$ et $Y = 1$ si $X > 0$	83
3.8	Structure d'un réseau à propagation avant classique	84
3.9	Structure d'un réseau à propagation avant	84
3.10	Perceptron à trois couches	88
3.11	Structure d'un réseau de neurones basée sur l'apprentissage de Hebb	95
4.1	Processus d'acquisition d'une image à l'aide d'une camera CCD [169].	106
4.2	Structure du réseau de neurones probabiliste.	110
4.3	Réseau de Baldi pour la décision.	112
4.4	Réseau de neurones connectés localement.	115
4.5	Neurones d'entrée et de connexion.	116
4.6	Structure du réseau de neurones proposé.	124
4.7	Quelques exemples d'images d'apprentissage	125
4.8	Illustration de la première Classification	129
4.9	Illustration de la deuxième classification	129
4.10	Processus d'identification par classe de données	130

4.11	Technique du Dropout	131
4.12	Exemples d'images de test provenant du même individu	134
4.13	TFA et TFR en fonction du seuil de décision	143
4.14	TFA et TFR en fonction du nombre de minuties	144
4.15	Quelques images de la base de données utilisée par Baldi lors de ses tests : 512 × 464 avec 8 bits par pixel	164
4.16	Quelques images de la base de données BDAL	165
4.17	Quelques images de la base de données BDAL (2)	166

Liste des tableaux

1.1	Comparaison des systèmes biométriques selon différents critères : (Acc) Acceptabilité, (Univ) Universalité, (Uni) Unicité, (Perm) Permanence, (Fenreg) Facilité d'enregistrement, (Perf) Performance, (Infal) Infalsifiabilité. Notation : (+) fort, (=) moyen, (-) faible	19
1.2	Probabilité de collision de 2 empreintes selon plusieurs auteurs [124]	26
3.1	Tableau de comparaison ordinateur de Von Neumann et cerveau	79
4.1	Tableau de comparaison des méthodes de correspondance.	119
4.2	Résultats de l'apprentissage avec les blocs de taille 8×8	135
4.3	Résultats de l'apprentissage avec les blocs de taille 8×8	136
4.4	Résultats de l'apprentissage avec les blocs de taille 8×8	137
4.5	Résultats de l'apprentissage avec les blocs de 16×16	138
4.6	Résultats de l'apprentissage avec les blocs de 16×16	139
4.7	Résultats de l'apprentissage avec les blocs de 16×16	140

Résumé

L'authentification fiable d'individus par empreintes digitales entraîne des faux rejets élevés pour des systèmes construits rigoureusement. En effet, les auteurs préfèrent que le système rejette un pattern à tort lorsque ce dernier ne satisfait pas à un certain nombre de critères de correspondance préétablis. Dans ce travail, après avoir discuter des techniques existantes, la proposition d'un nouvel algorithme pour réduire les taux de faux rejets lors de l'authentification par empreintes digitales, est faite. Cet algorithme permet d'extraire les minuties de l'empreinte digitale avec leurs orientations relatives et de les classer en fonction des différentes classes déjà établies ([74]); puis d'effectuer la correspondance entre deux gabarits par simples calculs de probabilités à partir des réseaux de neurones à apprentissage profond. La fusion de ces opérations permet d'obtenir des résultats très prometteurs aussi bien sur la base de référence internationale NIST 4 [163], que sur celle des bases de données construites localement (BDAL, SOCOFinger).

Mots clés : Authentification, empreintes digitales, réseaux de neurones, reconnaissance de motifs, apprentissage profond, algorithme.

Abstract

Faultless authentication of individuals by fingerprints results in high false rejections for rigorously built systems. Indeed, the authors prefer that the system erroneously rejects a pattern when it does not meet a number of predetermined correspondence criteria. In this work, after discussing existing techniques, we propose a new algorithm to reduce false rejections during the authentication by fingerprint. This algorithm extracts the minutiae of the fingerprint with their relative orientations and classifies them according to the different classes already established ([74]); then, perform the correspondence between two templates by simple probability calculations from deep learning neural networks. The merging of these operations provides very promising results both on the international database NIST 4 [163], and on a locally constructed database (BDAL, SOCOFinger).

Keywords : authentication, fingerprint, neural networks, pattern recognition, deep learning, algorithm.

Publications

- o S. Kouamo, C. Tangha, "*Handwritten character recognition with artificial neural network*". Distributed Computing and Artificial Intelligence. Advances in Intelligent and Soft Computing. *Vol.*, 151.pp535 – 543. Springer Verlag 2012.
- o S. Kouamo, C. Tangha, "*Image compression with artificial neural networks*". Advances in Intelligent Systems and Computing. *Vol.*, 189.pp515 – 524. Springer Verlag 2013.
- o S. Kouamo, C. Tangha, "*Fingerprint Recognition with Artificial Neural Networks : Application to E-Learning*". Journal of Intelligent Learning Systems and Applications *Vol.8 N°.2*, Pub. Date : May 30, 2016, *DOI* : 10.4236/jilsa.2016.82004.
- o S. Kouamo, C. Tangha, O. Kouamo, "*Reduction of false rejection in an authentication system by fingerprint with deep neural networks*". Journal of Software Engineering and Applications. 13, 1–13, march 2020. *DOI* : 10.4236/jsea.2020.131001.

Introduction Générale

Contexte et justification

Les récents progrès de l'Informatique et des réseaux de communication ont favorisé entre autres : le besoin de se former à distance tout en pratiquant une autre activité, l'accès à distance à des informations sensibles et le développement des *systemes biométriques* durant ces dernières décennies. Ces systèmes sont de plus en plus présents dans les applications liées à *l'identification* et/ou *l'authentification* des individus.

Les méthodes usuelles d'identification/authentification basées sur ce que l'on possède (carte à puce, carte d'identité, badge magnétique, etc.) ou sur ce que l'on sait (mot de passe, code crypté, etc.) posent de gros problèmes de fiabilité [5, 106, 112]. Depuis les récents actes terroristes et les menaces qui pèsent sur de nombreux pays, couplés à l'évolution grandissante de la cyber criminalité dans un monde où toutes les tâches tendent à devenir automatiques, *l'identification et/ou l'authentification* fiable des personnes physiques est devenue un problème crucial pour des raisons de sécurité (contrôle aux frontières, accès aux sites publics ou virtuels, transport, ...).

Tous ces problèmes ont favorisé le développement accru des techniques biométriques comme outil d'identification et d'authentification. L'utilisation des empreintes digitales est l'une des méthodes d'identification/authentification biométrique la plus ancienne et la plus aboutie [112]. S'agissant d'une empreinte digitale, C'est le Britannique *F. Galton* qui prouva en premier l'existence des *dessins papillaires* de la naissance à la mort [58]; cet arrangement particulier des *lignes papillaires* forme des points appelés *minuties* qui sont à l'origine de l'unicité et de l'inaltérabilité de l'empreinte digitale. C'est donc grâce à cette singularité et immuabilité que les empreintes digitales sont en train de s'imposer depuis quelques décennies dans de nombreux secteurs, comme moyen efficace et fiable de la sécurisation des données. En effet, l'identification/authentification peut être appliquée dans de nombreux domaines (la police, les élections locales, l'éducation, la recherche aérospatiale, la communication, l'agriculture, la géo-localisation, ...) de diverses manières, le but final étant toujours le souci de sécuriser l'accès aux informations ou à un environ-

nement bien déterminé. La biométrie en général et l'utilisation des empreintes digitales en particulier apparaît comme l'un des moyens infailibles d'identification des individus depuis quelques années, dans divers secteurs d'activité [169]. En guise d'exemple, le besoin de se former tout en exerçant une autre activité, relié à l'évolution de la technologie susmentionnée a poussé de nombreuses personnes à se tourner vers la formation en ligne (ou E-learning). Il existe aujourd'hui plusieurs pôles physiques de formation en ligne, qui ne sont pas toujours accessibles aux apprenants qui disposent parfois de l'outil nécessaire (tablette numérique, Smartphone, Ordinateur portable, etc.) et ne souhaitent pas quitter leur lieu d'habitation parce qu'exerçant une activité parallèle. Les techniques usuelles d'authentification ayant démontré leurs limites en ce qui concerne la fiabilité, l'utilisation de la biométrie en général et des empreintes digitales en particulier apparaît comme un début de solution à ce problème.

La vérification d'empreintes digitales est un problème plus facile à résoudre et plus pertinent que la reconnaissance, surtout lorsqu'il s'agit des demandes de sécurité et du marché électronique grand public. La mise en application de la loi par exemple, est intéressée par la reconnaissance rapide de l'empreinte digitale d'un individu dans une grande base de données de crimes. Et donc l'ensemble de données d'empreintes digitales peut être très utile dans ce domaine. Les applications de sécurité, d'un autre côté, sont principalement concernées par la vérification d'un individu avec pour objectif l'autorisation d'accès à un système. Dans ce scénario, la personne a déjà fait état de son identité et l'empreinte digitale biométrique est utilisée juste pour confirmer cette prétention ; c'est le cas pour les applications d'authentification pour l'accès à un système de formation en ligne, qui constitue la base d'application de notre travail.

Problématique

Une empreinte digitale dispose en moyenne d'une centaine de points papillaires, mais il faut au moins 12 à 30 de ces points sans discordance pour qu'on estime établie l'identification ou l'authentification [5, 78, 124]. Un chiffre inférieur au seuil minimum aboutit à l'exclusion de l'empreinte digitale comme élément de preuve. Plusieurs techniques sont utilisées pour l'identification et/ou l'authentification des individus par les empreintes digitales. Les méthodes utilisant les réseaux de neurones semblent être très prometteuses, pour plusieurs raisons.

Tout d'abord, les empreintes digitales forment une classe très spécifique de modèles avec une particularité singulière et des caractéristiques statistiques avérées. Ainsi, les problèmes de reconnaissance d'empreintes digitales semblent être bien plus contraignants que les autres problèmes classiques de reconnaissance de formes (tels que la reconnaissance de caractères manuscrits) où les réseaux de neurones ont déjà été appliqués avec

succès [101, 93, 168].

Deuxièmement, les réseaux de neurones permettent d'éviter certains des pièges inhérents à d'autres approches plus conventionnelles. On a connu pendant des siècles que les paires d'images d'empreintes digitales peuvent être associées par des opérateurs humains sur la base du détail des minuties et de leurs orientations relatives sur la surface du doigt [28, 116]. Or il est évident que la même empreinte digitale peut être capturée plusieurs fois avec le même appareil et fournir plusieurs images différentes. Les réseaux de neurones permettent de produire grâce aux connexions synaptiques qu'ils offrent, des modèles qui sont invariants à la translation et à la rotation des stries d'une empreinte digitale [149].

Troisièmement, les réseaux de neurones sont robustes, adaptatifs et traitent des exemples afin de s'adapter aux données. Ceci est particulièrement important puisque les images d'empreintes digitales peuvent inclure plusieurs sources différentes de déformation, propageant ainsi le bruit qui peut provenir de la qualité du dispositif de collecte, de la pression exercée lors de la capture de l'image d'empreinte ou de la saleté présente sur le doigt au moment du scannage. Par ailleurs, il est important d'observer que les exigences en termes de vitesse d'exécution, de puissance de calcul, de probabilité de faux rejet et de fausse acceptation, de capacité de stockage et de taille de la base de données peuvent varier considérablement selon le domaine [11, 15, 21]. Avoir accès à une résidence privée ou à un système de formation en ligne par exemple, implique un petit système économique avec une petite base de données modifiable de quelques personnes et un temps de réponse de quelques secondes. Dans le même ordre d'idée, des applications médico-légales peuvent exiger des recherches rapides par de très grandes bases de données de millions d'enregistrements utilisant de grands ordinateurs et un temps de réponse qui peut être très long. Les réseaux de neurones peuvent être ajustés et formés différemment pour adapter l'exigence particulière des demandes (d'applications) spécifiques [109].

Certains travaux de recherche proposent des solutions aux problèmes d'identification/authentification d'individus par les réseaux de neurones [11, 149, 172]. Deux philosophies majeures se dégagent de ces travaux :

- la première pensée propose d'effectuer la reconnaissance par classification. Les empreintes digitales qui ont des gabarits qui s'apparentent sont alors rangées dans des sous classes prédéfinies. Plusieurs classifications ont été réalisées [58], mais de nos jours, les classes les plus utilisées sont celles définies par *Henry* [74], à savoir : les boucles (à gauche et à droite), les tourbillons, les arches et les arches tentées ;
- la deuxième philosophie quant à elle propose d'effectuer la reconnaissance en se

basant sur d'autres éléments plus subtiles tels que : les minuties, l'orientation des stries, etc. Ce n'est donc plus l'image entière qui est prise en compte, mais un ensemble basé sur des éléments qui vont permettre de faire une analyse plus fine de l'image : le *gabarit*.

La deuxième philosophie est mieux adaptée quand le concepteur utilise une base de données et souhaite mettre en avant la fiabilité des résultats obtenus. Seulement, un compromis doit être fait entre le taux de reconnaissance et celui de faux rejets. En effet, plus le système est robuste mieux sont les résultats en terme de taux de reconnaissance ; mais en contrepartie les taux de faux rejets ont tendance à être élevés lorsque le système est conçu pour l'authentification. Ceci s'explique par le fait que lorsque le système est conçu pour l'authentification, il est beaucoup plus strict lors de la vérification et tend à éliminer une image d'empreinte qui ne respecte pas tous les critères de correspondance. Le système préférera donc rejeter une image dont la correspondance a été effectuée en dessous d'un certain seuil (assez élevé) au lieu de prendre le risque d'accepter une fausse image. Il serait donc intéressant de proposer un système robuste qui produit des taux de faux rejets acceptables (moins de 5%) tout en maintenant les taux de reconnaissance intéressants (plus de 90%).

Les questions abordées dans cette thèse sont les suivantes : Quels sont les taux d'erreur (taux de fausses acceptations, taux de faux rejets) obtenus en fonction du choix de l'algorithme utilisé ? Quel est le nombre de minuties obtenu en moyenne pour un seuil donné ? Comment choisir les données d'apprentissage et de test ? Quel est l'intérêt d'effectuer la comparaison sur des données traitées ou non ? comment concevoir un algorithme efficace et fiable permettant de réduire les faux rejets dans le processus d'authentification par empreintes digitales ?

Idée sous-jacente

Ce travail porte sur un système pour l'authentification, qui est une comparaison de type "*un contre un*" (voir 2.1.2). Ce genre de système a tendance à être plus rigoureux et, donc préfère rejeter un pattern qui ne présente pas tous les critères requis, même si ce dernier appartient à la classe correct (ce qui implique des taux faux rejets élevés). Notre idée est de proposer un algorithme qui produit de bons résultats (par rapport à ceux existants), tout en évitant de rejeter un pattern de la classe correcte. Ce qui implique agir sur la structure interne du système afin qu'il rejette le moins possible un pattern à tort.

Contribution

Cette thèse contribue à la résolution des problèmes sus-évoqués : offrir un système permettant d'effectuer la correspondance de deux empreintes digitales quelconque avec un taux de fiabilité élevé et un nombre de minutie minimum acceptable, ceci dans un délai de temps raisonnable.

Ce travail propose un nouvel algorithme d'extraction et de classification des minuties par une approche parcimonieuse ainsi que la structure du réseau de neurones à apprentissage profond appropriée. Ce nouvel algorithme s'exécute spécifiquement en deux niveaux (ce qui constitue notre principale contribution et sera présentée plus en détail à la section 4.3) :

- le premier niveau qui consiste en la classification de l'empreinte digitale passée en entrée du réseau en considérant sa topographie générale suivant la *classification d'Henry* [74] se décline en deux étapes : la première étape permet de classifier l'empreinte digitale entrée en la comparant aux prototypes de la première classe de données construites, par simple calcul de probabilités, exactement comme proposé par *Baldi* [11]; la seconde étape quant à elle, sert de contrôle de la première classification et permet de réduire le risque de réponses erronées. Ce deuxième contrôle contribue à réduire le taux de faux rejet, y compris lorsque le nombre de minuties détectées est relativement bas.
- le deuxième niveau qui est en fait consacré à la correspondance ou authentification, consiste à comparer lors de la phase d'entraînement du réseau, le gabarit (invariant à la translation et la rotation) de l'empreinte digitale en entrée à celui des empreintes appartenant à la classe vainqueur lors des deux classifications précédentes, ceci par une simple mesure de similarité (la *norme euclidienne* par exemple).

Plan du document

Cette thèse est structurée en quatre chapitres. Le premier chapitre, introduit la notion de biométrie et les applications qui en découlent. Le deuxième chapitre présentera les principes fondamentaux de la reconnaissance d'empreinte digitale, et parlera des opérations de pré-traitement algorithmique permettant d'améliorer la qualité d'une image d'empreinte digitale et des techniques d'extraction de minuties. Le troisième chapitre est consacré à une introduction aux réseaux de neurones. Le quatrième chapitre quant à lui présentera les apports des approches neuronales dans le domaine de la reconnaissance d'empreinte digitale et exposera les résultats expérimentaux des tests effectués sur les différentes techniques étudiées y compris celle proposée. Elle s'achève par une conclusion

et des perspectives.

La Biométrie et les empreintes digitales

1.1 Introduction

Ce chapitre présente la biométrie de manière générale ainsi que les diverses applications qui en découlent, en insistant sur l'utilisation des empreintes digitales. Puis il détaille les différentes étapes composant un système complet de reconnaissance d'empreintes digitales (ou Automated Fingerprint Identification System : *AFIS*).

1.1.1 Qu'est ce que la Biométrie ?

La biométrie se réfère à l'identification automatique d'une personne vivante grâce à ses caractéristiques physiologiques ou comportementales. Il existe différents types de technologies biométriques sur le marché [28, 109] :

- la reconnaissance faciale,
- la reconnaissance d'empreintes digitales,
- l'identification de la géométrie des doigts et de la main,
- l'identification de l'iris de l'œil,
- l'identification de la structure des veines,
- l'identification de la voix,
- l'identification de la signature,
- les autres techniques (démarche, l'oreille, l'ADN, l'odeur, ...).

La méthode d'identification biométrique est préférée aux méthodes traditionnelles impliquant des mots de passe ou des codes cryptés, pour différentes raisons dont la principale est que la personne à identifier doit être physiquement présente au point d'identification. L'augmentation rapide de l'utilisation des mots de passe, codes cryptés et autres techniques, due à la révolution des technologies de l'information, amène à devoir limiter l'accès aux données confidentielles et personnelles. En remplaçant les codes

et les mots de passe, les *techniques biométriques* sont plus commodes pour l'utilisateur et peuvent potentiellement empêcher l'accès non autorisé ou l'utilisation frauduleuse de distributeurs de billets, de systèmes de gestion des horaires, de téléphones portables, de cartes à puce, de postes de travail et de réseaux informatiques [16]. Les codes cryptés et les mots de passe peuvent être oubliés, ainsi que les autres méthodes usuelles d'identification, comme les passeports, les permis de conduire et les cartes d'assurance, qui peuvent aussi être volés ou perdus. Les techniques biométriques apportent donc des mesures palliatives à tous ces problèmes d'identification et d'authentification [112, 58, 169], comme en témoigne l'étude des perspectives du marché de la biométrie (Figure 1.1) qui montre une évolution annuelle sans cesse croissante des revenus de l'industrie de la biométrie dans le monde de 2009 à 2019.

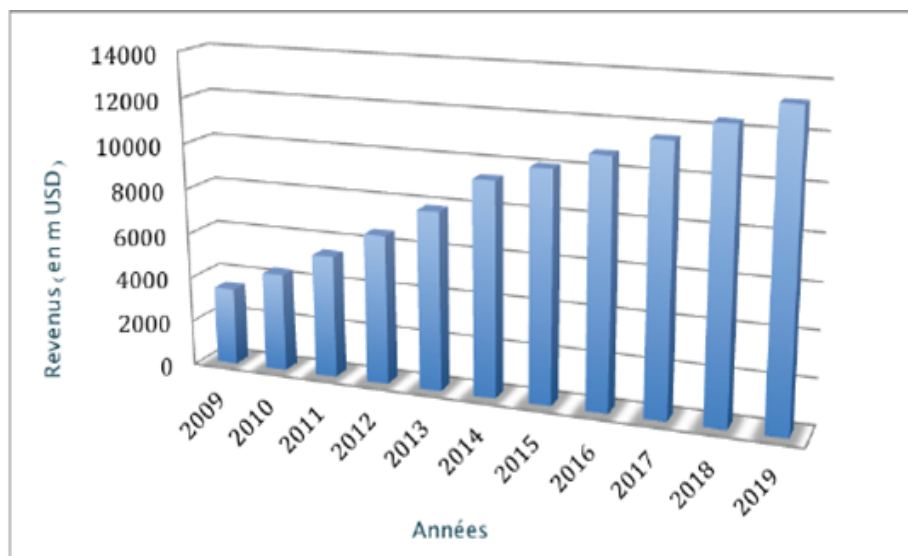


FIGURE 1.1 – Revenus annuels de la biométrie en millions de Dollars US de 2009 à 2019 (Copyright ©2019 International Biometric Group)

La reconnaissance basée sur les techniques biométriques évite ainsi la nécessité de se rappeler d'un mot de passe, de se servir d'un badge de proximité ou d'une carte magnétique.

La biométrie peut encore être définie comme étant une technique globale visant à établir l'identité d'une personne en mesurant une de ses caractéristiques physiques. Il peut y avoir plusieurs types de caractéristiques physiques, les unes plus fiables que les autres, mais toutes doivent être infalsifiables et uniques pour pouvoir être représentatives d'un et un seul individu. D'autre part, comme nous allons le voir dans les sections qui suivent, les caractéristiques physiques sont loin d'être parfaites et précises et, l'on

atteint très vite des limites pour ces techniques.

Il existe deux types de systèmes de reconnaissance biométrique : ceux basés sur la *vérification* et ceux basés sur *l'identification* [109, 120]. La vérification, également appelée *authentication*, consiste à confirmer ou infirmer l'identité d'une personne (*suis-je celui que je prétend être ?*). Il s'agit d'une comparaison du type "*un-contre-un*"; les caractéristiques de l'individu sont comparées à celles présentes dans un enregistrement de référence. C'est le cas pour l'authentification d'un individu qui souhaite accéder à un système de formation en ligne.

Quant à l'identification elle permet d'établir l'identité d'une personne (*qui suis-je ?*) à partir d'une base de données, il s'agit d'une comparaison du type "*un-contre-plusieurs*". L'application qui vient immédiatement à l'esprit est l'utilisation des empreintes digitales par les forces de l'ordre pour fichier les criminels. A titre d'exemple le *FBI* (Federal Bureau of Investigation) Américain, posséderait une base de données de 250 millions d'empreintes digitales [106, 109].

Outre l'utilisation policière, la biométrie apporte des solutions dans de nombreuses applications telles que : le contrôle d'accès à un réseau local pour éviter la fraude par mot de passe (cloud computing, réseau Intranet, ...), le contrôle d'accès à un lieu physique (aéroport, gare, casino, ...), la surveillance (systèmes de vidéosurveillance des rues, contrôle des détenus dans les prisons, ...), le paiement en ligne (pour sécuriser la transaction), etc.

1.1.2 Les différentes techniques biométriques

Les différentes techniques biométriques existantes se distinguent en trois grandes catégories :

- l'analyse morphologique dans laquelle on retrouve : les empreintes digitales, l'iris de l'œil, la forme de la main, les traits du visage, le réseau veineux de la rétine ;
- l'analyse des traces biologiques qui concerne : l'*ADN*, le sang, la salive, l'urine, l'odeur, la thermographie¹ ;
- l'analyse comportementale qui est constituée de : la reconnaissance vocale, la dynamique des frappes au clavier, la dynamique de signature, la manière de marcher.

Parmi les nombreuses méthodes d'identification biométrique, l'utilisation des empreintes digitales est la méthode la plus aboutie avec près de 48% de parts de marché en 2013, comme le montre la Figure 1.2 (la section suivante y reviendra plus en détail) [9].

1. une technique permettant d'obtenir une image thermique d'une scène par analyse des infrarouges

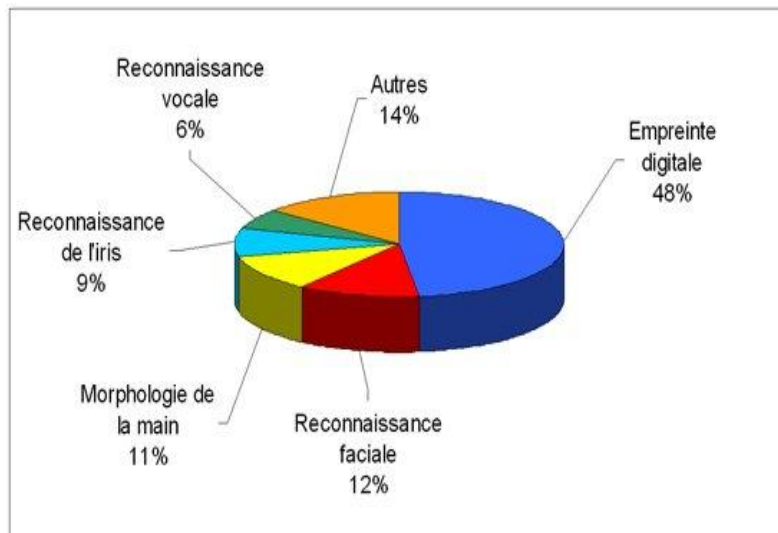


FIGURE 1.2 – Part des marchés des différentes techniques biométriques

Néanmoins d'autres techniques commencent à trouver leur place sur le marché de la biométrie (voir Figure 1.3). Il s'agit de : la reconnaissance de la forme du visage, l'iris de l'œil, la géométrie de la main ou du doigt, la reconnaissance vocale et la dynamique du tracé de la signature.

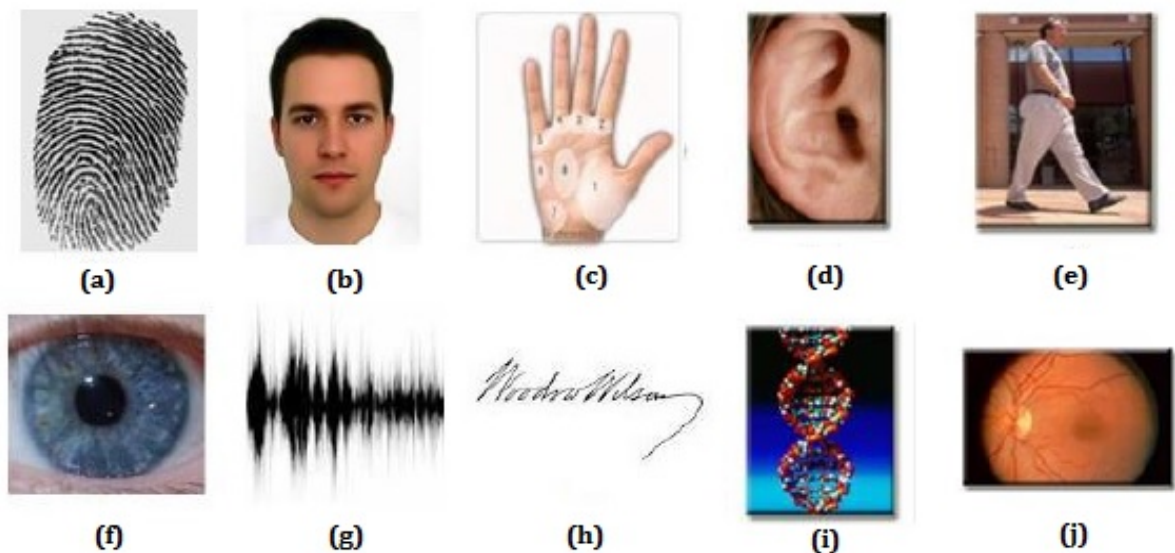


FIGURE 1.3 – Exemples de quelques caractéristiques biométriques : (a) Empreinte digitale, (b) visage, (c) main, (d) oreille, (e) démarche, (f) iris de l'œil, (g) empreinte vocale, (h) signature, (i) ADN, (j) rétine.

1.1.2.1 La forme du visage

C'est la technique la plus simple et la moins contraignante. Mais elle a encore de gros progrès à faire. En mode vidéo, les résultats ne sont pour l'instant pas très probants.

Principe

Il s'agit ici de faire une photographie plus ou moins évoluée pour en extraire un ensemble de facteurs qui se veulent propres à chaque individu [26, 9, 174]. Une webcam capture l'image du visage présenté, et l'envoie à un logiciel pour la numériser. Le logiciel repère d'abord la position des yeux pour procéder à un "alignement". En fonction de cet alignement, un relevé de différents points caractéristiques du visage est effectué (ailes du nez, forme du menton, écartement des yeux, ...). Un tracé géométrique personnel est alors enregistré comme gabarit (chaque visage est codé sous forme de fichier de 84 octets), et c'est sur ce dernier que s'effectueront ensuite les recherches [3, 10].

Une des technologies les plus utilisées est celle de "l'*Eigenface*" [152, 155], développée par le *Massachusetts Institute of Technology (MIT)* à partir d'analyses statistiques de milliers de visages. Elle consiste à décomposer le visage en plusieurs images faites de nuances de gris, chacune mettant en évidence une caractéristique particulière. Le système des "réseaux de neurones" sont encore plus performants, surtout dans des conditions difficiles de capture. Ils emploient un autre algorithme pour déterminer la similitude entre des captures d'images de visage et des gabarits. Pour plus de sécurité lors du recoupement, on utilise plusieurs algorithmes de recherche et on signale plusieurs personnes susceptibles de correspondre à la requête. Certains systèmes sont capables de comparer jusqu'à 60 millions de visages par minute [174].

Avantages

Le principal avantage, c'est que cette technologie est non intrusive, et peut même se dérouler à l'insu de l'utilisateur pour la vidéo. Les progrès accomplis ces dernières années sont considérables. Au point que les systèmes sont aujourd'hui capables de reconnaître des individus même avec des artifices (fausses moustaches, barbe, lunettes). "*Seule une chirurgie importante du cartilage du visage pourrait tromper le système*" affirme ainsi la société *Zalix*, qui vend une solution de reconnaissance faciale.

Limites

En position fixe et éclairée, les taux de reconnaissance sont effectivement très élevés, mais en vidéo la technique n'est pas encore au point. Même en imaginant que la technique s'améliore, le visage est trop changeant pour être un repère biométrique suffisamment

fiable. Une étude américaine a par exemple montré qu'un changement d'angle de 45° de la caméra rendait le système quasi inutilisable. Du coup, plusieurs aéroports ont abandonné la technique.

Applications

La technologie de reconnaissance faciale est aujourd'hui en pratique dans plusieurs casinos dans le monde (identification des joueurs interdits ou individus fichés), aéroports, stades (refoulement de supporters connus et dangereux), centres commerciaux ou grands magasins.

Mais pour des usages à grande échelle, la reconnaissance faciale est le plus souvent associée à une autre technique biométrique. A titre d'exemple : le gouvernement israélien utilise un système combiné "*forme de la main/ reconnaissance faciale*" pour filtrer les palestiniens en provenance de la *bande de Gaza*. En *Ouganda*, les élections se déroulent à partir d'un fichier de "*visages*" d'électeurs.

1.1.2.2 La géométrie de la main/du doigt (*hand – scan*)

Principe

Ce type de mesure biométrique est l'un des plus répandus, notamment aux *Etats – Unis*. Cantines scolaires, entreprises, clubs de sport, ... De plus en plus de clients se tournent vers cette technologie, facile à utiliser et peu coûteuse. Ce système fait appel à la forme de la main : longueur et épaisseur des doigts, largeur de la paume, forme des articulations. En tout, 90 points caractéristiques sont nécessaires pour identifier une personne [48, 82]. En général, une caméra infrarouge prend l'image de la main sous deux angles différents pour obtenir les trois dimensions. L'image scannée est convertie en modèle numérique et associée à un code. Quand une personne passe son badge devant le lecteur, celle-ci cherche le code dans la base de données et procède à la comparaison.

Comme pour les empreintes digitales, on vérifie la validité de la mesure par la conductivité ou la température de la main. On peut aussi l'associer à l'empreinte du réseau veineux.

Avantages

Simple à mettre en œuvre, peu intrusive, cette technologie est appréciée des utilisateurs. Les images numérisées sont peu volumineuses, comparée à celles de l'empreinte digitale (10 à 20 octets contre 250 à 1000 octets).

Limites

Le taux d'erreurs dans la reconnaissance en utilisant cette technique est assez élevé, en particulier pour des personnes appartenant à une même famille en raison d'une forte ressemblance. De plus, la forme de la main est moins stable dans le temps que les empreintes digitales. Des déformations importantes des doigts peuvent en effet survenir avec l'âge (*arthrite*² par exemple). Le scanner est plus encombrant que pour les empreintes digitales, ce qui rend la technologie inaccessible aux systèmes portatifs.

Applications

Moins associée à la police que les empreintes digitales, cette technologie est particulièrement appréciée dans les cas où aucun motif de sécurité n'est nécessaire (cantines scolaires, pointeuses dans les entreprises, etc.). *Coca – Cola*, *Pfizer*, ou *TWA* par exemple l'utilisent déjà. Lors des Jeux Olympiques de 1996, c'est la solution qui avait été retenue pour accéder au village olympique. C'est aussi la solution qu'a retenu le service américain des migrations pour identifier les travailleurs frontaliers, qui effectuent des allers-retours fréquents entre les *Etats – Unis* et les autres pays. Au Musée de *Louvre*, l'accès à certaines salles est réservé pour le nettoyage ; l'ouverture se fait par identification palmaire. Et avec la baisse des coûts, la biométrie palmaire s'implante de plus en plus dans les quartiers résidentiels privés.

1.1.2.3 L'iris de l'œil

Particulièrement fiable, cette technique est considérée comme impossible à frauder. Elle est toutefois plus lourde à mettre en œuvre que les empreintes digitales.

Principe

Il faut tout d'abord faire la distinction entre l'iris et la rétine de l'œil. L'étude de l'iris se porte sur la partie visible de l'œil ; l'individu se place en face d'un capteur (par exemple : une caméra *CCD/CMOS*) qui scanne son iris. L'iris est la membrane colorée de l'œil, elle est constituée d'un réseau de tubes fins dont le diamètre est inférieur à celui d'un cheveu [34, 150]. La forme de l'iris, c'est-à-dire l'enchevêtrement des tubes ne varie que très peu durant la vie de l'individu. L'iris présente une quasi-infinité de points caractéristiques qui ne varient pratiquement pas pendant la vie d'une personne.

Une caméra proche des infrarouges "*photographie*" une tranche de l'iris. Elle relève les caractéristiques particulières du relief (sillons de contraction, anneaux, etc). On peut

2. Inflammation d'une articulation

distinguer jusqu'à 244 points de comparaison, et le taux d'erreur des produits disponibles sur le marché est quasi nul (la probabilité de trouver 2 iris identiques serait de 1 sur 1072) [34]. D'autant plus qu'on peut faire doubler la simple photographie de certaines caractéristiques dynamiques de l'œil : réactivité de la pupille (dilatation/rétraction) par rapport à la quantité de lumière, étude de l'iris dans l'infrarouge et l'ultraviolet, etc. La caméra doit capturer au moins 70 pixels de rayon sur l'iris pour exploiter l'image.

Avantages

La texture de l'iris est parfaitement stable au cours du temps et les derniers progrès de l'optique ont permis d'alléger les procédures. Selon *Sagem*, la vérification de l'identité prend aujourd'hui moins de 4 secondes [38, 8]. La technique reste extrêmement fiable même à travers des lunettes ou des lentilles.

Limites

La prise de vue n'est pas très simple : la taille de l'iris est très variable suivant la lumière ambiante ou l'état de fatigue, et les utilisateurs ont tendance à bouger. Par ailleurs, la fiabilité diminue proportionnellement à la distance entre l'œil et la caméra. L'infrastructure est plus lourde que pour les empreintes digitales (compter 12 000 euros soit 7 800 000 FCFA pour un équipement de base).

Applications

Cette technique est très utilisée en Asie, mais assez peu dans le reste du monde. La biométrie par l'iris a été freinée en Europe par le fait que la majeure partie des algorithmes mathématiques de modélisation et de reconnaissance de l'iris sont brevetés aux Etats-Unis (c'est *John Daugman*, alors enseignant à l'Université de Harvard, qui a déposé les brevets en 1994) [38]. Depuis 2003, le Canada a mis en place un contrôle douanier facilité pour les passagers volontaires. Ces derniers enregistrent l'image de leur iris sur une carte, et évitent ainsi les tracasseries douanières habituelles. Le HCR (Haut-Commissariat aux Réfugiés) a utilisé en 2003 des systèmes biométriques en Afghanistan pour l'authentification des réfugiés ; après authentification, le réfugié recevait à son retour des coupons d'aide alimentaire. L'Australie devrait elle aussi bientôt recourir à l'identification par l'iris pour le contrôle de ses frontières.

1.1.2.4 Le réseau veineux

Relativement nouvelle, cette méthode semble séduire. Elle se profile toutefois comme complément, par exemple à la géométrie de la main ou à l'empreinte digitale.

Principe

Le sang chargé en oxygène arrive dans les mains par les artères, puis repart vers le cœur par les veines. Le sang appauvri en oxygène n'a pas le même filtre d'absorption que le sang artériel : il absorbe la lumière à des longueurs d'onde proches de l'infrarouge (autour de 760 micromètres) [41, 26]. Quand la main est éclairée avec une lumière infrarouge, le réseau veineux apparaît en noir. Il est enregistré sous forme de "*carte d'identité*" dans une base de données, et pourra ensuite servir de comparaison lors de l'authentification.

Il existe deux types de "*photographie*" : la méthode "*réflective*" et la méthode par transmission. La méthode réflective est plus fiable que la méthode par transmission (capture de la lumière qui passe à travers la main). En effet, quand il fait froid, les veines se contractent et le contraste n'est pas assez visible par transmission. Par contre, la lumière reflétée n'est pas modifiée, et peut utiliser la méthode réflective même par temps froid. Par ailleurs, les deux parties du dispositif sont au même endroit, puisque la lumière est renvoyée [82]. C'est donc plus facile à intégrer dans un petit appareil, par exemple un téléphone portable.

Avantages

Il n'existe aucun moyen de frauder, car on ne peut pas "*photographier*" les veines de la main. Le réseau vasculaire est propre à chaque individu : même les vrais jumeaux en ont un différent. Selon *Fujitsu*, qui a comparé 140 000 paumes de main en 2014, le taux de fausses acceptations (personnes authentifiées comme quelqu'un d'autre) est de moins de 0,00008%. Il n'y a aucun contact lors de l'authentification comme avec les empreintes digitales, donc pas de problème d'hygiène.

Limites

La méthode est encore trop récente pour être correctement évaluée. Le scanner est relativement encombrant comparé aux capteurs d'empreintes digitales.

Applications

Fujitsu a développé une technologie de reconnaissance veineuse, le "*Palm Vein Pattern*" [79]. Le système intéresse en premier lieu les établissements bancaires. Le réseau veineux du client est enregistré soit dans une base de données, soit dans une puce sur la carte bancaire. La banque japonaise de *Tokyo-Mitsubishi* distribue déjà des cartes biométriques à ses clients, qui peuvent ainsi retirer ou déposer de l'argent dans toutes les agences équipées d'un scanner.

Le système de *Fujitsu* sert aussi au contrôle d'accès dans des immeubles, pour l'enregistrement des bagages dans les aéroports, ou pour mettre en marche des machines sensibles dans des usines par exemple [79]. En Corée du Sud, la police, l'armée, les services de réanimation et plusieurs chantiers utilisent un système concurrent, celui de la société *Techsphere*.

1.1.2.5 *La dynamique du tracé de la signature*

Principe

Il s'agit d'une analyse comportementale où différents éléments (mesure de la vitesse, ordre d'écriture, pression exercée, accélérations) sont mesurés lors de la signature [9, 126, 164].

Ce type de biométrie est à l'heure actuelle peu utilisé mais ses défenseurs espèrent l'imposer assez rapidement pour des applications spécifiques, à savoir : les documents électroniques, les rapports, les contrats,...

Limites

La principale limite repose sur le fait que la falsification est possible en passant par une phase d'apprentissage. L'auteur peut alors voir sa signature imitée par un tiers qui a pris le soin préalable d'apprendre minutieusement à signer exactement comme lui. De plus, la signature peut varier selon le stress de l'utilisateur.

1.1.2.6 *Les autres techniques*

Elles sont constituées de la reconnaissance vocale, la rétine, la démarche, l'oreille, l'ADN, les ongles, l'odeur, la thermographie et la dynamique des frappes au clavier [126, 164].

i) *La reconnaissance vocale*

Le principe général de cette technique consiste à analyser les caractéristiques du timbre de la voix et de la prononciation. La qualité de l'enregistrement peut poser problème et il est possible de frauder avec un échantillon vocal pré-enregistré [164]. Les données utilisées par la reconnaissance vocale proviennent à la fois de facteurs physiologiques et comportementaux. Ils ne sont en général pas imitables.

ii) *La rétine (retina – scan)***Principe**

Cette mesure biométrique est plus ancienne que celle utilisant l'iris, mais elle a été moins bien acceptée par le public et les utilisateurs, sans doute à cause de son caractère trop contraignant. En effet, la mesure doit s'effectuer à très faible distance du capteur (quelques centimètres), qui effectue ensuite un balayage de la rétine. Il est physiquement impossible d'effectuer une mesure rétinienne à une distance de 30 *cm* ou plus sur un sujet mobile comme on peut le voir dans certains films (cette méthode requiert des sujets coopératifs et entraînés). Pourtant cette technique semble être tout aussi fiable que celle de l'iris ; elle se base sur le fait que le schéma et le dessin formé par les vaisseaux sanguins de la rétine (la paroi interne et opposée de l'œil) sont uniques pour chaque individu, différents entre jumeaux et assez stables durant la vie de la personne. La mesure peut ainsi fournir jusqu'à 400 points caractéristiques du sujet, que l'on peut comparer aux 30 à 40 points fournis par une empreinte digitale [41, 110].

En conclusion, la mesure rétinienne est la plus difficile à utiliser mais également la plus difficile à contrefaire.

iii) *La démarche*

Il s'agit de reconnaître un individu par sa façon de marcher et de bouger (vitesse, accélération, mouvements du corps), en analysant des séquences d'images. La démarche serait en effet étroitement associée à la musculature naturelle et donc très personnelle. Mais des vêtements amples, par exemple, peuvent compromettre une bonne identification.

iv) *L'oreille*

A priori, la technique serait efficace, car il n'existe pas deux formes d'oreilles identiques. Mais il n'existe encore aucune application commerciale.

v) *La voix*

Un spectrogramme³ relève la tonalité, le rythme, l'intensité et la fréquence de la voix. Réputée peu fiable, c'est quand même la seule technique qui permet une authentification à distance. Elle ne peut cependant pas constituer une preuve suffisante pour la justice.

3. Représentation graphique ou photographique du spectre d'un rayonnement lumineux

vi) L'ADN

L'empreinte génétique est la marque biologique la plus sûre du monde. Dans le cas des tests de paternité, on atteint une fiabilité de 99,999%. Mais les analyses d'ADN nécessitent des délais de plusieurs semaines, ce qui interdit toutes les applications d'identification en temps réel.

vii) Les ongles

La technique est basée sur les stries longitudinales des ongles, qui dépendent de la structure de l'épiderme sous-jacent. On peut révéler le relief de l'ongle grâce à un interféromètre⁴, et le cartographier.

viii) L'odeur

Ici, le vivant dépasse la machine. Même si des "*nez électroniques*" existent, le chien reste le plus performant pour détecter et identifier les odeurs individuelles. Mais peu de personnes sont prêtes à accepter de se faire renifler tous les matins! Du coup, la technique est surtout utilisée dans l'agroalimentaire.

ix) La dynamique des frappes au clavier (*keystroke – scan*)

Les frappes au clavier sont influencées par plusieurs choses; tout d'abord, selon le texte que l'on tape et de manière plus générale selon sa nature, on aura tendance à modifier sa façon de taper au clavier. C'est d'ailleurs un des moyens utilisés par certaines attaques (*timing attacks*) pour essayer d'inférer le contenu ou la nature du texte tapé de façon à remonter jusqu'à un mot de passe par exemple [61].

Ces techniques sont assez satisfaisantes mais restent néanmoins statistiques. Ensuite, le facteur comportemental entre en jeu et sera différent pour chaque individu. Les facteurs sont à peu de chose près identiques à ceux évoqués précédemment : ce sont les durées entre frappes, la fréquence des erreurs, la durée de la frappe elle-même,... La différence se situe plus au niveau de l'analyse, qui peut être soit statique et basée sur des réseaux neuronaux, soit dynamique et statistique (comparaison continue entre l'échantillon et la référence).

Certaines de ces techniques sont très prometteuses (*iris*) et commencent à émerger, d'autres sont encore au stade expérimental (*analyse comportementale*). Mais l'utilisation des empreintes digitales reste la méthode la plus aboutie et la plus utilisée actuellement.

4. Méthode de mesure qui exploite les interférences intervenant entre plusieurs ondes cohérentes entre elles.

Le tableau 1.1 ci-après fait état de la comparaison de quelques techniques biométriques suivant plusieurs critères déterminés.

Critère Modalité	Acc	Univ	Uni	Perm	Fenreg	Perf	Infal
ADN	-	+	+	+	-	+	-
Démarche	+	=	-	-	+	-	=
Frappe au clavier	=	-	-	-	=	-	=
Empreinte digitale	=	=	+	+	=	+	+
Géométrie de la main	=	=	=	=	+	=	=
Iris	-	+	+	+	=	+	+
Rétine	-	+	+	=	-	+	+
Signature	+	-	-	-	+	-	+
Visage	+	+	-	=	+	-	-
Voix	+	=	-	-	=	-	-

TABLE 1.1 – Comparaison des systèmes biométriques selon différents critères : (Acc) Acceptabilité, (Univ) Universalité, (Uni) Unicité, (Perm) Permanence, (Fenreg) Facilité d’enregistrement, (Perf) Performance, (Infal) Infalsifiabilité. Notation : (+) fort, (=) moyen, (-) faible

1.1.3 Les champs d’application de la biométrie

Les techniques biométriques sont prisées sur le marché de l’industriel notamment pour le développement des systèmes automatiques de reconnaissance d’individus. Depuis la fin des années 1980, le marché de la biométrie a connu une croissance exponentielle, venant ainsi remplacer les systèmes traditionnels basés sur les mots de passe et les codes PIN. Les applications biométriques sont nombreuses et diversifiées, contribuant à accroître le marché relatif à ces dernières. Ainsi on distingue entre autres domaines d’application :

– **sur le plan industriel :**

- le domaine du développement des équipements électroniques à usage confidentiel (par exemple : ordinateur doté de reconnaissance digitale, tablettes à reconnaissance vocale, etc.);
- le domaine de la finance et des banques (par exemple : appareil doté d’un système de reconnaissance automatique de la signature d’un client, etc.);

- le domaine de la police criminelle et des frontières qui est à l'origine du *passport biométrique* (afin de créer des bases de données internationales pour lutter efficacement contre la montée en puissance du terrorisme et du grand banditisme dans le monde entier) ;
- le domaine de l'aviation civile et militaire ;
- **sur le plan politique :**
 - le domaine de la signature à distance par les chefs d'Etat ou autres autorités administratives, notamment lors de l'authentification automatique à distance ;
 - le domaine de la téléconférence ou visioconférence ;
 - le domaine de la sécurisation d'un accès public/privé (salle de bunker, etc.) ;
- **sur le plan économique et social :**
 - le domaine de l'éducation à distance (par exemple : un système d'authentification d'étudiants à distance par empreinte digitale, ...) ;
 - le domaine de la médecine (très important pour l'ouverture des fichiers médicaux uniques par individus basés sur certaines caractéristiques physiologiques telles que : les empreintes, la disposition des veines de la main, etc.) ;
 - le domaine de l'automobile et de l'immobilier (en mettant en œuvre des systèmes de sécurité biométriques basés sur un très petit nombre d'acteurs afin de limiter les cas de vols et d'actes de criminalité) ;
 - le domaine de la télésurveillance en dotant les salles de surveillance d'un système d'accès à reconnaissance biométrique, afin d'éviter les actes de sabotages ;
 - le domaine électoral en mettant en place des systèmes électoraux basés sur l'authentification biométrique des électeurs locaux et externes ;
 - etc.

1.2 Les empreintes digitales

1.2.1 Historique

La reconnaissance d'empreinte digitale représente la méthode la plus ancienne d'identification biométrique. Son histoire remonte au moins à 6 000 ans avant *Jésus-Christ* (voir dans [116]). Bon nombre de sculptures et poteries archéologiques qui datent de la préhistoire portent des traces d'empreintes digitales humaines. Deux siècles avant *Jésus-Christ*, les Chinois apposaient déjà des empreintes sur les documents importants. Les peuples anciens étaient donc au courant de l'individualité des empreintes digitales,

mais leur connaissance n'avait aucun fondement scientifique. C'est vers la fin du 16^e siècle que les techniques scientifiques modernes ont été initiées dans ce domaine (voir [58, 106]). Cependant, le livre Perse du 15^e siècle intitulé "*Jaamehol-Tawarikh*" qui se traduit "*histoire universelle*", attribué à *Khajeh Rashiduddin Fazlollah* [128], contient des commentaires sur la pratique d'identifier les personnes par leurs empreintes digitales.

En 1684, le morphologiste⁵ anglais *Dr. Nehemiah Grew* publie le premier article [67] détaillé sur les empreintes digitales. Dès lors, de nombreuses études et recherches sur les empreintes digitales sont effectuées. C'est ainsi qu'un an plus tard l'anatomiste néerlandais *Govard Bidloo* publie un article [65] sur l'anatomie du corps humain, dans lequel il met en exergue l'existence de certains traits caractéristiques (crêtes papillaires). Cependant aucune mention n'est faite sur l'unicité et la permanence des stries jusqu'ici.

En 1823, le physiologiste tchèque *Johannes Purkinje* propose la première classification des empreintes digitales en 9 catégories selon la structure des lignes de l'empreinte digitale [127].

En 1863, le Professeur *Jean – Paul Coulier*, du Val-de-Grâce à Paris, publie ses observations sur les empreintes latentes et met en évidence le fait qu'elles peuvent être développées sur du papier en utilisant de l'iode. Il explique également comment préserver les imprimés et mentionne la possibilité d'identification des empreintes digitales (principalement des suspects) par l'utilisation d'une loupe (voir [178]).

En 1880, le médecin écossais *Henry Fauld* confirme la spécificité des empreintes digitales chez chaque individu ; sa conviction est cependant basée sur une observation empirique (voir [176, 179]). Il propose la première méthode d'enregistrement des empreintes digitales : l'impression avec de l'encre d'imprimerie.

En 1888, l'anthropologue anglais *Francis Galton* introduit la notion de minuties⁶ pour réaliser la comparaison d'empreintes digitales (*fingerprints matching* en anglais). Quatre ans plus tard, *Galton* publie son ouvrage "*Fingerprint*" [58] où il propose une classification rigoureuse des empreintes digitales et démontre qu'il y a seulement une chance sur 64 milliards que deux individus aient une même empreinte digitale.

5. La morphologie est une branche de la biologie qui étudie la forme et la structure des organismes.

6. Dans la biométrie des empreintes digitales, les minuties sont des traits caractéristiques d'une empreinte digitale (bifurcation, terminaison, îlot, ...).

A la fin des années 1899, la dactyloscopie⁷ connaît une avancée majeure avec la mise en place d'un nouveau système de classification des empreintes digitales par le chef de la police londonienne *Edward Richard Henry*; ce système très célèbre est connu sous le nom de *Henry system* [74].

Vers la fin du 20^e siècle, la reconnaissance des empreintes digitales est admise comme moyen valide d'identification d'un individu. Les agences de l'empreinte digitale voient le jour partout dans le monde et les premières bases de données d'empreintes digitales sont établies; l'une des premières à être créée est celle de l'agence de l'empreinte digitale du Federal Bureau of Investigation (FBI) en 1924 aux USA, avec 810 000 cartes⁸ d'empreintes digitales (voir [178, 175]). Notons que l'identification était effectuée manuellement par les experts des empreintes digitales. Suite à la croissance explosive des bases de données, ce travail est devenu quasi impossible.

En 1960, le FBI en collaboration avec le département de police de Paris, commence à étudier la mise au point d'un système automatique de reconnaissance des empreintes digitales et en 1972, ce travail aboutit à la mise en place du premier système automatique de reconnaissance des empreintes digitales, *AFIS* (Automatic Fingerprint Identification System), qui utilisait un scanner conçu par le laboratoire aéronautique de *Cornell University* et un lecteur d'empreintes digitales conçu par *North American Aviation*⁹ (voir [178, 179]).

De nos jours, cette technologie de reconnaissance automatique des empreintes digitales est tellement utilisée que les systèmes biométriques des empreintes digitales sont devenus synonymes de la biométrie (par abus de langage bien sûr).

1.2.2 Définitions

Une empreinte digitale est le dessin formé par les lignes de la peau des doigts, des paumes des mains, des orteils ou de la plante des pieds. Ce dessin se forme durant la période fœtale. Il existe deux types d'empreintes digitales :

- *l'empreinte directe*, est celle qui laisse une marque visible après le relevé de l'empreinte digitale ou après le passage du doigt sur un capteur d'empreinte digitale; elle est donc propre, visible et facilite les opérations de traitement à effectuer pour l'identification ou l'authentification.

7. L'étude des empreintes digitales.

8. Chaque carte comprenait les empreintes des 10 doigts d'une personne, imprimées avec de l'encre.

9. Le tout premier constructeur aéronautique militaire américain.

- *l'empreinte latente* qui est celle obtenue avec de la saleté, de la sueur ou de tout autre résidu déposé sur un objet avant le scannage à l'aide d'un capteur ou avant le relevé de la trace d'empreinte. Ce type d'empreinte pose d'énormes problèmes lors des opérations d'identification/authentification et continue de faire l'objet de nombreuses recherches, notamment avec pour objectif d'éliminer le bruit (saleté) sans toutefois endommager les traits caractéristiques de l'empreinte [124, 149].

Les empreintes digitales sont *uniques* et *immuables*, elles ne se modifient donc pas au cours du temps (sauf par accident comme une brûlure par exemple).

1.2.3 Caractéristiques d'une empreinte digitale

Une empreinte digitale est formée d'un ensemble de lignes dessinées sur l'épiderme appelées *stries*, et d'espaces entre les stries appelés *vallées*. En regardant la structure globale d'une empreinte digitale, on remarque que dans certaines régions les dessins de stries adoptent certaines formes distinctes. Ces régions s'appellent des *régions caractéristiques* ou *régions singulières* ; elles sont classées en trois grandes catégories : *boucle*, *delta* et *spires* ou *tourbillon* ; symbolisées respectivement par \cap , Δ et O . C'est ainsi que Francis Galton a classé les empreintes en trois classes principales : *Boucle*, *Arche* et *Spires* [58] ; et Edward Henry les a subdivisées en sous-classes : *boucle à gauche*, *boucle à droite*, *arche*, *arche penchée ou tentée*, *spires* et *spires imbriquées ou boucles jumelles* [74] comme le montre la figure 1.4.

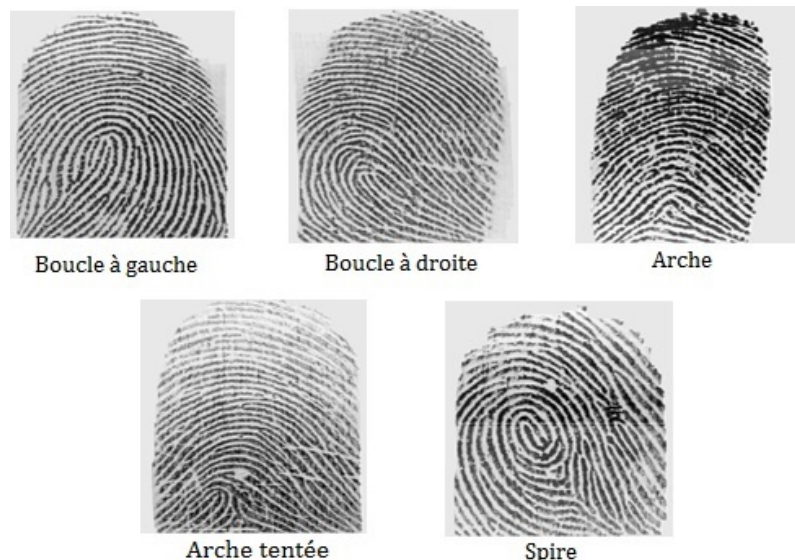


FIGURE 1.4 – Principales classes d'empreintes digitales selon Galton-Henry [58]

Dans ce système, le classement repose sur la topographie générale de l'empreinte

digitale et permet de définir des caractéristiques.

Les trois classes principales d'empreintes digitales décrites plus haut par *Galton*, regroupent 95% des doigts humains répartis comme suit : 60% pour les boucles, 30% pour les spirales et 5% pour les arches [58]. On différencie les motifs entre eux à l'aide de "*points singuliers*" sur les boucles, arches ou spires (voir section 2.4) :

- Points singuliers globaux :
 - Noyau ou centre : lieu de convergences des stries ;
 - Delta : lieu de divergences des stries.
- Points singuliers locaux (appelés aussi *minuties*) : points d'irrégularité se trouvant sur les lignes papillaires (terminaisons, bifurcations, îlots-assimilé à 2 terminaisons, lacs, ...).

Une autre caractéristique globale d'une empreinte utilisée dans certains algorithmes de comparaison d'empreintes digitales est le *cœur ou centre de l'empreinte* ou le *centre* . C'est le point supérieur de la ligne la plus au centre de l'empreinte comme le montre la figure 1.5.



FIGURE 1.5 – Caractéristiques d'une empreinte digitale [109]

On voit les régions singulières delta (triangle rouge) et le centre (cercle rouge) qui apparaissent clairement comme éléments prépondérants dans la caractérisation d'une empreinte digitale. Les caractéristiques globales d'une empreinte sont donc les régions singulières et le cœur de l'empreinte. Les empreintes digitales possèdent également des traits caractéristiques locaux appelés *minuties* (littéralement : petits détails), dont on retrouve quelques uns en carré rouge sur la Figure 1.5. Dans la biométrie des empreintes

digitales, ce terme désigne les différentes discontinuités des lignes d'une empreinte. En effet, une strie peut bifurquer (la minutie s'appelle alors la *bifurcation*), s'arrêter soudainement (on parle alors de la *terminaison*), se déchirer au milieu pour former une sorte de trou (c'est le *lac*), ...

Il existe cependant plusieurs sortes de minuties [9, 106]. L'institut américain qui supervise le développement de normes *ANSI* (*American National Standard Institute*) a proposé un standard de 4 types : les *terminaisons*, les *bifurcations*, les *trifurcations* et les *sous-terminaisons* [9]. On recense cependant, environ 16 types différents de minuties permettant de classifier les empreintes digitales et d'en assurer leur unicité (les bifurcations ayant 5 sous-classes), dont les 6 plus fréquents sont : *bifurcation*, *lac*, *terminaison*, *crochet*, *sillon indépendant*, *croisement ou pont*. La figure 1.6 ci-dessous illustre les différents types de minutie.

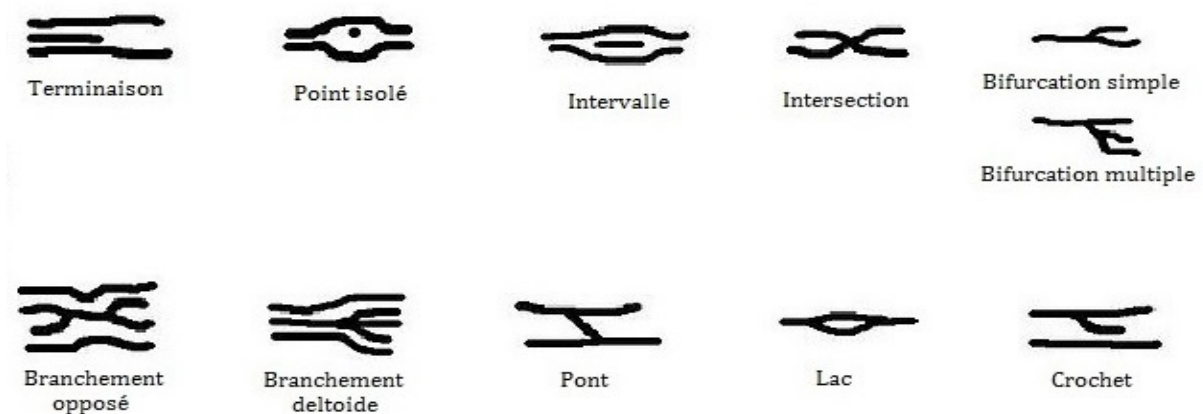


FIGURE 1.6 – Les différents types de minutie

En 1977, *Osteburg* dans son article [124], a analysé la fréquence d'apparition de chaque type de minuties et a montré que les *branches* et les *terminaisons* sont les plus fréquentes sur les lignes papillaires d'une empreinte digitale.

La probabilité que deux personnes aient la même empreinte digitale est de 10^{-24} , ce qui est très faible à l'échelle de la population humaine. De plus, son caractère aléatoire s'affranchit des risques de ressemblances entre individus partageant un même patrimoine génétique : des individus *monozygotes*¹⁰ comme des jumeaux ou des quadruplés par exemple auront chacun un jeu d'empreintes digitales qui leur sera propre et différent de celui des autres individus de la même fratrie, de même pour les empreintes légèrement différentes entre la main gauche et la main droite. L'explication est que les gènes sont

10. Terme qualifiant les jumeaux qui proviennent d'un même œuf. On dit aussi "vrais jumeaux"

responsables de l'architecture générale des empreintes (les trois grands motifs) alors que le développement embryonnaire et l'environnement de la vie intra-utérine influent sur les points singuliers.

Le Tableau 1.2 ci-dessous fait un récapitulatif selon plusieurs auteurs, des différentes probabilités de collision de deux empreintes digitales provenant de deux individus différents ; Où N représente le nombre de minuties, M et R étant les paramètres de régions

<i>Auteur(s)</i>	<i>Probabilité de collision</i>	$N = 36$ $R = 24$ $M = 72$	$N = 12$ $R = 8$ $M = 72$
Galton (1892)	$\frac{1}{16} \times \frac{1}{256} \times (\frac{1}{2})^R$	1.45×10^{-11}	9.54×10^{-7}
Pearson (1930)	$\frac{1}{16} \times \frac{1}{256} \times (\frac{1}{36})^R$	1.09×10^{-41}	8.65×10^{-17}
Henry (1900)	$(\frac{1}{4})^{N+2}$	1.32×10^{-23}	3.72×10^{-9}
Balthazard (1911)	$(\frac{1}{4})^N$	2.12×10^{-22}	5.96×10^{-8}
Bose (1917)	$(\frac{1}{4})^N$	2.12×10^{-22}	5.96×10^{-8}
Wentworth & Wilder (1918)	$(\frac{1}{50})^N$	6.87×10^{-62}	4.10×10^{-21}
Cummins & Midlo (1943)	$\frac{1}{31} \times (\frac{1}{50})^N$	2.21×10^{-63}	1.32×10^{-22}
Gupta (1968)	$\frac{1}{10} \times \frac{1}{10} \times \frac{1}{10}^N$	1.00×10^{-38}	1.00×10^{-14}
Roxburgh (1933)	$\frac{1}{1000} \times (\frac{1.5}{10 \times 2.412})^N$	3.75×10^{-47}	3.35×10^{-18}
Trauring (1963)	$(0.1944)^N$	2.47×10^{-26}	2.91×10^{-9}
Osterburg et al. (1980)	$(0.1944)^{M-N} \times (0.234)^N$	1.33×10^{-27}	3.05×10^{-15}
Stoney (1985)	$\frac{N}{5} \times 0.6 \times (0.5 \times 10^{-3})^{N-1}$	1.2×10^{-80}	3.5×10^{-26}

TABLE 1.2 – Probabilité de collision de 2 empreintes selon plusieurs auteurs [124]

1.2.4 Le relevé des traces digitales

Principe

Pour relever les traces digitales directes, les dactylo-techniciens, spécialistes des relevés, photographient ces empreintes avec une échelle, puis les transfèrent sur un support en matière plastique souple, mis sous scellé et envoyé au laboratoire pour être analysé.

Pour relever les traces digitales latentes, les dactylo-techniciens emploient depuis les origines le pinceau et une poudre très fine (céruse, alumine, oxyde de cuivre, poudre magnétique) [78, 29]. Différentes poudres sont à leur disposition :

- poudre noire pour les surfaces blanches.
- poudre blanche (à base d'aluminium) pour les surfaces lisses (bois, verre, mur, etc.)
- poudre fluorescente pour des fonds multicolores.

Lorsque les poudres se révèlent inefficaces (ne permettent pas de recueillir un minimum de traits caractéristiques), on utilise des techniques alternatives. Pour les surfaces poreuses (papier, carton, kraft, etc...), la pièce est d'abord plongée dans un bain chimique, une solution appelée *DFO* (*Diazafluornone*), puis envoyée dans une étuve pour le séchage. L'empreinte est alors révélée par une lumière ultraviolette spéciale. La ninhydrine permet également de révéler ce type d'empreintes qui apparaissent en pourpre et se révèlent souvent de meilleure qualité qu'avec la DFO. Les meilleurs résultats sont obtenus à partir d'une solution combinée d'*indanedione* et de *chlorure de zinc* 18.



FIGURE 1.7 – relevé des traces d'empreintes digitales

Pour les supports lisses type latex, on emploie une "*technique de fumigation*" : on fait chauffer la *cyanoacrylate* dans une enceinte contenant les pièces à analyser jusqu'à ce qu'elle se vaporise. Les vapeurs de cette colle se déposent alors sur les composants de l'empreinte : une belle trace blanche apparaît (sur les surfaces blanches on utilise plutôt des colorants).

Pour révéler les empreintes les plus ténues (comme sur les textiles ou certains papiers), on utilise le procédé de métallisation sous vide : on fixe l'objet dans un gros caisson métallique, dans lequel on crée un vide si poussé que les métaux (d'or et de zinc) placés à l'intérieur dans de petites coupelles se transforment en gaz. L'or se dépose sur toute la surface de l'objet tandis que le zinc se dépose entre les crêtes papillaires. L'or ne se mélange donc au zinc qu'aux endroits où aucune empreinte ne se trouve : l'empreinte digitale apparaît en négatif. Cette technique coûteuse est efficace sur tout type de support, à condition que l'objet ne soit pas trop volumineux ni compressible (comme le polystyrène, par exemple).

Plus récemment, on utilise le "Crimescope" ou le "Polilight" : ces lasers lumineux émettent, par l'intermédiaire de fibres optiques, des longueurs d'onde allant de l'infrarouge à l'ultraviolet. Après obscurcissement de la pièce, les traces digitales sont éclairées dans cet intervalle du spectre électromagnétique, réfléchissent la lumière par le phénomène de luminescence et sont révélées (au même titre que les poils, les fibres et des minuscules résidus biologiques).

1.3 Les capteurs d'empreintes digitales

Le point fort de l'identification par empreinte digitale est qu'elle peut être utilisée dans des environnements très divers. C'est une technologie prouvée et le fait de pouvoir utiliser différents doigts permet d'augmenter considérablement l'exactitude du système ainsi que sa flexibilité. La première phase dans la construction d'un système d'identification par empreinte digitale consiste à obtenir une image de l'empreinte du doigt. Longtemps le seul moyen existant a été l'utilisation du papier et de l'encre, ce qui a rendu la tâche de reconnaissance ardue. En effet la qualité de l'image était plutôt mauvaise (plusieurs acquisitions étaient nécessaires) et l'extraction des minuties était effectuée visuellement par un expert (processus très long et pénible). Le développement de l'Informatique et de la microélectronique a été à l'origine de l'apparition de nouveaux moyens d'acquisition de l'image d'une empreinte digitale (le *capteur*), permettant ainsi d'accélérer le processus de traitement en l'automatisant [28, 109, 112, 169].

Définition

Un *capteur photographique* est un composant électronique photosensible servant à convertir un rayonnement électromagnétique (visible ou infrarouge) en un signal électrique analogique. Ce signal est ensuite amplifié, puis numérisé par un convertisseur *analogique-numérique* et enfin traité pour obtenir une image numérique. Il existe plusieurs types de capteurs d'empreintes digitales. Parmi les plus connus on distingue :

les capteurs optiques d'empreintes, les capteurs électriques-thermiques, les capteurs capacitifs, les capteurs de champ-électrique, les capteurs de pression et les capteurs sans contact.

1.3.1 Les capteurs optiques d'empreintes

La méthode optique est l'une des techniques les plus communes. Un appareil-photo encore appelé *Charge – Coupled Device (CCD)* est utilisé au cœur du capteur optique. Le *CCD*, ou en français "*dispositif à transfert de charge (DTC)*" est le plus simple à fabriquer. Inventé par *George E. Smith* et *Willard Boyle* dans les Laboratoires *Bell* en 1969 [138] (cette invention leur rapportera la moitié du Prix Nobel de physique en 2009), il a rapidement été adopté pour des applications de pointe (imagerie astronomique) puis popularisé sur les caméras et appareils photos. Un appareil-photo *CCD* se compose simplement d'une rangée de *diodes* sensibles légères appelées *photosites*. En général, le doigt est placé sur une surface en verre et l'appareil-photo *CCD* prend la photo. Le système *CCD* contient une rangée de *LED* (diodes électroluminescentes) qui illumine les creux et les bosses du doigt [169]. Les principaux fabricants de capteurs optiques sont *Delsy*, *Dermalog*, *Smiths Heimann Biometrics*. La figure 1.8 ci-dessous montre un exemple de capteur optique CCD.

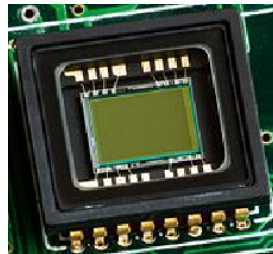


FIGURE 1.8 – Capteur optique CCD

Avantages

Les avantages des capteurs optiques sont de divers ordres : ils résistent très bien aux fluctuations de température, sont intrinsèquement protégés contre les décharges électrostatiques et permettent d'avoir des images précises et nettes.

Inconvénients

Comme inconvénients, nous pouvons relever qu'ils ont un coût assez élevé, posent un problème d'encombrement car ils sont dotés d'un assemblage mécanique peu commode

(pour les systèmes portables) et enfin ils offrent parfois une distorsion d'image liée à l'optique et à l'écrasement du doigt sur le prisme. L'autre problème est celui des empreintes latentes : l'empreinte digitale du doigt précédent, qui a été placé sur le capteur, peut rester.

Différents types de capteur optique

Il existe trois types de CCD :

- Le *CCD "plein cadre" (full frame)* : où l'ensemble de la surface contribue à la détection. C'est le plus sensible mais il présente plusieurs inconvénients : les électrodes (grilles) en silicium polycristallin circulent au-dessus de la couche photosensible et absorbent une part importante de la partie bleue du spectre (0,35 – 0,45 micromètre) ; il nécessite un obturateur externe pour permettre le cycle de transfert de charge sans illumination ; il est très sensible à l'éblouissement (*blooming*). Quand un photosite déborde, il inonde ses voisins. Pour pallier cet inconvénient, il peut être équipé d'un dispositif dit "*drain d'évacuation de charges*" (*LOD-Lateral Overflow Drain*) qui élimine les électrons en trop plein des photosites et limite la propagation de l'éblouissement, mais diminue la sensibilité. Les CCD "plein cadre" récents ont des *photosites* au pas de 6 micromètres capables de stocker jusqu'à 60 000 électrons et ont un rendement quantique supérieur à 20%. On sait, en 2013, fabriquer des CCD "plein cadre" de 80 mégapixels (surface utile de $53,7 \times 40,4$ mm) [169].
- Le *CCD "à transfert de trame" (full-frame transfer)* : il associe deux matrices CCD de même dimension, l'une exposée à la lumière, l'autre masquée. On peut ainsi procéder à un transfert rapide de la matrice d'exposition vers la matrice de stockage puis à la numérisation de celle-ci en parallèle avec l'acquisition d'une nouvelle image. Le principal inconvénient est qu'il diminue par deux la surface du photosite à taille de capteur égale (sensibilité moitié moindre), les autres inconvénients (réponse spectrale, éblouissement) demeurent.
- Le *CCD "interligne"* : plus complexe, il associe une photodiode à chaque cellule CCD. C'est lui qui est principalement utilisé dans les photoscopes. La photodiode spécialisée permet de retrouver une réponse spectrale couvrant correctement le spectre visible (0,35 – 0,75 micromètre), il est généralement équipé d'un drain d'évacuation de charges qui limite la propagation de l'éblouissement. Il est par contre intrinsèquement moins sensible, les photodiodes ne représentant que 25% à 40% de la surface totale. Ce défaut est partiellement corrigé par un réseau de microlentilles convergentes qui améliore le rendement quantique de 15% à 35 – 45%. Les CCD interlignes récents ont des *photosites* au pas de 8 micromètres capables de

stocker jusqu'à 100 000 électrons [169].

1.3.2 Les capteurs en silicium

Pour remédier aux défauts des capteurs optiques, les ingénieurs cherchent à éliminer les éléments volumineux et onéreux à produire que sont le prisme, la lampe et la lentille ; de façon à ce que le doigt soit en contact direct avec le capteur [154]. Après cette élimination de caractéristiques contraignantes, les ingénieurs tentent de réduire la chaîne de traitement de l'information. Les procédés en développement conservent la structure classique des capteurs optiques (c'est à dire une matrice de pixels adressée en lignes et colonnes), mais seulement, alors que dans les circuits de capteurs optiques, les microcapteurs placés sur chaque pixel transforment la lumière en une information électrique (effet électro-optique), on fait recourt à des microcapteurs d'un type nouveau. Il utilise l'un des quatre effets des semi-conducteurs (piezo-électrique, capacitif, thermo-électrique ou photo-électrique) et combine un encombrement très réduit avec un coût modéré. C'est ce qu'utilise le lecteur *Eikon* et qui lui permet d'arborer son design fin [154].



FIGURE 1.9 – Capteur en silicium © *Eikon*

Avantage

Ce capteur a une durée de vie assez longue et permet de diminuer le bruit d'une image.

Inconvénients

Par contre il présente les inconvénients ci-après : les composés piézo-électriques sont insuffisamment sensibles, il consomme beaucoup d'énergie, est fragile aux décharges électrostatiques et présente des coûts un peu élevés (bien qu'il reste moins cher qu'un capteur optique).

1.3.3 Les capteurs électriques-thermique

La méthode électrique-thermique est moins habituelle. Actuellement, le seul capteur thermique est le *FingerChip* fabriqué par *Atmel* [32]. La méthode unique du *FingerChip*, pour reconnaître l’empreinte, consiste à faire glisser le doigt le long du capteur. Les images successives ainsi capturées sont analysées par un logiciel spécial qui reconstruit l’image de l’empreinte digitale. Cette méthode permet au *FingerChip* de renvoyer une haute qualité de l’image capturée, 500 *dpi* et 256 niveaux de gris. Le capteur mesure le différentiel de température entre les creux de la peau et l’air capturé dans les bosses de l’empreinte digitale. Cette méthode donne une image d’excellente qualité même sur des empreintes de qualité médiocre telles que celles provenant des doigts secs avec peu de profondeur entre les creux et les bosses. La technologie thermique fonctionne également dans des conditions environnementales difficiles, comme lors de températures extrêmes, de taux d’humidité ou de poussière élevé, ou de contamination d’eau.



FIGURE 1.10 – Capteur électrique-thermique

Cette méthode a également l’avantage de nettoyer le capteur, évitant ainsi que les empreintes digitales restent après le passage d’une personne. En effet, les empreintes latentes peuvent poser des problèmes de lecture à chaque passage mais peuvent aussi être copiées et utilisées pour parvenir aux accès contrôlés. En fait, cette méthode, s’appuyant sur la technologie thermique, permet au *FingerChip* d’être un des plus résistant par rapport aux autres technologies. Le *FingerChip* fonctionne aussi dans des environnements où la température peut être basse et le taux d’humidité élevé.

Avantage

L’avantage de cette technologie est donc la possibilité d’avoir une image de bonne qualité et un capteur toujours propre.

Inconvénients

L'inconvénient principal est que la qualité de l'image dépend des compétences de l'utilisateur à savoir manipuler le capteur. Le deuxième inconvénient est le chauffage du capteur qui augmente la consommation électrique. La chaleur du capteur est nécessaire afin d'éviter un équilibre thermique entre le capteur et la surface du doigt.

1.3.4 Les capteurs capacitifs

La méthode capacitive est une des méthodes les plus populaires. Comme les autres capteurs, le capteur capacitif d'empreinte digitale reproduit l'image des creux et des bosses qui composent une empreinte digitale. Le capteur capacitif emploie des condensateurs de courant électrique pour mesurer l'empreinte digitale et se compose d'une rangée de cellules minuscules [154]. Chaque cellule inclut deux plaques conductrices recouvertes par un revêtement protecteur. La figure 1.11 montre deux types de capteurs capacitifs.

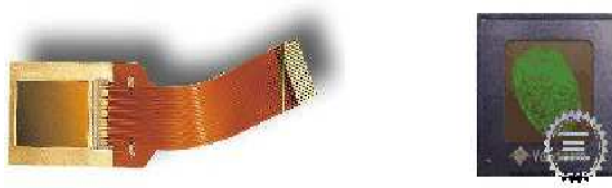


FIGURE 1.11 – Capteur capacitif

1.3.5 Les capteurs de champ-électrique

Le capteur de champ-électrique fonctionne avec un champ-électrique et le mesure au delà de la couche extérieure de la peau où l'empreinte digitale commence. La technologie de champ-électrique peut être utilisée dans des conditions extrêmes, c'est-à-dire même si le doigt est sale ou sec. Cette technologie crée un champ entre le doigt et le semi-conducteur adjacent qui imite la forme des creux et des bosses de la couche épidermique du doigt. Un amplificateur de sous-pixel est utilisé pour mesurer les signaux. Les capteurs fonctionnent ensemble afin de rendre une image propre correspondant exactement au modèle de l'empreinte digitale [169]. On parvient ainsi à une image plus claire que ce que peuvent donner les technologies optiques ou capacitives. Cela permet à la technologie de champ-électrique d'obtenir des images d'empreinte digitale que d'autres technologies ne parviendraient pas à avoir.

Avantages

Dans la technologie de champ-électrique, les rangées d'antenne mesurent la surface de la peau en générant et en détectant les formes linéaires géométriques. La couche vivante des cellules de la peau commence sous la surface de la peau... Cela est contraire aux géométries sphériques ou tubulaires de champ produits par le capteur capacitif simple, qui lit seulement la surface supérieure de la peau. En conséquence, des doigts, qui seraient difficiles ou impossibles à scanner par des capteurs capacitifs, le seraient avec succès grâce à la technologie de champ-électrique.

Inconvénients

L'inconvénient majeur de cette méthode est la basse résolution d'images et une trop petite zone d'image, ce qui a pour conséquence de générer un taux d'erreur élevé ($EER = Equal-Error-Rate$) par rapport aux autres capteurs.

1.3.6 Les capteurs de pression

Principe

Le principe du capteur de pression réside dans le fait que lorsqu'un doigt est placé au-dessus de la zone du capteur, seules les bosses de l'empreinte digitale rentrent en contact avec les rayons *piezo* du capteur [169]. Les creux en contraste n'ont aucun contact avec les cellules du capteur. La figure 1.12 ci-dessous montre un exemple de capteur de pression.



FIGURE 1.12 – Capteur de pression

Avantages

Un capteur de pression est capable d'enregistrer des doigts aussi bien humides que secs. De plus, sa large zone d'enregistrement permet de scanner complètement l'empreinte digitale ce qui amène à un faible taux d'erreur (EER).

Inconvénients

Une des différences fondamentales pour la reconnaissance et la comparaison d'empreinte digitale, par rapport aux capteurs sur le marché, demeure le capteur de pression qui génère une image binaire de 1 – bit. Une image de 1 – bit contient moins d'informations qu'une image de niveau de gris de 8 – bits.

1.3.7 Les capteurs sans contact

Le capteur sans contact fonctionne comme un capteur optique. En général, il y a une glace optique de précision à une distance de 5 à 7 cm de l'empreinte digitale lorsque le doigt est scanné. L'empreinte digitale est mise sur un support avec une ouverture.

Inconvénients

Un des inconvénients à considérer est que la poussière et la saleté peuvent se déposer sur la vitre optique, donnant un mauvais résultat d'image [146, 169]. D'autre part, les empreintes digitales scannées sont sphériques ce qui amène à des algorithmes de comparaison plus complexes.

1.3.8 La scannerisation à partir du papier

Cette technique consiste à numériser une empreinte digitale en utilisant du papier. Premièrement l'empreinte est trempée dans de l'encre puis apposée sur du papier. Ensuite, ce papier est scanné directement à l'aide d'un scanner numérique qui permet ainsi d'obtenir l'image d'empreinte digitale numérisée.

Elle a la particularité qu'elle permet à l'utilisateur de créer sa propre base de données en y introduisant un nombre d'exemplaires souhaité, ceci par classe de données. Ce qui est un atout lorsqu'on sait que la plupart des bases de données d'empreintes digitales existantes ne sont pas publiques. Cependant, elle contribue largement à augmenter le bruit de l'image du fait que le doigt (ou l'objet) à scanner n'est pas directement passé à travers un capteur numérique. Ceci peut parfois entraîner une augmentation des opérations de filtrage et accroître les coûts en terme de temps de calculs.

Cette technique est de moins en moins utilisée et tend à être remplacée par la capture de l'image d'empreinte directement à l'aide de l'un des capteurs numérique définis ci-dessus, dont les coûts commencent à être abordables. Néanmoins la scannerisation peut être utilisée en fonction de l'objectif principal qu'on s'est fixé à l'avance.

1.4 Les problèmes liés à la biométrie

Lors de leur utilisation pour l'identification/authentification, les systèmes biométriques estiment le degré de similitude entre deux patterns durant la phase finale de reconnaissance et le comparent à un seuil fixé à l'avance par l'utilisateur en fonction du résultat escompté. Ainsi le résultat n'atteint jamais un degré de fiabilité de 100% mais s'en approche selon le réglage du seuil [82]. L'évaluation des performances de ces systèmes biométriques fait apparaître deux principaux types d'erreur [28, 78, 82, 109] :

- le Taux de Faux Rejets (TFR) qui correspond au pourcentage de personnes rejetées à tort.
- le Taux de Fausses Acceptations (TFA) qui correspond au pourcentage de personnes qui ont été acceptées mais qui n'auraient pas dû l'être.

La relation qui lie le TFR et TFA est illustrée par la figure 1.13. On peut voir que plus

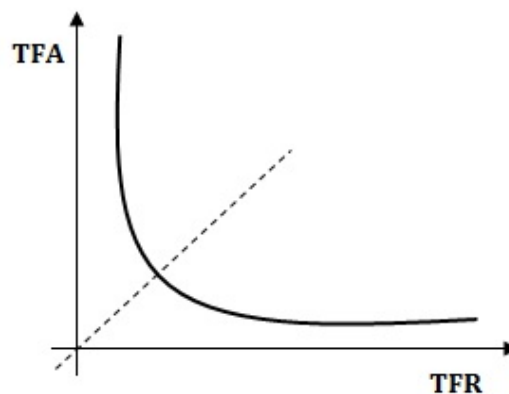


FIGURE 1.13 – Relation entre le TFR et le TFA

le TFR est faible, plus le TFA est élevé et inversement. Ces deux éléments dépendent du réglage du seuil qui est le résultat d'un compromis selon le choix de l'application [85]. Pour des applications de haute sécurité (accès à un coffre fort de banque par exemple), on cherchera à limiter au maximum la possibilité d'intrusion, ce qui se traduit par un faible TFA . Par contre, dans le cadre d'une application utilisée à des fins policières (pour l'identification des criminels), on ne pourra pas se permettre de rejeter une personne par erreur ce qui implique un TFR le plus faible possible.

Les systèmes biométriques posent aussi des problèmes de sécurité car il n'est pas vérifié que les données biométriques en entrée proviennent de leur réel possesseur. En effet les moyens de flouer un tel système existent, et de plus les caractères biométriques ne sont pas secrets [64]. Il est relativement facile de se procurer une photo du visage, une

empreinte digitale ou un enregistrement audio d'une personne. Une équipe japonaise a d'ailleurs démontré au début des années 2000, que les systèmes actuels d'acquisition d'empreintes acceptent très bien une fausse empreinte créée à partir d'un moule en gélatine¹¹ [114]. Le problème vient de l'absence de vérification que la donnée entrée provient bien d'un être vivant.

L'utilisation de la biométrie au quotidien pose également des problèmes de vie privée et de libertés [85]. Ce qui a d'ailleurs suscité de la part de nombreux pays le non stockage en masse de caractéristiques biométriques des personnes face aux risques de réutilisations abusives de ces données.

1.5 Conclusion

Dans ce chapitre, après avoir présenté les différentes techniques biométriques existantes (dont la liste ne cesse d'augmenter), impliquées dans le processus d'identification ou de l'authentification, nous avons abordé les empreintes digitales et ressorti leurs caractéristiques fondamentales. Il en ressort que les méthodes d'identification basées sur les empreintes digitales sont les plus abouties de nos jours, bien que les techniques biométriques dans leur ensemble posent encore de nombreux problèmes. Enfin, les deux principales méthodes d'obtention d'une image d'empreinte digitale, à savoir le capteur et le relevé des traces d'une empreinte, ont été étudiées. La suite de ce travail, s'intéresse aux principes fondamentaux de la reconnaissance d'individus pas les empreintes digitales et à l'apport des approches neuronales dans cette branche de la biométrie.

11. Gelée que l'on obtient en faisant bouillir les os, la peau et les cartilages des animaux

Principes fondamentaux de la reconnaissance d'empreintes digitales

2.1 Introduction

Le principe de la reconnaissance d'empreintes digitales consiste à comparer une empreinte fournie au système, à une ou plusieurs autres empreintes (*les templates*) que le système dispose préalablement dans sa base de données biométrique. Le système de reconnaissance automatique (qui ne traite que des images numériques) renvoie un résultat positif au cas où l'empreinte fournie en entrée correspond à l'un des templates (dans le cas où elle est comparée à plusieurs templates) et un résultat négatif dans le cas contraire.

Pour pouvoir obtenir une image numérique d'une empreinte digitale, on utilise l'un des capteurs d'empreintes décrits à la section 1.3 ci-dessus. Cependant, différents facteurs jouent sur la qualité de l'image capturée ; à savoir :

- la pression que l'on exerce sur le capteur,
- la lumière ambiante lors du scan du doigt,
- les doigts sales ou abîmés, ...

Après la capture de l'image de l'empreinte digitale, elle devra subir divers traitements (contraste, filtrage, binarisation, squelettisation, ...) permettant de définir et d'extraire plus facilement les minuties. La figure 2.1 ci-dessous illustre l'organigramme du principe général d'un système de reconnaissance d'empreintes digitales.

On distingue dans la littérature deux catégories d'algorithmes de reconnaissance d'empreintes digitales : la première catégorie concerne les algorithmes plutôt "*conventionnels*" qui s'appuient sur la position relative des minuties entre elles [82, 109], alors que la seconde regroupe les algorithmes visant à extraire d'autres particularités de l'empreinte digitale telles que la direction locale des sillons [28], ou encore les composantes

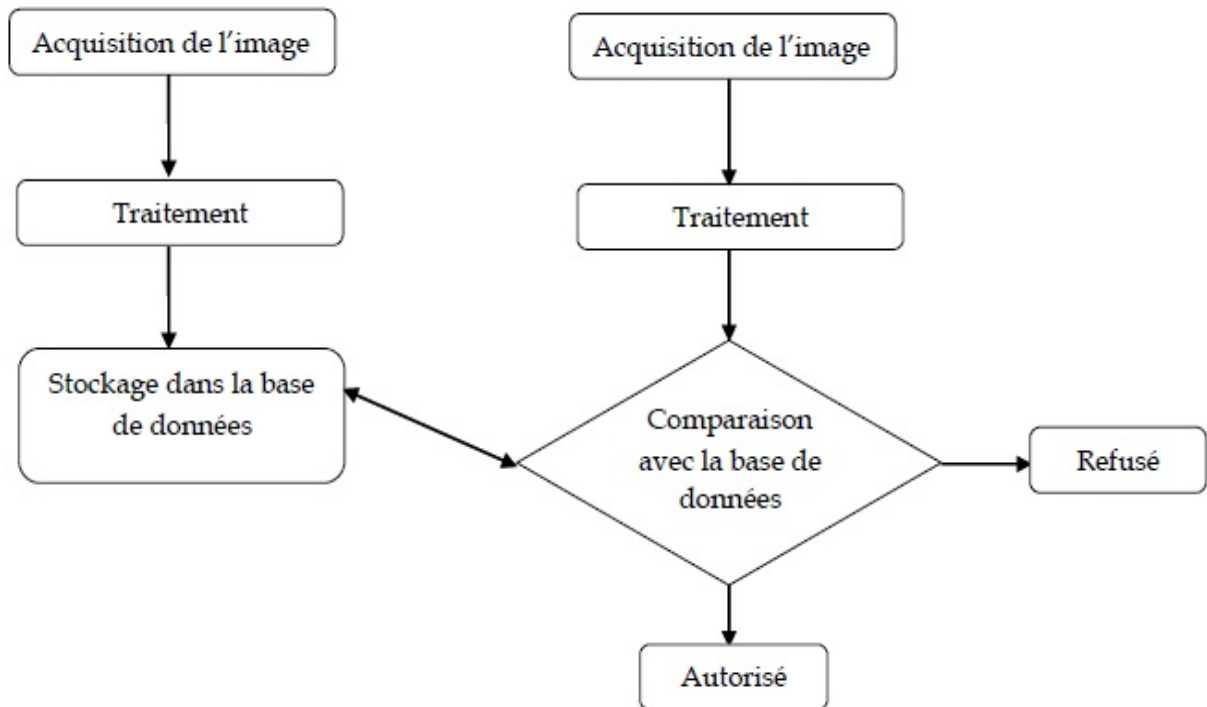


FIGURE 2.1 – Principe générale d'un système de reconnaissance d'empreintes digitales

fréquentielles locales de la texture au cœur de l'image [78].

Ce travail étudie en profondeur la première catégorie, qui est vraisemblablement la plus connue et la plus utilisée [82, 109]. On réalise donc successivement le filtrage directionnel, la binarisation de l'image, l'amincissement (ou squelettisation) des sillons, puis on détermine la position des minuties au sein de l'image pour quantifier les caractéristiques de ressemblance entre deux empreintes par "*point pattern matching*".

De façon globale, un système complet de reconnaissance d'empreintes digitales consiste en deux étapes principales : *l'enregistrement* et *l'identification*.

2.1.1 L'enregistrement des templates dans la base

Le processus d'*enrôlement* ou *enregistrement* consiste à construire la base de données qui servira à identifier ou authentifier un individu qui souhaite accéder au système conçu. En effet, lors du processus d'enregistrement, l'image scannée de l'empreinte digitale est recueillie par le système, puis un contrôle de la qualité de l'image est effectué. Ce contrôle consiste en deux phases successives ayant pour buts respectifs de rendre les pixels de l'image de l'empreinte uniformes en les ramenant tous dans l'intervalle $[0..1]$, ensuite de donner la même taille aux lignes de l'empreinte ; il s'agit de deux phénomènes bien

précis, à savoir la *binarisation* et la *squelettisation* de l'image d'empreinte. La figure 2.2 ci-après illustre la procédure d'*enregistrement* des templates.

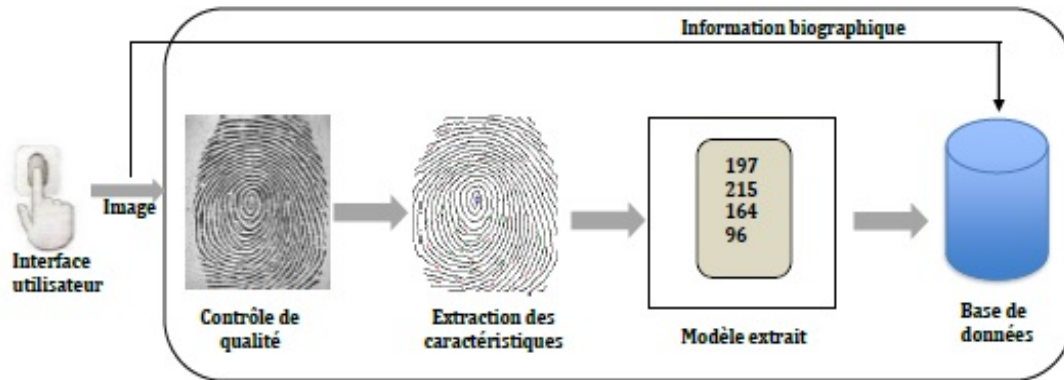


FIGURE 2.2 – Illustration du processus d'enregistrement/enrôlement

Il faut tout de même noter qu'une empreinte sérieusement endommagée (par une brûlure grave par exemple) sera intraitable et automatiquement rejetée par le système (voir [112]).

En effet la binarisation d'une image consiste à mettre l'image de l'empreinte en noir et blanc et la squelettisation quant à elle permet d'éliminer le bruit et de faire ressortir clairement les vallées et les stries qui faciliteront à leur tour l'extraction des minuties et la détermination de leur direction dans l'image (voir figure 2.12) [109, 112]. Une fois que l'on dispose de l'image *binnaire squelettisée*, les minuties (singularités) sont mieux visibles; on procède alors à leur détection. La détection des minuties sera exposée en détail à la section 2.2.

2.1.2 *L'identification/authentication*

L'étape de l'identification/authentication quant à elle permet au système de prendre la décision finale sur la validation ou non de l'image d'empreinte prise en entrée. Il est à noter qu'il existe une différence fondamentale entre l'authentification et l'identification.

L'authentification consiste à recueillir une image numérique d'empreinte digitale en entrée du système et de la comparer à une autre image provenant de la base de données construite lors de l'étape d'enregistrement précédemment décrite. La comparaison ici est donc de type "*un contre un*" et le système renvoie un message d'authentification (suivant un seuil prédéfini) pour autoriser ou non l'accès à un utilisateur quelconque. La

figure 2.3 ci-dessous illustre le processus d'authentification d'un individu par empreintes digitales. A la fin du processus, l'utilisateur est autorisé ou non à entrer dans le système.

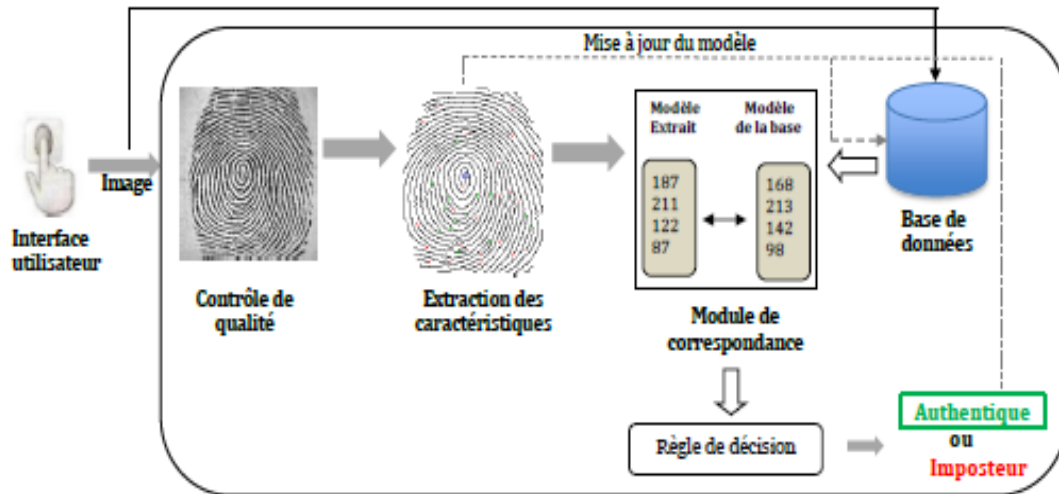


FIGURE 2.3 – Illustration du processus d'authentification

L'**identification** quant à elle consiste à recueillir une image numérique d'empreinte digitale en entrée du système et de la comparer à plusieurs autres images qui ont été au préalable enregistrées dans la base de données (lors de l'enrôlement). La comparaison dans ce cas est plutôt de type "un contre plusieurs" (voir figure 2.4).

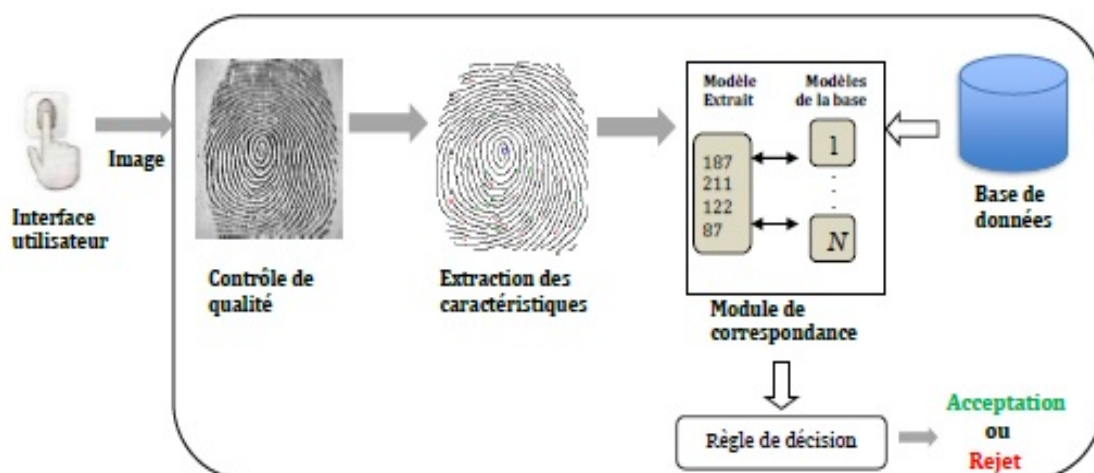


FIGURE 2.4 – Illustration du processus d'identification

Lors de l'identification, le système scanne l'empreinte de la personne à identifier et extrait les minuties après quelques opérations de pré-traitement exactement comme à la phase d'enregistrement. Le gabarit (ensemble formé par les différentes minuties détectées) de l'empreinte est ensuite comparée à celle des autres empreintes de la base de données. Si aucune d'elles ne correspond à l'empreinte fournie en entrée (suivant un seuil de tolérance préalablement défini), le système renvoie une réponse négative (l'accès au système est alors refusé à la personne), sinon une réponse positive est retournée en sortie (l'accès est autorisé) comme le montre la figure 2.4.

2.2 *Traitement et analyse de l'empreinte*

La reconnaissance d'une empreinte digitale est directement liée à la qualité de l'image obtenue. Ainsi dans la plupart des cas, un pré-traitement est nécessaire pour améliorer la qualité de l'image avant son enregistrement dans la base de données ou son utilisation lors de la phase de test. En effet avant la constitution de la base de données et la comparaison des empreintes digitales (celle en entrée du système avec celles contenues dans la base de données), trois grandes étapes cruciales doivent être effectuées afin de faciliter la procédure d'identification. Il s'agit du contrôle de qualité de l'image d'empreinte, du filtrage et de l'extraction des minuties qui s'appliquent aussi bien lors de la phase d'enregistrement que pendant celle de l'identification.

2.2.1 *Le contrôle de qualité de l'image d'empreinte*

L'objectif du traitement ici est d'obtenir une image simplifiée contenant toutes les informations de l'empreinte (sans bruit, avec un niveau de gris uniforme et des lignes identiques).

2.2.1.1 *Le ré-échantillonnage de l'image*

En fonction de l'outil d'obtention de l'image brute de l'empreinte digitale, certaines images peuvent se présenter sous la forme d'une matrice non carrée de taille $n \times m$. Dans ce cas un ré-échantillonnage peut s'avérer essentiel afin de rendre l'image d'empreinte carrée et d'éliminer ainsi les parties qui contiennent les informations peu ou pas utiles. Plusieurs techniques permettant d'effectuer efficacement le ré-échantillonnage d'une image qui existe. Cependant la méthode de réduction par interpolation bilinéaire est l'une des plus satisfaisantes- [78, 109].

i) La réduction de l'image par interpolation bilinéaire

Soit M_0 l'image d'origine de taille $n \times m$ et M_1 l'image après réduction verticale. La méthode d'interpolation bilinéaire consiste à attribuer à chaque pixels (i, j) de M_1 la valeur du pixel de M_0 en tenant compte de ses voisins par convolution bilinéaire d'un rapport :

$$k = \frac{n}{256} ; \text{On note } \theta = k.j - \lfloor k.j \rfloor .$$

La valeur du pixel (i, j) dans la nouvelle image est alors définie par :

$$M_1(i, j) = (1 - \theta) \times M_0(i, \lfloor k.j \rfloor) + \theta \times M_0(i, \lceil k.j \rceil) . \tag{2.1}$$

Cette étape n'étant pas primordiale au fonctionnement du système de reconnaissance, elle doit être rapide.

2.2.1.2 Estimation de l'orientation locale des lignes d'une empreinte

La notion de l'orientation locale des lignes d'une empreinte est très importante dans le processus de reconnaissance d'empreintes digitales. Elle est utilisée par de nombreux algorithmes [78, 109, 112, 163] de détection de traits caractéristiques d'une empreinte, notamment *l'algorithme de l'index du point carré* (voir en ii) qui sert à détecter les régions singulières. L'idée consiste donc à avoir une estimation de l'orientation des lignes d'une empreinte sur une petite zone de l'image (voir figure 2.5).



FIGURE 2.5 – (a) les lignes ont une orientation verticale, (b) les lignes ont une orientation oblique à gauche, (c) les lignes ont une orientation horizontale.

C'est l'orientation de ces différentes lignes qui donnera celle de l'image globale de l'empreinte digitale.

Avant de poursuivre, notons tout d'abord une petite notion qui sera utile pour comprendre la suite de cette sous-section : l'image numérique.

i) L'image numérique

De nos jours, il est fréquent de conserver ses photos sur un support électronique ou sur un ordinateur. Lorsqu'on regarde les propriétés d'une photo en faisant un clic droit de la souris puis un clic gauche sur l'onglet *propriétés*, on trouve entre autres informations la hauteur H et la largeur L de l'image en pixel(s). En fait, une image électronique peut être décomposée en une matrice $H \times L$ contenant des nombres entiers appelés *pixels* (*picture elements*), dont les valeurs déterminent l'intensité lumineuse de chaque point de l'image. Chaque point de l'image est donc représenté par un nombre situé à une ligne i et à une colonne j de la matrice des pixels de l'image tel qu'illustrée à la figure 2.6 ci-dessous ; ce nombre est généralement appelé le *pixel* $[i, j]$ avec $0 \leq i \leq 255$ et $0 \leq j \leq 255$.

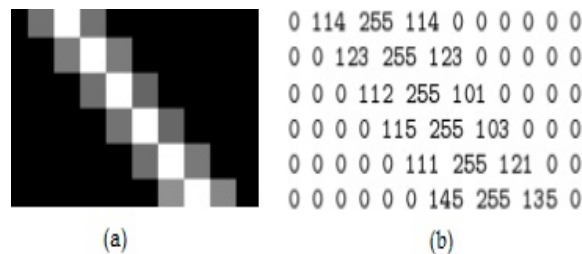


FIGURE 2.6 – (a) Image de hauteur 6 et de largeur 10, (b) la matrice 6×10 des pixels de l'image (a)

ii) Calcul de l'image directionnelle d'une empreinte

Dans la biométrie des empreintes digitales, une estimation de l'orientation des lignes de l'empreinte est reprise dans une matrice notée M , appelée *image directionnelle* ou tout simplement *orientation de l'empreinte digitale*. La case (i, j) de la matrice contient la valeur de l'angle θ_{ij} entre l'axe horizontal et l'orientation (approximative) des lignes de l'empreinte dans le voisinage du pixel $[i, j]$ comme l'illustre la figure 2.7.

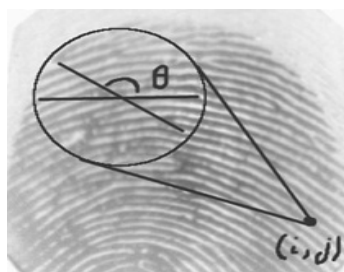


FIGURE 2.7 – L'axe oblique indique l'orientation des lignes au voisinage du point (i, j) .

Etant donné que les lignes d'une empreinte ne sont pas dirigées (c'est à dire ne possèdent pas de flèche à l'une des extrémités), l'orientation locale des lignes n'est pas dirigée non plus (voir l'axe oblique de la figure 2.7). La manière la plus courante et la plus simple de calculer la matrice \mathbf{M} consiste à calculer le gradient aux différents points de l'image \mathbf{E} (en noir et blanc) de l'empreinte. Le gradient d'une fonction f au point (x_i, y_j) , noté $\nabla f(x_i, y_j)$, est un vecteur $[\nabla_x(x_i, y_j), \nabla_y(x_i, y_j)]$ représentant la variation de f par rapport à la variation de ses variables x et y , où $\nabla_x(x_i, y_j)$ et $\nabla_y(x_i, y_j)$ sont respectivement la dérivée de f par rapport à x au point (x_i, y_j) et par rapport à y au même point [28, 116]. L'angle du gradient indiquant la direction de la variation maximale de f au point (x_i, y_j) est obtenu en calculant l'arctangente du rapport $\nabla_x(x_i, y_j) \div \nabla_y(x_i, y_j)$.

Dans le cas des empreintes digitales, l'intensité des pixels est la fonction à laquelle on s'intéresse. Etant donné que les valeurs présentes dans la matrice déterminent l'intensité des pixels de l'image, on va calculer le gradient à chaque pixel $[x_i, y_j]$ de l'image en faisant la dérivée de l'image E (ou plutôt de la matrice de pixels) au point $[x_i, y_j]$. L'angle $\theta_{i,j}$ du gradient au point x_i, y_j indique la direction de la variation maximale de l'intensité des pixels de E . En effet, étant donné qu'il s'agit d'une image en noir et blanc de l'empreinte (c'est-à-dire les lignes en "noir" et les vallées en "blanc"), les points de variation de l'intensité correspondent aux bords de lignes de l'empreinte ; c'est-à-dire les zones de transition entre les lignes et les vallées. L'angle du gradient indique donc l'orientation dans laquelle les lignes sont couchées. Il existe plusieurs manières de faire ce calcul, notamment l'algorithme de Sobel [156] qui est beaucoup utilisé dans l'imagerie.

a) L'algorithme de Sobel

Principe

Cet algorithme calcule le gradient de l'intensité en chaque pixel. Ceci indique la direction de la plus forte variation du clair au sombre, ainsi que le taux de changement dans cette direction [156]. On connaît alors les points de changement de luminosité, correspondant aux bords (les bords entre lignes et les vallées de l'empreinte dans notre cas), ainsi que l'orientation de ces bords (l'orientation des lignes de l'empreinte dans notre cas).

L'algorithme utilise deux matrices de convolutions (généralement de taille 3×3) pour calculer les approximations de dérivées par rapport à x et y . Notons E l'image source de l'empreinte, G_x et G_y deux images qui en chaque point contiennent des approximations respectivement de la dérivée par rapport à x et à y de chaque point. Ces images se calculent comme suit :

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \times E \text{ et } G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \times E$$

L'angle du gradient qui donne l'orientation de lignes de l'empreinte au voisinage d'un pixel $[x_i, y_j]$ est alors calculé par :

$$\theta = \arctan(G_x[x_i, y_j] \div G_y[x_i, y_j]). \tag{2.2}$$

La matrice D contiendra les différentes valeurs θ calculées à chaque pixel de l'image en noir et blanc de l'empreinte digitale ; c'est-à-dire une estimation de l'orientation des lignes de l'empreinte au voisinage de chaque pixel.

Limites

L'algorithme de *Sobel* donne une orientation approximative car il ne considère qu'un court voisinage 3×3 du pixel considéré, ajouté au fait que les lignes d'une empreinte ne sont pas tout à fait droites, mais les résultats sont suffisants pour se faire une idée de l'orientation des lignes sur une zone locale de l'image d'une empreinte digitale.

Bilan

Les informations de chaque minutie d'une empreinte seront alors reprises dans un vecteur de la forme (**type de minutie**, x , y , θ), ce qui est une manière pratique de sauvegarder les informations d'une empreinte digitale sur un support informatique. La figure 2.8 illustre un exemple de calcul de l'orientation d'une minutie d'une image d'empreinte.

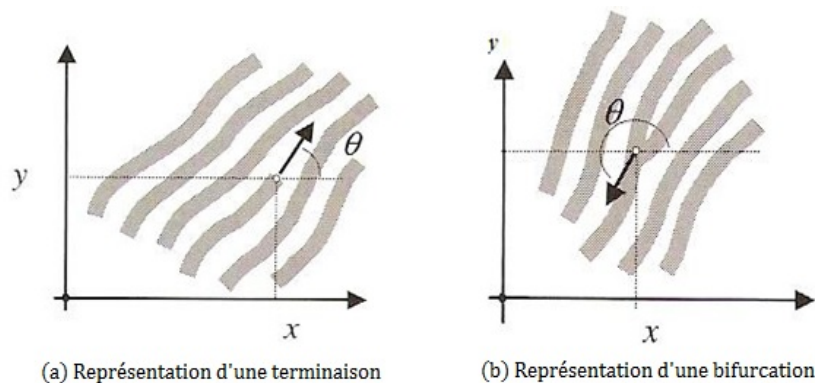


FIGURE 2.8 – Exemple de calcul d'orientation d'une minutie

Le vecteur contenant les informations de (a) sera du style (**terminaison**, $x, y, \theta = 60^\circ$), et celui de (b) (**bifurcation**, $x, y, \theta = 240^\circ$).

C'est l'ensemble de ces éléments qui constitueront la signature de l'image et seront consignés dans un vecteur qui servira à établir la comparaison entre les différents templates ainsi créés.

b) L'algorithme de l'index de Poincaré

Il existe plusieurs algorithmes de détection des caractéristiques globales d'une empreinte digitale [28]. Ce travail, met l'accent sur l'un des plus pratiques, simples et élégants : *l'algorithme de l'index de Poincaré* proposé par Kawagoe et Tojo [86].

Principe

Soit une courbe C dans un champ de vecteurs G . L'index de Poincaré $P_{G,C}$ est la somme des rotations des vecteurs de G le long de C [112, 116]. C'est-à-dire la somme des angles entre les vecteurs adjacents¹ le long de la courbe C comme illustré par la figure 2.9 ci-dessous.

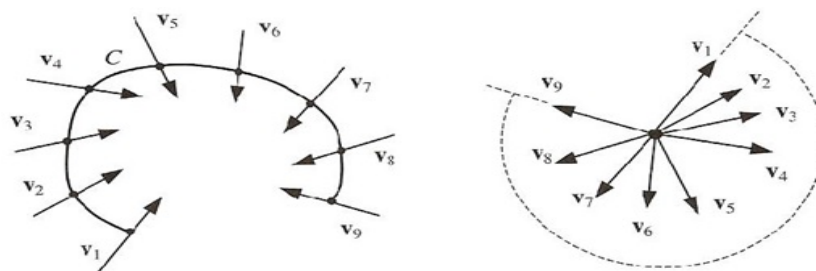


FIGURE 2.9 – L'index de Poincaré calculé sur une courbe C ; $P_{G,C} = -263^\circ$.

Dans le cas des empreintes digitales, G est remplacé par l'orientation de l'empreinte, D . Soit un pixel $[i, j]$ de la matrice D ; l'index de Poincaré $P_{G,C}(i, j)$ est calculé sur une courbe fermée C , entourant $[i, j]$, en faisant la somme des différences des orientations de lignes de l'empreinte aux points (pixels) adjacents de C . On considère donc un certain nombre de pixels entourant $[i, j]$ (un voisinage de 8 pixels par exemple) et on somme les angles entre les orientations aux pixels adjacents. Pour ce faire, chaque orientation se voit attribuée l'une des deux directions possibles (rappelez-vous que les lignes d'une empreinte ne sont pas dirigées).

1. L'angle entre 2 vecteurs se calcule en plaçant les deux vecteurs à un même point d'origine et en faisant la différence des angles qu'ils forment avec l'axe horizontal.

/	—	—	—	\
	$d_2 \nearrow$	$d_3 \rightarrow$	$d_4 \searrow$	
	$d_1 \uparrow$	$[i, j]$	$d_5 \downarrow$	
	$d_0 \nwarrow$	$d_7 \leftarrow$	$d_8 \swarrow$	
\	—	—	—	/

(a) $P_{G,C} = 360^\circ$

/	—	—	—	\
	$d_2 \nearrow$	$d_3 \rightarrow$	$d_4 \searrow$	\downarrow
	$d_1 \uparrow$	$[i, j]$	$d_5 \downarrow$	\downarrow
	$d_0 \downarrow \uparrow$	$d_7 \downarrow$	$d_8 \downarrow$	
\				/

(b) $P_{G,C} = 180^\circ$

/	/	/	\	\
/	$d_2 \nearrow$	$d_3 \nearrow$	$d_4 \nwarrow$	\
/	$d_1 \nearrow$	$[i, j]$	$d_5 \nwarrow$	\
/	$d_0 \nearrow \swarrow$	$d_7 \swarrow$	$d_8 \nwarrow$	\
/	/	—	\	\

(c) $P_{G,C} = -180^\circ$

Les matrices ci-dessus montrent un index de Poincaré calculé avec un voisinage de 8 pixels autour du pixel $[i, j]$ sélectionné. Le pixel de la matrice (a) appartient à une zone singulière en spire, tandis que celui de la matrice (b) appartient à une zone en boucle, et celui de la matrice (c) appartient à une zone singulière en delta. Pour la boucle et le delta (matrices (b) et (c)), on voit bien les deux choix possibles de la direction des lignes des empreintes au premier pixel d_0 .

Remarques

En fait, l'index de Poincaré sur une courbe fermée admet toujours l'une des valeurs suivantes : 0° , $\pm 180^\circ$, et $\pm 360^\circ$. Pour ce qui est des empreintes digitales, le pixel $[i, j]$ se trouve dans telle ou telle région singulière selon la valeur de l'index de Poincaré à ce point :

1. Si $P_{G,C} = 0^\circ$, alors $[i, j]$ ne se trouve pas dans une région singulière.
2. Si $P_{G,C} = 360^\circ$, alors $[i, j]$ se trouve dans une région singulière en spire.
3. Si $P_{G,C} = 180^\circ$, alors $[i, j]$ se trouve dans une région singulière en boucle.
4. Si $P_{G,C} = -180^\circ$, alors $[i, j]$ se trouve dans une région singulière delta.

Les matrices (a), (b) et (c) ci-dessus montrent trois régions singulières où le pixel $[i, j]$ est entouré de 8 éléments $d_k (k = 0..7)$ de la courbe C ; d_0 se voit assigné une des deux directions possibles (c'est-à-dire la flèche est placée à l'une des extrémités de d_0); ensuite $d_k (k = 1..7)$ est dirigé de telle sorte que la valeur absolue de l'angle entre d_k et d_{k-1} soit inférieure ou égale à 90° .

L'index de Poincaré est alors calculé par la formule suivante [116] :

$$P_{G,C}(i, j) = \sum_{k=0}^7 \text{angle}(d_k, d_{(k+1) \bmod 8}) \quad (2.3)$$

La figure 2.10 illustre un exemple de détection de régions singulières par l'algorithme de Poincaré. L'image (b) de la figure 2.10 montre que les points appartenant à une région

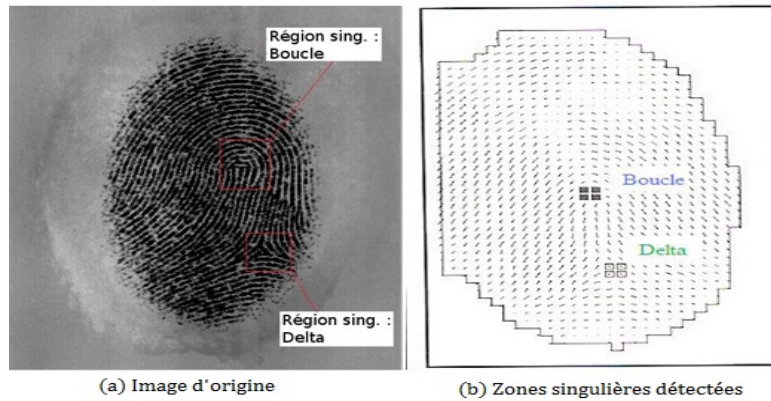


FIGURE 2.10 – Détection d'une région singulière par l'algorithme de Poincaré

singulière sont entourés de petits carrés remplis ($P_{G,C} = 180^\circ \Rightarrow$ boucle) et non remplis ($P_{G,C} = -180^\circ \Rightarrow$ delta). En pratique, plusieurs points sont détectés (ici 4) dans une région singulière.

2.2.1.3 Caractéristiques fréquentielles d'une empreinte

L'une des particularités d'une image d'empreinte digitale est qu'elle peut avoir localement des lignes parallèles. Ces lignes sont notamment appréciées dans le domaine fréquentiel et permettent de déterminer les caractéristiques fréquentielles d'une empreinte digitale. En effet la transformée de Fourier [107] de lignes parallèles va donner idéalement deux fonctions symétriques par rapport à l'origine, centrée sur la fréquence du motif de base et orientées perpendiculairement à la direction des lignes parallèles.

Soit \bar{d} la distance moyenne entre les stries présentes dans l'image. Il est communément admis que la distance inter-strie locale au sein d'une image est globalement constante et ne subit que de légères variations ($d_{min} \leq \bar{d} \leq d_{max}$) notamment au niveau des minuties. Localement la distance inter-strie d est associée à la direction des stries et va donc provoquer dans le spectre de l'image un pic de fréquence r de direction perpendiculaire à celle des stries.

Si N correspond à la largeur du spectre de l'image, on a : $r_{min} \leq \bar{r} \leq r_{max}$, avec :

$$r_{min} = \frac{N}{d_{max}}, \quad r_{max} = \frac{N}{d_{min}} \quad \text{et} \quad \bar{r} = \frac{N}{\bar{d}}$$

L'ensemble des pics de fréquence va alors produire théoriquement un anneau centré contenant toute l'information utile de l'image, le reste étant du bruit [107].

La valeur de \bar{d} varie selon les personnes et selon leur âge, elle évolue avec la croissance pour se stabiliser à l'âge adulte. De plus elle peut varier localement durant l'étape d'acquisition en raison de l'élasticité de la peau. Pour beaucoup de techniques de pré-traitement la détermination de ce paramètre est très critique, car une mauvaise estimation risque d'éliminer des fréquences importantes et donc de dégrader l'image. Généralement \bar{d} est estimé statistiquement sur une population donnée, le cas des enfants n'étant pas pris en compte.

2.2.1.4 Binarisation de l'image d'empreinte digitale

Afin de permettre la squelettisation, l'image doit d'abord être binarisée ; c'est à dire que l'image numérique avec 256 niveaux de gris obtenue après le scannage de l'empreinte est transformée en image binaire où les pixels noirs correspondent aux stries et les pixels blancs aux vallées.

Il existe de nombreuses techniques de binarisation d'images, cette étape n'occupant pas de façon générale une place très importante dans un système de reconnaissance d'empreinte digitale, la littérature [146] suggère d'utiliser une méthode de seuillage simple. Pour effectuer ce traitement la valeur de chaque pixel $P(i, j)$ est comparée à un seuil M et si cette valeur est supérieure au seuil, alors le pixel prend la valeur 1 (blanc), sinon, il prend la valeur 0 (noir). La figure 2.11 ci-dessous illustre un exemple d'image d'empreinte binarisée.

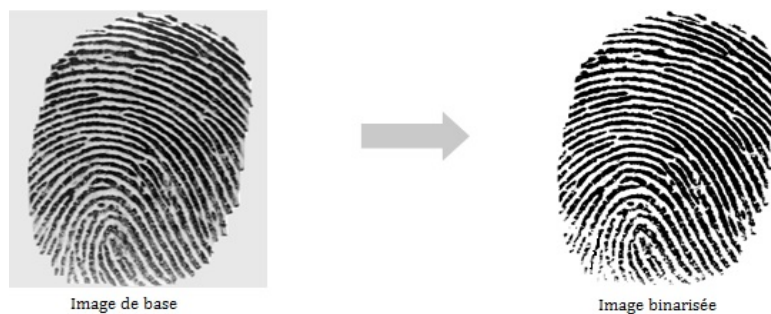


FIGURE 2.11 – Image binarisée avec des tailles fixes.

Le seuil peut être fixé dès le départ (on parle alors de seuillage global). Dans ce cas la binarisation s'effectue directement en sortie de l'étape de filtrage en comparant la valeur

filtrée obtenue au seuil fixé. Ceci permet d'éviter un nouveau balayage de l'image en insérant un simple comparateur en sortie du filtre. Pour le choix du seuil, il est à noter que, dans les différentes méthodes de filtrage développées à la section 2.2.1.6 ci-dessous, les filtres utilisés ont une composante continue nulle [146]. Ceci permet d'annuler la valeur moyenne du bloc filtré et de l'image, et donc de choisir $M = 0$. En réalité la valeur moyenne des blocs filtrés n'est pas nulle car le bloc filtré extrait ne tient pas compte du voisinage, néanmoins on s'en approche.

Le seuil peut également être calculé localement (seuillage local adaptatif) [40]. Dans ce cas l'image est divisée en N blocs et pour chaque bloc, la moyenne des pixels du bloc est calculée, le bloc est ensuite binarisé en fonction de cette moyenne qui va correspondre au seuil. Cette technique est moins rapide que la précédente, car elle nécessite un nouveau balayage complet de l'image et le calcul de N seuils.

2.2.1.5 Squelettisation de l'image d'empreinte ou détection des régions singulières

Dans l'image binarisée (noir et blanc) les lignes se voient plus clairement mais ont des tailles différentes. pour pouvoir détecter rapidement les minuties (terminaison, bifurcation, ...), il est nécessaire d'obtenir une image plus schématique de l'empreinte, dans laquelle toutes les lignes auront la même épaisseur : c'est l'opération appelée *squelettisation*. Pour terminer le pré-traitement de l'image d'empreinte digitale, on réalise une squelettisation, comparable à une opération d'amincissement.

En prenant une image binaire comme l'entrée, cette opération supprime les pixels avant-plans sélectionnés comme l'*érosion*, ce qui rend les minuties plus facilement détectables comme le montre la figure 2.12 ci-dessous.

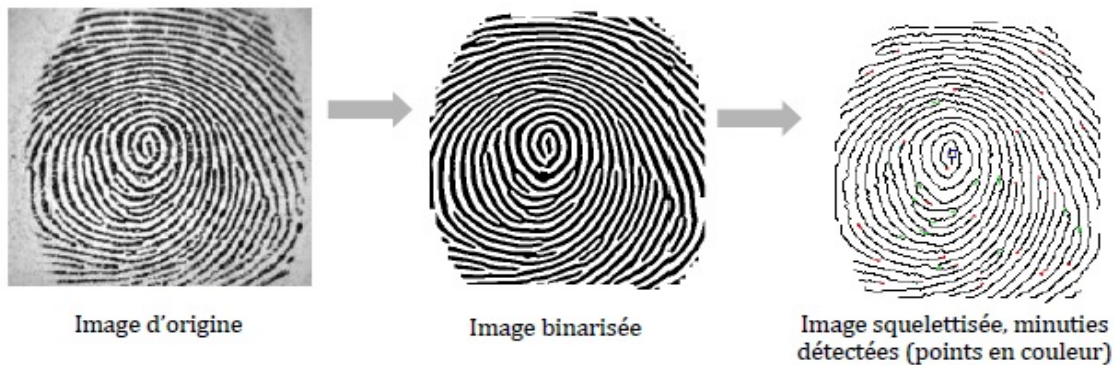


FIGURE 2.12 – Squelettisation d'une empreinte digitale

La méthode classique

Le principe de cette procédure est d'effectuer une succession d'opérations d'érosion conditionnelle, jusqu'à ce que le but recherché soit atteint [80]. L'érosion conditionnelle signifie qu'un pixel d'intensité non nulle ne peut être éliminé que si, une fonction logique déterminée des valeurs des pixels qui l'entourent, est vérifiée. Cette fonction logique a pour rôle de vérifier la participation ou non du pixel central au maintien de la continuité du tracé.

La procédure de squelettisation consiste donc, sur la base de ce principe, à balayer l'image, en passes successives, au moyen d'une fenêtre de dimension minimum 3×3 comme illustré par la figure 2.13. Le pixel du centre de cette fenêtre prend alors une

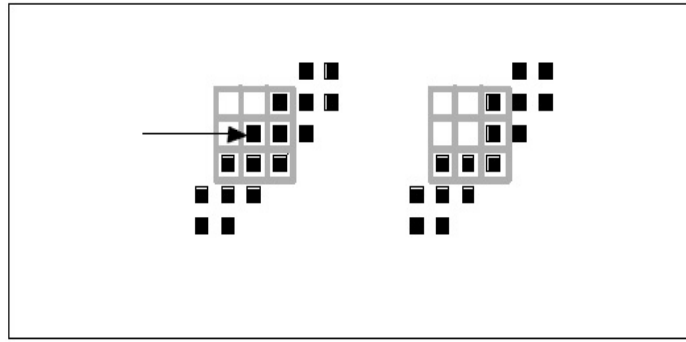


FIGURE 2.13 – Fenêtre d'érosion conditionnelle

valeur déterminée par une fonction logique de l'ensemble des 9 points de la fenêtre. Le balayage de l'image s'effectue de haut en bas et de gauche à droite, et le processus se répète autant de fois que nécessaire. L'arrêt se produit lorsque l'ensemble de la matrice qui contient l'image d'empreinte a été balayée sans qu'aucun nouveau pixel n'ait pu être éliminé. La fonction logique d'élimination des pixels doit être établie en examinant cas par cas les 256 (2^8) situations distinctes qui peuvent se présenter, en fonction des valeurs binaires des 8 pixels qui entourent le pixel central de la fenêtre. Ce dernier doit évidemment être d'intensité non nulle pour pouvoir être éventuellement éliminé, et la fonction logique n'est simplement pas évaluée dans le cas contraire. Diverses contraintes régissent la conception de cette fonction logique.

Comme l'objectif est d'obtenir un tracé final d'épaisseur unique aussi fidèle que possible au tracé initial, il est souhaitable que l'élimination des pixels se fasse de manière symétrique, de façon à déterminer la ligne médiane de ce dernier. De plus, le résultat de la procédure de squelettisation doit être *isotropique*, c'est-à-dire invariant aux rotations. Enfin, la vitesse d'exécution de l'ensemble de la procédure doit être la plus élevée possible.

Le problème qui se pose aux abords d'une image lors d'une opération de filtrage apparaît ici également, puisque la fonction logique fait alors intervenir des valeurs de pixels inexistantes. Toutefois il est impératif d'effectuer cette opération de squelettisation sans que la dimension de l'image ne soit réduite.

La méthode du "hit-and-miss"

La traduction de *hit – and – miss* est une méthode pour la détection de forme. Elle exige deux éléments de structure : un élément définit les dispositifs de premier plan, tandis que l'autre définit les dispositifs de fond pour détecter les zones d'érosion dans l'image. A proprement parler, l'un pourrait être dérivé de l'autre (le premier plan est toujours l'opposé du fond). Le *hit – and – miss* (HTM) est défini comme intersection de l'érosion de l'image en entrée (notée A) par le premier élément de structure et de l'érosion du complément de A par le deuxième élément de structure [78] :

$$HTM = intersection(erosion(A, SE1), erosion(\tilde{A}, SE2))$$

Ainsi, la sortie d'une transformation *hit – and – miss* est l'ensemble des points qui matchent l'érosion de l'image A par $SE1$ (le premier plan de l'image) avec ceux qui matchent l'érosion du complément de l'image A (notée \tilde{A} avec $SE2$ (le fond)). Le résultat est le mappage de certains dispositifs de bord. Puisque les SEs sont des dispositifs permettant la réalisation d'une fonction binaire, ils devraient être des compléments de l'un et de l'autre : si on a un "1" dans $SE1$, on doit avoir un "0" dans l'endroit correspondant dans $SE2$. L'opération d'amincissement morphologique est définie comme intersection de l'image A et le complément du hit-and-miss de A avec $SE1$ et $SE2$ dans le premier plan et le fond [1, 78] :

$$Amincissement = intersection(A, \widetilde{HTM}(A, SE1, SE2))$$

Dans la pratique, l'amincissement est appliqué en utilisant un ordre des SE , où la définition ci-dessus est appliquée itérativement en utilisant chaque paire de SE dans l'ordre. Le nombre maximum d'itérations varie en fonction de l'utilisateur.

Ces opérations d'amincissement sont parfois à l'origine de la détection de fausses minuties qui sont susceptibles de gêner considérablement les performances du système si elles sont considérées comme étant des bonnes minuties. C'est pourquoi il est important de les éliminer lorsqu'elles se manifestent. Abu Shamim et al. proposent une technique assez novatrice dans son article [6, 7].

Par ailleurs, le filtrage de l'image permet entre autre, de ne récupérer que l'information utile après la squelettisation.

2.2.1.6 Filtrage de l'image

L'un des buts du filtrage est de "nettoyer" le signal en éliminant le plus de bruit possible tout en préservant le maximum d'informations. En outre, l'information contenue dans un signal n'est pas forcément entièrement pertinente : il faut "sélectionner" l'information utile suivant l'usage que l'on veut en faire. Une autre finalité du filtrage est donc de sélectionner et de renforcer certaines bandes de fréquences porteuses de l'information intéressante.

Le filtrage des images a la même finalité que celui des signaux 1D. Il s'agit essentiellement d'enlever le bruit (parasite) ou de sélectionner certaines fréquences. Si la notion de haute fréquence ou basse fréquence est naturelle en signal 1D (son aigu ou grave), la fréquence spatiale est un concept plus délicat qui découle du fait que les images appartiennent au domaine spatial. La fréquence est une grandeur qui caractérise le nombre de phénomènes qui se déroulent au cours d'un temps donné [116].

i) Principe générale du filtrage

Dans une image, les détails se répètent fréquemment sur un petit nombre de pixels, on dit qu'ils ont une *fréquence élevée* : c'est le cas pour les bords et les contours dans une image. Au contraire, les *fréquences basses* correspondent à des variations qui se répètent peu car, diluées sur de grandes parties de l'image, par exemple des variations de fond de ciel. Le bruit d'une image désigne les pixels dont l'intensité est très différente de celle de ses voisins. Le bruit peut provenir des différentes méthodes d'acquisition de l'image (la façon dont est scannée l'empreinte), la saleté présente sur le doigt lors du scanning, la coopération de la personne pendant l'opération d'acquisition de l'image d'empreinte, etc.

Si f est l'image à filtrer (ou à rehausser) et g le filtre spatial (ou *PSF* - Point Spread Function ou masque) on a :

$$f(x, y) * g(x, y) = F^{-1} \{F(f(x, y))\} \tag{2.4}$$

$F(g(x, y))$ est la fonction de transfert du filtre (*FFT*) ou (*File Function Transfert* en anglais), x et y les pixels de l'image à filtrer.

Le noyau de convolution (masque PSF) du filtre k est à support compact inclus dans $[x_1, x_2] \times [y_1, y_2]$ [107] :

$$g(x, y) = (f \times k)(x, y) = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} f(x - i, y - j)k(i, j). \tag{2.5}$$

Généralement le filtre est de dimension d_i impaire et symétrique. Dans ce cas :

$$[x_1, x_2] = \left[-\frac{d_1}{2}, \frac{d_1}{2}\right] \text{ et } [y_1, y_2] = \left[-\frac{d_2}{2}, \frac{d_2}{2}\right]$$

ii) Filtrage sur la FFT

Principe

Le principe du filtrage sur la FFT d'une image consiste à calculer la transformée de fourrier F de l'image d'origine f , ensuite on obtient N_d nouvelles images TF_i en multipliant F par les masques H_i puis on calcule leur transformée inverse. On dispose alors de N_d images Tf_i dont chacune est améliorée dans la direction qui lui est adoptée.

$$Tf_i = FFT^{-1}(TF_i) = FFT^{-1}(F \times H_i) = FFT^{-1}[FFT(f) \times H_i]$$

Cette approche est très coûteuse en espace mémoire et en opérations de calcul. Pour

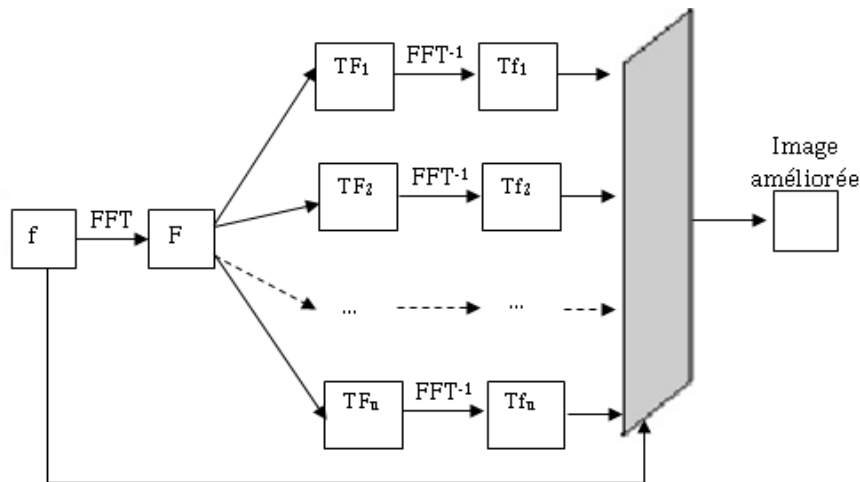


FIGURE 2.14 – Principe du filtrage sur la FFT

accélérer les calculs, on peut utiliser l'algorithme de la transformée de fourrier rapide (TFR) qui est optimal lorsque la dimension de l'image est un multiple d'une puissance de 2. Les images ayant une taille $H < 256$ sont complétées par des pixels blancs pour former une image carrée de 256 pixels de large [85, 107].

Pour construire l'image finale la carte directionnelle de l'image d'origine est sollicitée. En effet chaque pixel (x, y) appartient à un bloc de direction θ_i , on peut donc lui faire

correspondre le pixel $TF_i(x, y)$ puisque TF_i correspond à l'image d'origine améliorée dans la plage de directions contenant θ_i . Etant donné que l'on dispose de toutes les images un meilleur résultat peut être obtenu en effectuant une interpolation des deux images les plus proches de θ_i . Ceci permet d'éliminer les effets de bord entre les blocs adjacents.

$$I(x, y) = a(x, y).TF_{j(x,y)}(x, y) + (1 - a(x, y)).TF_{k(x,y)}(x, y) \quad (2.6)$$

avec

$$j(x, y) = \left\lfloor \frac{\theta(x, y)}{\pi/N} \right\rfloor, k(x, y) = (j(x, y) + 1) \equiv N$$

et

$$a(x, y) = \frac{\theta(x, y)}{\pi/N} - j(x, y)$$

Bilan

Cette technique de filtrage fournit de bons résultats y compris sur des images de qualité moyenne. Néanmoins dans le cadre du développement d'une application avec des ressources limitées elle est peu adaptée. En effet elle nécessite d'importantes ressources mémoires car il faut pouvoir stocker N_d (où N_d représente le nombre de directions d'une image) images temporaires de taille 256×256 contenant des nombres réels codés sur 4 octets. De plus, elle est coûteuse en temps de calcul car les masques de taille 256×256 sont calculés en temps réel.

iii) *Filtrage spatial (Filtre de Gabor)*

a) Principe du filtrage spatial

La méthode précédente fonctionne correctement mais nécessite d'importantes ressources. Ici, c'est le filtrage spatial qui nous intéresse. L'image est tout d'abord découpée en blocs de taille fixe de telle sorte que le bloc contienne suffisamment d'information utile (orientation et fréquence des stries). Pour chaque bloc B on estime la direction θ qui lui est associée au moyen de la carte directionnelle définie plus haut, ensuite un filtre directionnel passe bande F_θ lui est associé. Ensuite chaque bloc est convolué avec le filtre correspondant comme suit [170] :

$$FB_\theta(i, j) = \sum_{k=1}^N \sum_{l=1}^N B(i, j).F_\theta(i - k, j - l) \quad (2.7)$$

où $B(i, j)$ représente le bloc du pixel (i, j) .

Pour éviter les effets de bord le voisinage du bloc est utilisé au cours du filtrage. Il est évident que plus la taille des blocs est grande, plus l'orientation des stries varie à l'intérieur du bloc. Ceci a pour effet de diminuer la fiabilité de l'estimation directionnelle, et va également provoquer une dégradation de l'image car l'ensemble des directions n'est pas pris en compte par le filtre. Par contre, plus les blocs sont petits et plus les temps de calculs sont importants. Le choix des blocs pour effectuer l'opération de filtrage constitue donc un compromis. La figure 2.15 ci-dessous illustre un exemple d'application d'un filtre spatial sur un bloc d'une image avec des stries de longueurs variables. L'effet indésirable qui se produit est une dégradation visuelle de l'image avec un risque de la diminution de l'estimation directionnelle.

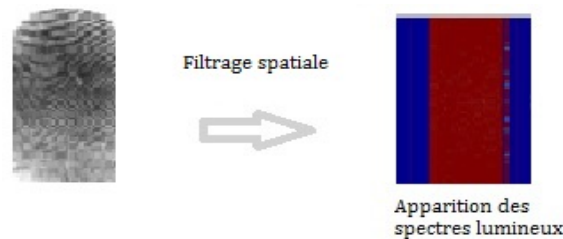


FIGURE 2.15 – Application du filtrage spatial sur un bloc d'image

b) Principe du filtrage de Gabor

Les filtres de *Gabor* sont beaucoup plus utilisés pour l'analyse de direction [56]. Ce sont des filtres orientés passe-bande. Les paramètres de ce filtre sont : l'échelle de l'enveloppe Gaussienne (σ_a, σ_b) , l'angle du filtre, la fréquence et la phase de la sinusoïde (w_0, P_0) .

Un filtre de *Gabor* est un filtre linéaire dont la réponse impulsionnelle est une sinusoïde modulée par une fonction gaussienne (également appelée ondelette de *Gabor*) [109, 112]. Il porte le nom du physicien anglais d'origine hongroise *Dennis Gabor*. Le filtre de *Gabor* est directement lié aux ondelettes de *Gabor*, puisqu'ils peuvent être conçus pour le nombre d'érosions et de rotations. Cependant, généralement l'expansion n'est pas appliquée pour des ondelettes de *Gabor*, puisque ceci exige le calcul des ondelettes bi-orthogonales, qui peuvent prendre beaucoup de temps [116]. Les relations entre les activations pour un endroit spatial spécifique sont très distinctives entre les objets dans une image.

Le filtre spatial de *Gabor* est composé du produit d'un filtre passe-bas gaussien agissant suivant l'orientation locale des stries et d'un filtre passe bande agissant perpendiculairement aux stries (filtre de *Gabor* à une dimension)

$$g(x) = \exp(2j\pi fx + \theta) \times \exp\left(-\frac{(x - x_0)^2}{\sigma_x}\right) \tag{2.8}$$

En outre, des activations importantes peuvent être extraites à partir de l'espace de *Gabor* afin de créer une représentation clairsemée d'objets. Ce qui donne en deux dimensions [56, 170] :

$$G(x, y, \theta, f) = \exp\left\{-\frac{1}{2}\left[\frac{(x \cos \theta)^2}{\sigma_x^2} + \frac{(y \sin \theta)^2}{\sigma_y^2}\right]\right\} \cos(2\pi fx \cos \theta) \tag{2.9}$$

où θ est l'orientation et f est la fréquence ; σ_x et σ_y sont les constances d'enveloppe *Gaussien* pour les pixels x et y respectivement. L'application d'un filtre de Gabor g de masque M de rayon r à une image I de largeur m et de hauteur n , se résume donc à la formule suivante :

$$g(I) = J = M \times I$$

où J est une matrice de dimension m, n et pour $i, j \in \mathbf{N}, r \leq i < m-r$ et $r \leq j < n-r$:

$$J_{i,j} = \sum_{k=-r}^r \sum_{l=-r}^r M_{k,l} \times I_{(i-k),(j-l)} = \sum_{k=-r}^r \sum_{l=-r}^r G(k,l,\theta,f) \times I_{(i-k),(j-l)} \tag{2.10}$$

les filtres de *Gabor* permettent d'isoler les contours d'une image d'orientation perpendiculaire à θ et répondant à une certaine épaisseur, qui dépend de f . Ceci justifie le fait que pour détecter l'ensemble des contours d'une image, on lui applique généralement un ensemble de filtres de Gabor. La figure 2.16 ci-dessous montre un exemple d'application du filtre de Gabor sur deux images d'empreinte digitale avec diverses orientations.

L'application d'un seul filtre avec une orientation horizontale fait ressortir les contours orientés perpendiculairement par rapport à son orientation, alors que l'application à plusieurs orientations permet de détecter des contours plus nombreux. Une application fréquentielle fait ressortir un tracé beaucoup plus net, le filtre est sensible à de plus nombreux contours en termes d'épaisseur.

Enfin, si l'on fait varier σ_x de manière à ce qu'il soit inférieur à σ_y , les filtres deviennent moins sensibles aux contours et le prototype se comporte (dans un état limite) comme un simple flou gaussien avec mise en avant des niveaux de gris. Si l'on applique la variation inverse, le résultat, légèrement plus net, ne présentera pas de grandes différences avec un filtrage symétrique.

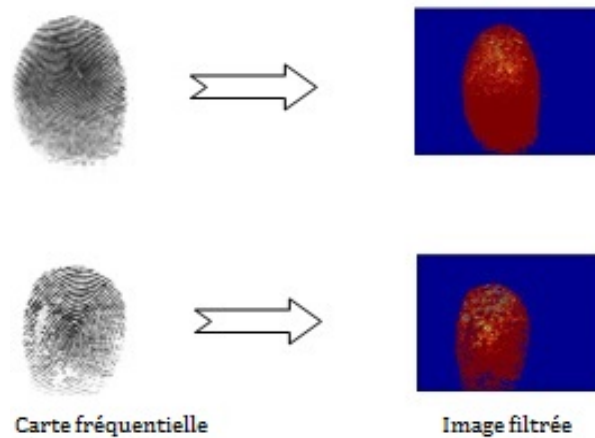


FIGURE 2.16 – Application du filtre de Gabor

Bilan

Cette approche nécessite au préalable l'estimation des cartes directionnelles et fréquentielles. En raison du mode de balayage du capteur utilisé, la possibilité d'une variation importante de la distance inter-strie au sein de l'image impose de calculer les filtres en temps réel ; car les paramètres du filtre sont directement liés aux caractéristiques locales des stries. Bien que les filtres de Gabor aient prouvé leur efficacité, ils souffrent de quelques défauts. En effet nous avons vu que leur bande passante maximale est limitée à une octave (4 octets), ce qui nécessite de les recalculer pour chaque bloc de l'image afin de les adapter aux caractéristiques locales. Ceci est coûteux en nombre d'opérations. De plus cela rend la méthode très sensible au bruit car la bonne estimation de deux paramètres (fréquence et direction locales) est obligatoire pour assurer l'amélioration de chaque bloc.

iv) *Filtrage de Log-Gabor*

L'utilisation d'un filtre de Gabor nécessite d'importantes ressources pour calculer les filtres et le résultat obtenu est fortement dépendant de la bonne estimation de la carte fréquentielle. Une alternative consiste à utiliser à la place un filtre de Log-Gabor [23, 170] dont le masque fréquentiel en coordonnées polaires est défini par l'équation suivante :

$$H(f, \theta) = H_r(f) \times H_\theta(\theta) \tag{2.11}$$

avec :

$$H_r(f) = \exp \left\{ -\frac{[\ln(f/f_0)]^2}{2[\ln(\sigma_r/f_0)]^2} \right\} \tag{2.12}$$

et :

$$H_{\theta}(\theta) = \exp\left\{-\frac{(\theta - \theta_0)^2}{2\sigma_{\theta}^2}\right\} \tag{2.13}$$

où :

- σ_r définit la bande passante radiale en octaves. $B = 2\sqrt{2/\ln 2} \times |\ln(\sigma_r/f_0)|$;
- f_0 la fréquence centrale ;
- θ_0 correspond à la direction du filtre ;
- σ_{θ} définit la bande passante angulaire du filtre.

Contrairement aux filtres de Gabor simples, les filtres de Log-Gabor ont toujours une composante continue nulle et permettent donc de gérer une bande passante fréquentielle supérieure à 4 hertz.

Bilan

Afin de pallier aux faiblesses des filtres de Gabor, les filtres de Log-Gabor ont été introduits. Ces derniers autorisent une bande passante fréquentielle beaucoup plus importante, ce qui permet d'éviter la dépendance du filtre de la fréquence locale. L'ensemble des filtres peut alors être calculé à l'extérieur du système pour y être ensuite stocké. Cela permet d'accélérer grandement les calculs. En contrepartie une augmentation de l'espace mémoire et une dégradation de la sélectivité en fréquence sont enregistrés.

v) Filtrage fréquentielle auto-adaptatif

Comme cela a déjà été vu, la majorité des algorithmes de traitement de l'image est basée sur l'estimation de l'orientation et de la fréquence locale des stries. Ces estimations ont l'obligation d'être très fiables pour assurer un filtrage correct, mais dans le cas d'une image très bruitée, ces estimations sont plus qu'aléatoires et peuvent provoquer l'effet inverse de celui recherché (une dégradation de l'image). *Wiilis* et *Myers* [165] proposent une autre méthode qui ne nécessite pas ces estimations et s'avère plus efficace pour traiter les images, principalement celles de mauvaise qualité.

Principe

De manière similaire au calcul d'une carte directionnelle, l'image est divisée en blocs de telle sorte que les stries d'un bloc soient localement parallèles, puis la Transformée de Fourier Discrète [25, 107] est calculée pour chaque bloc comme suit :

$$F(u, v) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y).e^{-i2\pi(ux+vy)/N} \tag{2.14}$$

Pour $(u, v) \in [0, N - 1]^2$.

La composante continue $F(0, 0)$ est d'abord annulée, puis chaque pixel du bloc fréquentiel obtenu est multiplié par une puissance p de son amplitude comme suit :

$$G(u, v) = F(u, v) \times |F(u, v)|^p \tag{2.15}$$

C'est cette multiplication qui permet d'améliorer le bloc. Finalement la partie réelle de la transformée de Fourier Discrète est calculée ci-dessous pour chaque nouveau bloc et fournit l'image finale :

$$g(x, y) = \Re \left(\frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} G(u, v) \cdot e^{-i2\pi(ux+vy)/N} \right) \tag{2.16}$$

Pour $(x, y) \in [0, N - 1]^2$.

La figure 2.17 ci-dessous montre un exemple d'application du filtrage fréquentiel auto-adaptatif sur deux images d'empreinte digitale avec diverses orientations.

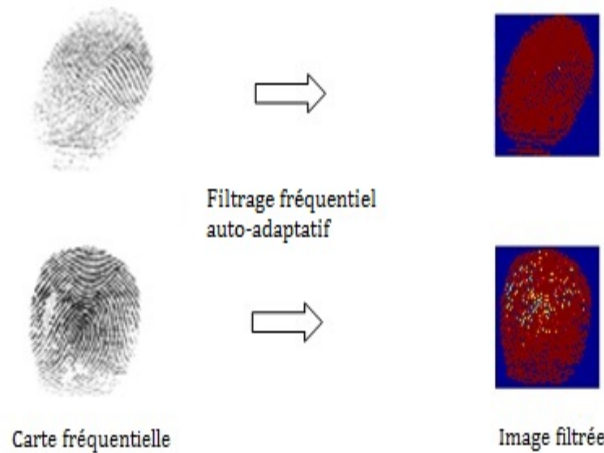


FIGURE 2.17 – Application du filtre fréquentiel auto-adaptatif

Bilan

Contrairement aux approches précédentes, ce filtre est indépendant de toute estimation fréquentielle ou directionnelle, ce qui s'avère très efficace dans le cas des images de très mauvaise qualité. Cependant, certaines lignes verticales ont tendance à s'introduire même après l'amélioration de l'image car on ne privilégie aucune fréquence particulière. Pour gérer ce problème, une étape supplémentaire ayant pour but de lisser la carte filtrée doit être ajoutée.

Si les méthodes de filtrage permettent parfois d'améliorer la qualité de représentation de l'image, un soin suffisant apporté à l'empreinte ainsi que de bonnes conditions d'acquisition, lorsqu'il est possible de les réaliser, demeurent malgré tout, les meilleurs garanties de l'obtention de bonnes performances de reconnaissance.

2.2.2 La détection des minuties

La détection de minuties est une tâche cruciale dans la reconnaissance d'empreinte digitale car le processus d'identification de presque tous les systèmes de reconnaissance automatique des empreintes consiste principalement en l'appariement de minuties (*minutiae matching* en anglais). Plusieurs méthodes ont été développées [109, 112, 130] pour réaliser une extraction efficace des minuties, ceci pouvant s'expliquer par le fait qu'il existe un grand nombre de types de minuties.

En effet, bien que certains types de minuties soient plus utilisés que d'autres, une société qui développe un système automatique de reconnaissance d'empreintes digitales peut choisir d'adopter une technique qui s'applique à tel ou tel type de minuties pour des raisons diverses (exemple : problème de budget car il existe des mécanismes plus coûteux à mettre en place que d'autres, se démarquer des autres, ...).

Toutefois, même si ces méthodes se différencient les unes des autres, il convient de noter quelques-uns de leurs points communs tels que le fait qu'elles traitent toutes une image de l'empreinte en niveaux de gris (c'est à dire pas en couleur) et que la plupart d'entre elles nécessitent la binarisation de l'image en niveaux de gris (la conversion de l'image en une image en deux couleurs : noir et blanc). Cependant, certaines méthodes procèdent à l'extraction des minuties en traitant directement l'image en niveaux de gris sans passer par la binarisation [28, 33].

Les différentes techniques de détection de minuties peuvent donc être classées en deux grandes catégories : celles qui se basent sur la binarisation de l'image et celles qui travaillent directement sur l'image. Dans la plupart des ouvrages on parle de *Binarization-based methods* et *Direct gray – scale methods* [112, 130].

2.2.2.1 Méthodes basées sur la binarisation

Etant donné que les minuties sont de très petits détails de l'empreinte difficilement visibles, il convient de bien les mettre en évidence avant de procéder à leur extraction. C'est la grande utilité des opérations de binarisation et de squelettisation de l'image de l'empreinte. L'idée consiste à donner une même intensité aux lignes et une autre intensité très différente aux vallées, afin de bien distinguer ces deux régions de l'empreinte (voir section 2.2.1.4). Ainsi, ayant une faible intensité au départ, les lignes de l'empreinte

seront en noir (valeur 0) et les vallées seront en blanc (plus forte intensité car $1 > 0$).

Une fois l'image binarisée, on procède à la squelettisation de celle-ci exactement comme à la section 2.2.1.5. A ce stade, les minuties sont bien visibles et facilement détectables par n'importe quelle méthode classique.

Limites

- il est possible de perdre beaucoup d'informations lors du processus de binarisation ;
- la squelettisation peut introduire un certain nombre de fausses minuties ;
- appliquées sur une empreinte de mauvaise qualité, ces méthodes produisent de mauvais résultats.

Il existe cependant certains processus d'amélioration de l'image avant de procéder à l'extraction de minuties, "*enhancement step*" [112]. Le système peut éventuellement rejeter l'image de l'empreinte lorsque le contrôle de qualité révèle une image de très mauvaise qualité.

Ces méthodes restent tout de même très pratiques car elles sont faciles à mettre en place et donc moins coûteuses. Elles sont souvent combinées avec d'autres méthodes plus complexes pour assurer la fiabilité et la robustesse du système [5].

2.2.2.2 Méthodes d'extraction directes

A l'inverse des techniques précédemment décrites, ces méthodes n'effectuent ni la binarisation ni la squelettisation. L'idée est d'utiliser directement l'image filtrée pour en extraire les minuties. Après les opérations de pré-traitement classiques (calcul de la carte fréquentielle et directionnelle), l'image est filtrée en appliquant un des filtres présentés à la section 2.2.1.6 qui ne tient pas compte de la binarisation. Pour aboutir à une image qui s'apparente à celle obtenue après binarisation et squelettisation, on pourrait avoir besoin de filtrer l'image d'origine plus d'une fois, ceci avec des filtres différents.

Il apparaît évident que les images obtenues en combinant les opérations de filtrage avec celles de binarisation et de squelettisation offriront une meilleure aisance pour effectuer l'extraction et la détection des minuties. Cependant il est tout à fait possible de procéder directement à ces opérations de détection des minuties sur une image qui n'a subi que les opérations de filtrage.

Limites

- les opérations de filtrage sont très coûteuses en temps de calcul ;
- étant donné que l'on peut être amené à effectuer plusieurs filtrages, cela implique un surcoût en terme de nombre de calculs à opérer sur les blocs individuels de l'image ;

- les opérations de filtrage (filtrage spatial principalement) sont très sensibles au bruit ce qui peut entraîner la perte de certains éléments essentiels de l'image.

Cependant, le choix judicieux du(des) filtre(s) à appliquer dans le bon ordre s'avère très efficace surtout pour le cas des images de très mauvaise qualité.

2.2.2.3 Détection des minuties par crossing number

Cette partie met l'accent sur la première catégorie (les méthodes basées sur la binarisation) car c'est la plus utilisée et la plus simple à mettre en œuvre.

Le *crossing number* $CN(p)$ [5] d'un pixel p est défini comme la moitié de la somme des différences en valeur absolue des pixels adjacents dans un voisinage de 8 pixels entourant p . En analysant le squelette binaire de l'image de l'empreinte, on remarque que les pixels correspondant aux minuties possèdent un *crossing number* différent de 2. Le *crossing number* d'un pixel p se calcule par la formule suivante :

$$cn(p) = \frac{1}{2} \sum_{i=1}^8 |val(p_{i \bmod 8}) - val(p_i - 1)| \tag{2.17}$$

Où : p_0, p_1, \dots, p_7 sont les 8 pixels au voisinage de p et $val(p)$ vaut 0 ou 1 (car l'image est binarisée).

Il est alors facile de remarquer qu'un pixel p dont $val(p) = 1$:

- est un point d'une ligne de l'empreinte si $cn(p) = 2$;
- correspond à une terminaison si $cn(p) = 1$;
- correspond à une minutie plus complexe (bifurcation, îlot,...) si $cn(p) \geq 3$.

La figure 2.18 ci-dessous illustre le calcul du crossing number de quelques minuties et montre que : (a) point d'une ligne de l'empreinte, (b) terminaison, (c) bifurcation.

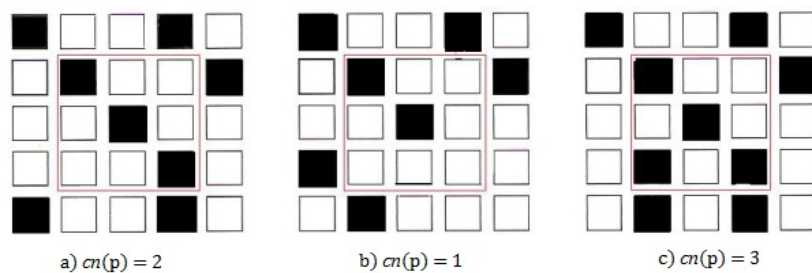


FIGURE 2.18 – Calcul du crossing number dans un voisinage de 8 pixels (cadre rouge)

Les pixels de valeur 1 et ceux de valeur 0 sont respectivement symbolisés par les carrés noirs et les carrés blancs.

2.3 *La comparaison d'empreintes digitales*

Un système de reconnaissance automatique d'empreintes digitales renvoie un résultat positif ou négatif après avoir comparé l'empreinte à identifier à toutes les autres empreintes qu'il possède dans sa base de données. Comme dans la plupart de la littérature ([112, 130] par exemple), l'empreinte à identifier sera représentée par I (pour *input*) et celles de la base de données par T (pour *template*). La comparaison de deux empreintes n'est pas une tâche facile. En effet, plusieurs impressions d'une même empreinte peuvent donner des images très différentes selon que :

- un même doigt est positionné différemment sur la surface de contact du lecteur d'empreintes (un scanner en général). Ainsi, une impression peut subir une légère translation et/ou rotation par rapport à une autre ;
- une partie du doigt est placée en dehors de la surface de contact du lecteur. Lorsque le système compare le petit bout d'image de l'empreinte avec une autre image correctement imprimée de la même empreinte, le résultat est négatif alors qu'il s'agit bel et bien du même doigt. Par exemple, un déplacement du doigt de 2mm (imperceptible à l'œil humain) provoque une translation d'à peu près 40 pixels dans une image scannée à une résolution de 500 dpi (dots per inch) [112] ;
- un même doigt est plus ou moins appuyé contre la surface de contact du lecteur lors de différentes impressions. A cause de la plasticité² des doigts, une empreinte peut être plus ou moins distordue selon l'intensité de la pression du doigt sur le lecteur d'empreintes ;
- le doigt n'est pas propre (entaché par exemple de graisse, de chocolat...).

Tous ces paramètres doivent donc être pris en compte par les algorithmes de comparaison d'empreintes digitales pour avoir un système robuste. C'est ainsi qu'on s'est par exemple rendu compte qu'il est parfois intéressant de pouvoir comparer une image entière correctement imprimée d'une empreinte avec un bout d'image de la même empreinte. On effectuera cette opération en comparant simplement de petites zones de l'empreinte correctement imprimée à celle de l'image imprimée en partie [15].

En effet, sur les lieux de crime par exemple, on ne dispose pas toujours de traces d'empreintes entières. Les enquêteurs doivent donc pouvoir exploiter les petits bouts de traces d'empreintes dont ils disposent. La figure 2.19 illustre quelques exemples de différentes images provenant d'une même empreinte.

2. Caractère de ce qui est plastique, de ce qui peut être modelé

Chaque cadre contient deux images différentes d'une même empreinte.



FIGURE 2.19 – Matching d'empreintes deux à deux

- (a) le doigt était penché à gauche sur la surface de contact du lecteur tandis, (a') le doigt était penché à droite ;
- (b) le doigt a été plus pressé, (b') le doigt a été très légèrement pressé ;
- (c) le doigt a été correctement apposé contre la surface de contact du lecteur, (c') une partie du doigt n'a pas été apposée sur le lecteur.

Dans la plupart de la littérature, les différentes approches de comparaison des empreintes digitales sont classées en trois grandes familles [112, 116, 130]. A savoir : la comparaison basée sur la corrélation, celle basée sur les minuties et enfin la comparaison basée sur les singularités de lignes d'empreintes digitales.

2.3.1 La comparaison basée sur la corrélation

Cette méthode consiste à comparer tout simplement les matrices de pixels des images de deux empreintes, et de calculer la corrélation qui existe entre ces pixels là. L'image à reconnaître (M) tout comme les images constituant la base de données sont toutes numérisées et enregistrées sous forme d'une matrice de pixels notée M_i . La reconnaissance consiste donc à comparer la matrice prise en entrée à toutes les matrices M_i de la base de données pour le cas de l'identification, ou alors à une seule matrice spécifique M_r dans le cas de l'authentification.

Les techniques basées sur la corrélation de pixels ne sont pas très efficaces car nous avons vu que l'on peut obtenir des images très différentes d'une même empreinte digitale (voir figure 2.19), et donc avec des valeurs pixels différentes ; ce qui amènera le système à refuser la correspondance alors que les empreintes sont identiques. Si les images d'une

même empreinte sont différentes, les valeurs de leurs pixels sont d'office différentes. De plus, deux images prises à des résolutions différentes (par deux scanners différents par exemple) ne peuvent avoir les mêmes valeurs de pixels. On peut toujours se donner une marge de tolérance (exemple : si la différence des valeurs de deux pixels est inférieure à 3, on considère que ces pixels sont égaux), mais ces techniques ne sont clairement pas rigoureuses [130].

2.3.2 La comparaison basée sur les minuties

C'est la technique la plus utilisée car elle se base sur des traits individuels propres à chaque personne : *les minuties*. Ces dernières sont extraites de deux empreintes, et représentées comme un ensemble de points dans le plan à deux dimensions selon le modèle de coordonnées (voir figure 2.8). La comparaison consiste à trouver un bon alignement de minuties de deux empreintes (I et T) qui produit un maximum de paires de minuties semblables (voir figure 2.20). La figure 2.20 ci-contre montre que les minuties

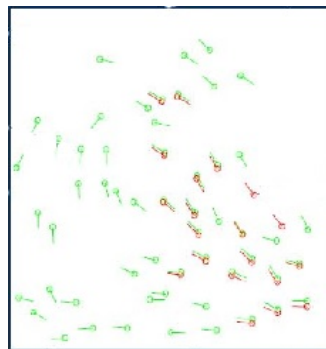


FIGURE 2.20 – La comparaison de minuties de deux empreintes

des deux empreintes à comparer sont représentées en différentes couleurs (le rouge et le vert). Les minuties rouges et vertes côte à côte sont identiques.

La comparaison de minuties (généralement appelée le "*matching de minuties*") consiste à comparer les minuties de deux empreintes en les plaçant dans le plan bidimensionnel et à retrouver les paires de minuties qui ont le même emplacement et la même orientation : on dit que ces minuties ont été "*matchées*" [157]. On peut donc voir les deux empreintes à comparer I et T comme des vecteurs de minuties, où chaque minutie m est représentée à son tour par un vecteur (x, y, θ) où x et y sont les coordonnées de l'emplacement de la minutie dans l'image de l'empreinte, et θ l'angle de la minutie (c'est à dire l'orientation de la minutie, voir figure 2.8). Cette façon de représenter les minuties d'une empreinte digitale est appelée *le modèle de coordonnées* [112].

$$T = \{m_1, m_2, \dots, m_m\} \quad m_i = \{x_i, y_i, \theta_i\} \quad i = 1..m$$

$$I = \{m'_1, m'_2, \dots, m'_n\} \quad m'_j = \{x'_j, y'_j, \theta'_j\} \quad j = 1..n$$

Où m et n sont respectivement le nombre de minuties de T et I .

On considère qu'il y a un "match" entre une minutie m'_j de T et une minutie m_i de I lorsque la distance sd (pour *spatial distance*) qui les sépare est inférieure ou égale à une tolérance r_0 et la différence dd (pour *direction difference*) de leurs angles est inférieure ou égale à une certaine tolérance angulaire θ_0 :

$$sd(m'_j, m_i) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \leq r_0 \quad \text{et} \quad (2.18)$$

$$dd(m'_j, m_i) = \min(|\theta'_j - \theta_i|, 360 - |\theta'_j - \theta_i|) \leq \theta_0 \quad (2.19)$$

La distance entre les deux minuties se calcule en prenant le minimum entre $|\theta'_j - \theta_i|$ et $360 - |\theta'_j - \theta_i|$ parce que les angles sont circulaires. La différence entre 2° et 358° par exemple, est de 4° et pas de 356° .

Comme expliqué au début de cette section, il est extrêmement difficile de recueillir deux empreintes exactement de la même façon. On ne peut par exemple pas affirmer que les deux doigts ont été placés au même endroit et appuyés contre la surface du scanner exactement de la même manière. On se donne alors une certaine marge de tolérance (*tolerance box*) pour compenser ces petites différences ainsi que les erreurs introduites par les algorithmes d'extraction de minuties [52, 112]. Cependant, les deux empreintes sont souvent recueillies de façon telle que la marge de tolérance ne suffit pas pour les comparer de façon optimale. C'est pourquoi pour placer les minuties dans le plan bidimensionnel, on se donne une tolérance $(\Delta x, \Delta y, \theta)$ supplémentaire pour adapter l'alignement de minuties de la première empreinte (généralement T) à celui de la deuxième (généralement I) comme le montre la figure 2.21. C'est-à-dire que l'alignement $(\Delta x, \Delta y, \theta)$ est ajouté à l'emplacement $(\Delta x, \Delta y, \theta_j)$ de chaque minutie m_j de T .

Etant donné que le but est d'avoir un nombre maximum de "matching", la comparaison peut s'effectuer avec plusieurs alignements d'une famille soigneusement choisie :

$$\{\Delta x_1, \Delta x_2, \dots, \Delta x_k, \Delta y_1, \Delta y_2, \dots, \Delta y_k, \theta_1, \theta_2, \dots, \theta_k\}$$

On considère que deux empreintes sont identiques lorsqu'un certain nombre t de minuties identiques est atteint. Ce nombre peut varier d'un système à un autre suivant le degré de sécurité choisi par les développeurs [112]. Une simulation de matching de quelques minuties peut être illustré à la figure 2.21. On constate que lors du matching de deux images d'empreintes digitales, certaines distorsions résiduelles (minuties en vert dans l'image) peuvent apparaître entre les minuties détectées des deux images. Par ailleurs on peut parfois noter la présence de fausses minuties (minuties en rouge dans l'image).

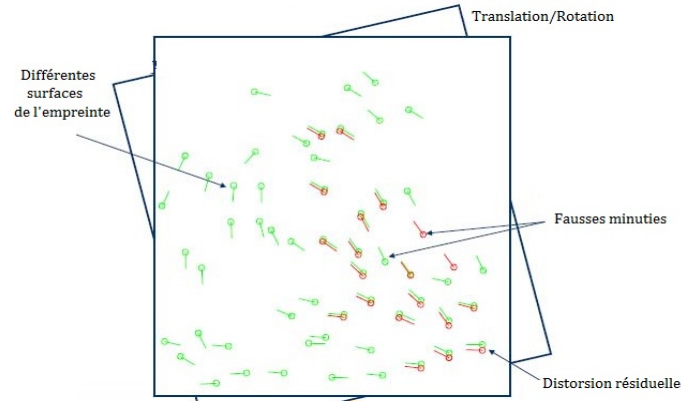


FIGURE 2.21 – Le matching de minuties

Soit une fonction mm (pour *minutiae matcher*) de matching de minuties. $mm(m_j, m_i)$ vaut :

- 1 si $sd(m_j, m_i) \leq r_0$ et $dd(m_j, m_i) \leq \theta$,
- 0 sinon.

Le problème de matching se formule alors comme suit :

$$\max \sum_{i=1}^m mm(m'_{p(i)}, m_i)$$

Où $p(i)$ est une fonction telle que :

- $p(i) = j$ si m'_j est la minutie la plus proche de m_i parmi toutes les minuties $\{m_k \mid k = 1..n, mm(m_k, m_i) = 1\}$. C'est-à-dire que si plusieurs minuties de T satisfont les équations 2.18 et 2.19, on considère la plus proche de m_i .
- $p(i) = null$ si $\{k = 1..n, mm(m_k, m_i) = 0\}$.

Chaque minutie m_i de I est donc comparée à toutes les minuties de T pour trouver la plus semblable à m_i . Ce processus peut prendre un temps considérable pour une base de données contenant des milliers de templates. De nos jours, si l'on peut identifier une empreinte digitale d'une manière assez fiable, la complexité des algorithmes de comparaison d'empreintes laisse à désirer [112, 130]. Plusieurs recherches sont actuellement en cours pour essayer de résoudre ce problème.

2.3.3 La comparaison basée sur les singularités de lignes d'empreintes digitales

Lorsqu'on dispose d'une image de mauvaise qualité, il est très difficile (voire impossible si la qualité est très mauvaise) de réaliser l'extraction de minuties [27, 87]. En

revanche, d'autres singularités de lignes d'empreintes digitales telles que l'orientation locale des lignes, la taille, la position géométrique, la fréquence, la texture locale, les pores de respirations... , peuvent être extraites plus facilement que les minuties.

Ces singularités sont cependant très instables et très variables d'une personne à une autre et d'un doigt à un autre y compris celui d'une même personne. Cette technique consiste donc à comparer un ensemble de traits caractéristiques des lignes d'une empreinte digitale préalablement définie [27, 165]. Elles sont basées sur des modèles mathématiques très complexes, car elles doivent prendre en compte des éléments très diversifiés et parfois elle doivent même jouer un rôle de prédicteur pour le cas des images relevées partiellement dans un lieu public (accessible à des milliers voire des millions de personnes) [112, 130, 165], dans le but de servir d'élément de preuve. Pour toutes les raisons susmentionnées, ces méthodes de comparaison basée sur les singularités de ligne de l'empreinte sont donc très peu utilisées et n'ont pas fait l'objet d'une étude en profondeur dans ce travail.

2.4 *La classification des empreintes digitales*

Comme évoqué dans la section précédente, le processus d'identification d'une empreinte digitale peut être très long. En effet, l'identification d'une empreinte consiste à la comparer à toutes les autres empreintes présentes dans la base de données, laquelle peut en contenir de l'ordre de quelques millions. Il est donc intéressant de classer les empreintes selon certains critères pour diminuer le nombre de comparaisons à effectuer. Cette classification peut s'effectuer selon certains critères classiques comme le sexe, la race, l'âge, etc.

Mais ces informations ne sont pas toujours disponibles ; pour une empreinte relevée sur un lieu de crime par exemple, aucune analyse ne peut indiquer si elle appartient à une personne de tel sexe, tel âge ou telle race [5, 15, 29, 145]. Une manière fiable d'effectuer une classification d'empreintes digitales devrait donc se baser sur des critères propres aux empreintes. Bien que la comparaison d'empreintes digitales se base sur des caractéristiques locales comme les minuties, la classification des empreintes digitales se base sur les caractéristiques globales. La première classification rigoureuse a été introduite en 1892 par Francis Galton [58], qui a défini trois grandes classes d'empreintes digitales : boucles, arches et spires comme illustrées à la figure 2.22 ci-dessous. La classification de Galton a été élargie par Edward Henry (voir [74]) en subdivisant les trois classes de Galton en sous-classes (voir figure 2.22).

Actuellement, on peut rencontrer plusieurs sortes de classifications selon les besoins de l'application (voir dans [49, 112]), mais ce sont toutes des variantes de la classification



FIGURE 2.22 – Les trois classes d'empreintes digitales selon Galton

Galton – Henry. Cette dernière stipule que :

- Une empreinte appartient à la classe *arche* si ses lignes entrent d'un côté et ressortent du côté opposé et si elle ne contient ni boucle ni delta comme régions singulières ;
- Une empreinte de la classe *boucle* possède des lignes qui rentrent d'un côté et ressortent du même côté, ainsi que deux régions singulières (la boucle et le delta), les boucles qui entrent et ressortent du côté gauche sont dites des *boucles à gauche* tandis que celles qui rentrent et ressortent du côté droit sont dites des *boucles à droite* ;
- Une empreinte de la classe *spire* contient au moins une ligne qui fait 360° . Elle peut également contenir deux régions singulières boucles et deux régions deltas.

La figure 2.23 ci-après illustre les cinq classes définies par la classification *Galton – Henry*. Si les boucles jumelles ne sont pas représentées dans ce graphique c'est parce qu'elles sont généralement considérées comme une sous-classe des spires.



FIGURE 2.23 – Les cinq classes les plus utilisées.

Les critères de la classification *Galton–Henry* ont été utilisés pendant longtemps par les experts des empreintes digitales. Les systèmes automatiques ont adopté ces critères afin de réaliser une classification automatique des empreintes digitales. En effet, il suffit de détecter les régions singulières comme vu à la section 2.2.1.3 précédente.

2.5 Conclusion

En somme nous pouvons dire que, la biométrie a fait d'énormes progrès ces dernières décennies et est entrain de s'imposer comme moyen essentiel d'identification ou d'authentification d'individus dans de nombreux domaines, avec comme élément privilégié l'utilisation des empreintes digitales. Bien que la reconnaissance d'empreintes digitales a fait ses preuves dans le domaine de la biométrie, des améliorations doivent encore être apportées dans cette spécialité. En effet, l'inefficacité des systèmes automatiques face aux empreintes de mauvaise qualité, les taux d'erreur relativement acceptables et le temps nécessaire au système pour identifier une empreinte font actuellement l'objet de plusieurs recherches [159, 160]. De nombreuses recherches ont été initiées depuis les années 2010 sur les possibilités d'amélioration des temps de calculs des systèmes de reconnaissance d'empreintes digitales tout en réduisant les taux d'erreur enregistrés [141, 172]. Ces systèmes devraient permettre à terme d'accélérer la comparaison de l'empreinte à identifier avec des millions d'autres empreintes de la base de données préalablement conçue. Les solutions utilisant les réseaux de neurones semblent être très prometteuses grâce notamment aux processus d'apprentissage qu'ils offrent, qui permettent au système de s'adapter aux données à traiter et donc, d'améliorer les taux de reconnaissance. Cependant une contrepartie apparaît évidente dans les temps de calculs qui peuvent être trop longs et les taux d'erreur qui peuvent être assez élevés. La suite de ce mémoire, parle des réseaux de neurones et de leur apport dans le domaine de la reconnaissance d'empreintes digitales en mettant l'accent sur des solutions susceptibles d'améliorer les taux d'erreur.

Introduction aux réseaux de neurones

3.1 *Introduction*

3.1.1 *Historique*

La préhistoire

Les réseaux de neurones sont construits sur un paradigme biologique, celui du neurone formel (au même titre que les algorithmes génétiques le sont sur la sélection naturelle). Ces types de métaphores biologiques sont devenus courants avec les idées de la cybernétique.

Warren Sturgis, McCulloch et *Walter Pitts* ont mené les premiers travaux sur les réseaux de neurones à la suite de leur article fondateur : "*What the frog's eye tells to the frog's brain*" [105]. Ils ont constitué un modèle simplifié de neurone biologique communément appelé *neurone formel* durant les années 1943. Ils ont également montré théoriquement que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes. La fonction des réseaux de neurones formels à l'instar du modèle vivant est de résoudre des problèmes. A l'opposé des méthodes traditionnelles de résolution informatique, on ne doit pas construire un programme pas à pas en fonction de la compréhension de celui-ci. Les paramètres les plus importants de ce modèle sont les *coefficients (ou poids) synaptiques*. Ce sont eux qui construisent le modèle de résolution en fonction des informations données au réseau. Il faut donc trouver un mécanisme qui permette de les calculer à partir des grandeurs que l'on peut acquérir du problème. C'est le principe fondamental de l'*apprentissage*. Dans un modèle de réseaux de neurones formels, "*apprendre*" c'est d'abord calculer les valeurs des poids synaptiques en fonction des exemples disponibles.

Les travaux de *McCulloch* et *Pitts* n'ont pas donné d'indications sur une méthode pour adapter les poids synaptiques.

Cette question au cœur des réflexions sur l'apprentissage a connu un début de réponse grâce aux travaux du physiologiste américain *Donald Hebb* sur l'apprentissage en 1949

décrits dans son ouvrage "*The Organization of Behaviour*" [72]. Hebb a proposé une règle simple qui permet de modifier la valeur des poids synaptiques en fonction de l'activité des unités qu'ils relient. Cette règle aujourd'hui connue sous le nom de *la règle de Hebb* est presque partout présente dans les modèles actuels, même les plus sophistiqués.

Les premiers succès

A partir de cet article de Donald *Hebb* [72], l'idée se propagea au fil du temps dans les esprits, et elle germa dans l'esprit de *Franck Rosenblatt* en 1957 avec le modèle du *perceptron* [136]. C'est le premier système artificiel capable d'apprendre par expérience, y compris lorsque son instructeur commet quelques erreurs (ce en quoi il diffère nettement d'un système d'apprentissage logique formel).

En 1969, *Marvin Lee Minsky* et *Seymour Papert* ont publié un ouvrage [121] mettant en exergue quelques limitations théoriques du *perceptron*, notamment l'impossibilité de traiter des problèmes non linéaires ou de connexité. Ils ont étendu implicitement ces limitations à tous les modèles de réseaux de neurones artificiels. Cependant, les solides qualités de certains réseaux de neurones en matière adaptative, (exemple : *Adaline*), leur permettant de modéliser de façon évolutive des phénomènes eux-mêmes évolutifs, les amèneront à être intégrés sous des formes plus ou moins explicites dans le corpus des systèmes adaptatifs utilisés dans le domaine des télécommunications ou celui du contrôle de processus industriels.

L'ombre et le renouveau

Entre 1967 et 1982 les travaux sur les réseaux de neurones artificiels vont être mis en sommeil. Toutefois la recherche continue sous le couvert de domaines divers par des chercheurs chevronnés tels que : *Anderson*, *Grossberg*, *Kohonen*, etc.

En 1982, *John John Hopfield*, physicien reconnu, donna un nouveau souffle aux techniques neuronales en publiant un article [76] introduisant un nouveau modèle de réseau de neurones (complètement récurrent). Le neuronal redevint un sujet d'étude acceptable, bien que le *modèle de Hopfield* souffrait des principales limitations des modèles des années 1960, notamment l'impossibilité de traiter les problèmes non linéaires. A la même date, les approches algorithmiques de l'intelligence artificielle furent l'objet de désillusion, leurs applications ne répondant pas aux attentes. Cette désillusion motiva une réorientation des recherches en intelligence artificielle vers les réseaux de neurones (bien que ces réseaux concernent la *perception* artificielle plus que l'*intelligence* artificielle à proprement parler).

En 1984, c'est le système de *rétro propagation du gradient de l'erreur* qui est le sujet

le plus débattu dans le domaine [138].

Une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non-linéaires : le *perceptron multicouche* ne possède pas les défauts mis en évidence par *Minsky* [121]. Le Perceptron Multicouche apparaît en 1986, et est introduit par *Rumelhart* [139]. Ces systèmes reposent sur la rétro propagation du gradient de l'erreur dans des systèmes à plusieurs couches.

Les réseaux de neurones ont par la suite connu un essor considérable, et ont fait partie des premiers systèmes à bénéficier de l'éclairage de la théorie de la *régularisation statistique* introduite par *Vladimir Vapnik* [138]. Cette théorie, l'une des plus importantes du domaine des statistiques, permet d'anticiper, d'étudier et de réguler les phénomènes liés au *sur-apprentissage*. On peut ainsi réguler un système d'apprentissage pour qu'il arbitre au mieux entre une modélisation pauvre (exemple : la moyenne) et une modélisation trop riche qui serait optimisée de façon illusoire sur un nombre d'exemples trop petit, et serait inopérant sur des exemples non encore appris, même proches des exemples appris. Le *sur - apprentissage* est une difficulté à laquelle doivent faire face tous les systèmes d'apprentissage par l'exemple, que ceux-ci utilisent des méthodes d'optimisation directe (exemple : régression linéaire), itératives (exemple : descente de gradient), ou itératives semi-directes (gradient conjugué, estimation-maximisation de Baum...) et que ceux-ci soient appliqués aux modèles statistiques classiques, aux modèles de *Markov* ou aux réseaux de neurones formels.

3.1.2 Définition

Un *réseau de neurones* (ou *Artificial Neural Network* en anglais) est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement des vrais neurones (humains ou non). Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type statistique, si bien qu'ils sont placés d'une part dans la famille des applications statistiques, qu'ils enrichissent avec un ensemble de paradigmes permettant de générer de vastes espaces fonctionnels souples et partiellement structurés. Par ailleurs, ils sont placés dans la famille des méthodes de l'intelligence artificielle qu'ils enrichissent en permettant de prendre des décisions s'appuyant davantage sur la perception que sur le raisonnement logique formel [45, 171]. La structure simplifiée d'un réseau de neurones peut être vue à la figure 3.1 ci-après.

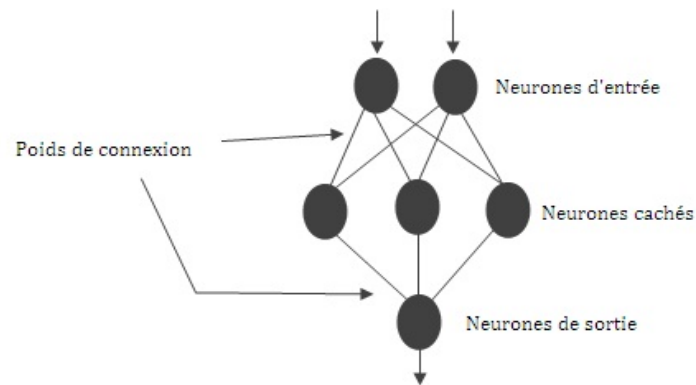


FIGURE 3.1 – Vue simplifiée d'un réseau de neurone

3.1.3 Utilités et limites

Utilités

Grâce à leur capacité de généralisation, les réseaux de neurones sont généralement utilisés dans des problèmes de nature statistique et perceptive, tels que la *classification* ou l'*évaluation*. Par exemple, une banque peut générer un jeu de données sur les clients qui ont effectué un emprunt, constitué de : leur revenu, de leur âge, du nombre d'enfants à charge, et s'il s'agit d'un bon client. Si ce jeu de données est suffisamment grand, il peut être utilisé pour l'entraînement d'un réseau de neurones. Alors, la banque pourra présenter les caractéristiques d'un potentiel nouveau client, et le réseau répondra s'il sera bon client ou non, en généralisant à partir des cas qu'il connaît. Si le réseau de neurones fonctionne avec des nombres réels, la réponse traduit une probabilité de certitude (par exemple : 1 pour "*sûr qu'il sera un bon client*", -1 pour "*sûr qu'il sera mauvais client*", 0 pour "*aucune idée*", 0.9 pour "*presque sûr qu'il sera bon client*").

Il est à noter que le réseau de neurones ne fournit pas de règle exploitable par un humain. Bien que de nombreux travaux soient en train d'être effectués dans ce sens [153], le réseau reste une *boite noire* qui fournit une réponse quand on lui présente une donnée, mais le réseau ne fournit pas de justification facile à lire sur la façon dont celle-ci est obtenue.

Les réseaux de neurones sont réellement utilisés pour :

1. la classification [49, 145] : par exemple pour la classification d'espèces animales par type, étant donnée une analyse *ADN*.
2. la reconnaissance de motifs : par exemple pour la reconnaissance optique de caractères (*OCR*), et notamment par les banques pour vérifier le montant des chèques, par la poste pour trier le courrier en fonction du code postal, etc. . . ;
3. le déplacement automatisé de robots mobiles autonomes ;

4. l'approximation d'une fonction inconnue ;
5. la modélisation accélérée d'une fonction connue mais très complexe à calculer avec exactitude : par exemple certaines fonctions d'inversions utilisées pour décrypter les signaux de télédétection émis par les satellites et les transformer en données sur la surface de la mer ;
6. l'estimation boursière :
 - apprentissage de la valeur d'une entreprise en fonction des indices disponibles : bénéfices, endettements à long et court terme, chiffre d'affaires, carnet de commandes, indications techniques de conjoncture,
 - tentatives de prédiction sur la périodicité des cours boursiers. Ce type de prédiction est très contesté pour deux raisons, l'une étant que le cours d'une action n'a jamais été périodique (le marché anticipe en effet largement les hausses comme les baisses prévisibles) et, l'autre que c'est l'avenir prévisible d'une entreprise qui détermine essentiellement le cours de son action et en aucun cas son passé ;
7. le traitement d'image : compression d'images, reconnaissance de caractères et de signatures, reconnaissance de formes et de motifs, cryptage, classification, ... [19, 42, 83, 94] ;
8. la modélisation de l'apprentissage et amélioration des techniques de l'enseignement ;
9. la fouille de données ;
10. traitement du signal : traitement de la parole, identification de sources, filtrage, classification, ...
11. le contrôle : diagnostic de pannes, commande de processus, contrôle qualité, robotique [117], ...

Limites

1. Les réseaux de neurones artificiels ont besoin de cas réels servant d'exemples pour leur apprentissage (on appelle cela *la base d'apprentissage*). Ces cas doivent être d'autant plus nombreux que le problème est complexe et que sa topologie est peu structurée. Par exemple, on peut optimiser un système neuronal de lecture de caractères en utilisant le découpage manuel d'un grand nombre de mots écrits à la main par de nombreuses personnes. Chaque caractère peut alors être présenté sous la forme d'une image brute, disposant d'une topologie spatiale à deux dimensions, ou d'une suite de segments presque tous liés. La topologie retenue, la complexité du phénomène modélisé et le nombre d'exemples doivent être en rapport. Sur un

- plan pratique, cela n'est pas toujours facile car les exemples peuvent être soit en quantité absolument limitée ou trop onéreux à collecter en nombre suffisant.
2. Impossibilité de réaliser la fonction du *XOR* pour le cas du perceptron simple [83].
 3. Les réseaux de neurones ne fournissent pas les explications concernant leurs résultats ce qui limite donc l'analyse des phénomènes existants. Cette limite est due à l'opacité des réseaux de neurones qui empêche une analyse pertinente des solutions obtenues. Les réseaux de neurones peuvent être assimilés à une boîte noire qui donne une réponse quand on lui fournit des données mais qui ne délivre pas toujours de justification simple à analyser.
 4. Le choix du nombre de couches : qui est en fait un *problème NP complet* [153, 171], car il n'existe pas a priori de méthode standard pour effectuer le choix du nombre de couches du réseau de neurones. Ce choix est effectué de façon subjective et varie en fonction des problèmes qu'on doit résoudre.
 5. Les RNA restent limités quant au critère d'interprétabilité [4]. Bien que certains travaux intéressants ont été effectués à ce sujet [153], il est difficile pour un utilisateur quelconque d'expliquer comment le résultat en sortie a été calculé.

3.1.4 Structure d'un réseau de neurones artificiels

Chez l'humain le siège de l'intelligence et de l'apprentissage est le cerveau. On peut s'en inspirer pour élaborer des algorithmes d'apprentissage à partir des réseaux de neurones artificiels (modèles connexionnistes) [138]. La représentation d'un neurone biologique est donnée à la figure 3.2.

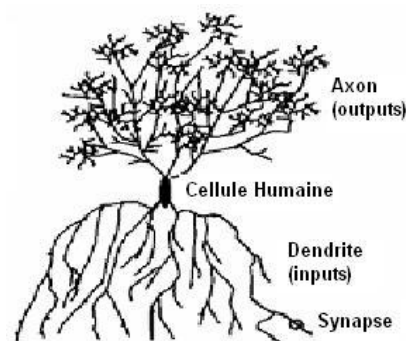


FIGURE 3.2 – Vue simplifiée d'un neurone biologique

Chaque neurone naturel est connecté à quelques milliers d'autres et leur transmet une information en envoyant des *ondes de dépolarisation* (grosso modo, des *pics électriques*). Plus précisément, le neurone reçoit en entrée les signaux provenant des autres,

par des synapses et émet en sortie une information par son axone. De façon grossièrement similaire, les neurones artificiels sont connectés entre eux par des liaisons pondérées et unidirectionnelles ; un réseau de neurones peut donc se représenter par un graphe orienté dont les nœuds sont les neurones artificiels, comme le montre la figure 3.3.

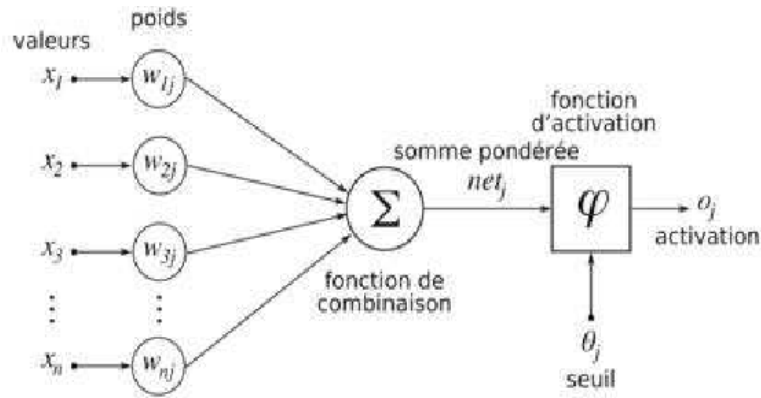


FIGURE 3.3 – Structure d'un neurone formel

Les x_i représentent les valeurs prises en entrée. Par exemple si on effectue la compression d'images [113], les x_i correspondront aux valeurs des niveaux de gris de l'image à compresser. Les w_{ij} représentent les poids de connexions (les liens directs) entre la couche d'entrée et la couche de sortie. Le neurone i utilise une *fonction de combinaison* pour calculer la somme de ses entrées, puis cette valeur passe à travers une *fonction d'activation* pour produire sa sortie o_j , comme le montre l'équation (3.1)

$$o_j = \varphi \left(\sum_{i=1}^N w_{ji} x_i \right) \quad (3.1)$$

<i>Ordinateur de Von Neumann</i>	<i>Cerveau humain</i>
Calcul et mémoire séparés et centralisés	Calcul et mémoire intégrés et distribués
Programme = séquence d'instructions	Calcul = satisfaction de multiples contraintes
Exécution d'un sous-programme à la fois	Combinaison simultanée de multiples sources d'informations
Un seul processeur très rapide	Des centaines de milliards d'unités de calcul très lentes

TABLE 3.1 – Tableau de comparaison ordinateur de Von Neumann et cerveau

Il y a des milliards de neurones dans un cerveau humain : environ 2×10^{12} , pour être

précis. La *période réfractaire* (durée qui succède immédiatement à l'activité d'un nerf ou d'un muscle) de ces neurones est d'environ $10ms$, soit un fonctionnement à la vitesse maximale de $100Hz$ (en fait, il est établi que les neurones traitent simultanément des signaux à diverses fréquences, l'ensemble étant désembrouillé par d'autres neurones en aval) [138]. La puissance d'un cerveau humain est donc "au banc", si l'on peut dire, de 2×10^{14} opérations logiques par seconde. Un processeur de type *PentiumIV*, *AMD64* ou *PowerPC970*, en 2004 travaille à une fréquence de $3GHz$ sur des mots de 32 (*Pentium*) ou 64 (*AMD64* ou *PowerPC*) bits, ce qui peut donner un ordre de grandeur correspond à une puissance brute de 2×10^{11} opérations logiques par seconde dans le cas du *powerPC*.

En dépit de ce différentiel de puissance (un rapport de 1 à 1000 est celui que l'on observe entre les vitesses d'un piéton et d'un chasseur supersonique), il est tentant de simuler le fonctionnement de neurones pour résoudre quelques problèmes simples. Une raison de se réjouir de cette simplicité est que pour finir d'éduquer correctement un cerveau de 2×10^{12} neurones, il ne faut tout de même pas moins de 25 ans, temps dont il est difficile de disposer en laboratoire [138].

Un réseau de neurones (on parle parfois aussi de *réseau neuromimétique*) est constitué d'un très grand nombre de petites unités de traitement identiques appelées *neurones artificiels*. Elles étaient électroniques dans les premières implémentations (perceptrons de *Rosenblatt*) [136]; on les simule le plus souvent sur ordinateur aujourd'hui pour des questions de coût et de commodité. La taille et la vitesse des réseaux leur permettent de traiter très correctement des questions de *perception* ou de *classification automatique* (et *approximative*).

La structure générale d'un neurone formel inclut des fonctions de combinaison et des fonctions d'activation.

Fonction de combinaison

Considérons un neurone quelconque, il reçoit des autres neurones en amont un certain nombre de valeurs via ses connexions synaptiques et produit une certaine valeur en utilisant une *fonction de combinaison*. Cette fonction peut donc être formalisée comme étant une fonction vecteur à scalaires, notamment :

- (1) Les réseaux de type *MLP* (Multi-Layer Perceptron) calculent une combinaison linéaire des entrées, c'est-à-dire que la fonction de combinaison renvoie le produit scalaire entre le vecteur des entrées et le vecteur des poids synaptiques.
- (2) Les réseaux de type *RBF* (Radial Basis Function) calculent la distance entre les entrées, c'est-à-dire que la fonction de combinaison renvoie la norme euclidienne du vecteur issu de la différence vectorielle entre les vecteurs d'entrée.

Fonction d'activation

Un réseau de neurones est constitué de *cellules* (ou *neurones*), connectées entre elles par des liaisons affectées de poids. Ces liaisons permettent à chaque cellule de disposer d'un canal pour "envoyer" et "recevoir" des signaux en provenance d'autres cellules du réseau. Chacune de ces connexions reçoit un poids (une pondération), qui détermine son impact sur les cellules qui sont directement connectées à elle. Chaque cellule dispose ainsi d'une entrée qui lui permet de recevoir de l'information d'autres cellules, mais aussi de ce que l'on appelle une *fonction d'activation*, qui est dans les cas les plus simples, une identité du résultat obtenu par l'entrée. Ainsi, pour un réseau de neurones avec N cellules en entrée, notées C_1, \dots, C_N , et N poids affectés aux liaisons et notés W_1, \dots, W_N l'entrée d'une cellule de la seconde couche sera généralement une somme pondérée des valeurs de sortie des neurones précédents :

$$X = W_1 \times C_1 + W_2 \times C_2 + W_3 \times C_3 + \dots + W_N \times C_N \quad (3.2)$$

Pour obtenir la valeur de sortie Y du neurone concerné, une fonction d'activation identité peut être utilisée. Elle est de type :

$$Y = id \times X \quad (3.3)$$

Mais le choix d'une fonction d'activation se révèle être un élément constitutif important des réseaux de neurones. Ainsi, l'identité n'est pas toujours suffisante, bien au contraire ; et le plus souvent des fonctions non linéaires et plus évoluées seront nécessaires. A titre illustratif voici quelques fonctions couramment utilisées comme fonction d'activation :

– la fonction logitistique

$$Y = \frac{2}{\left(1 + e^{(-2 \times X)} - 1\right)}$$

– la tangente hyperbolique

$$Y = \frac{2}{\left(1 + e^{(-2 \times X)} - 1\right)}$$

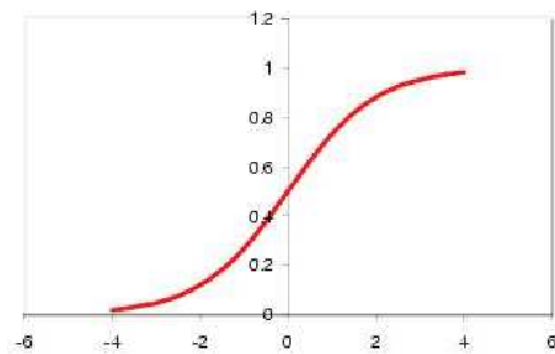
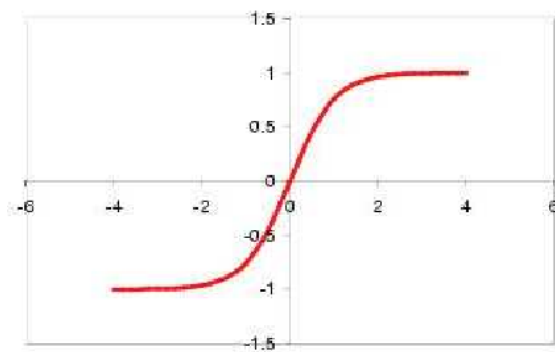
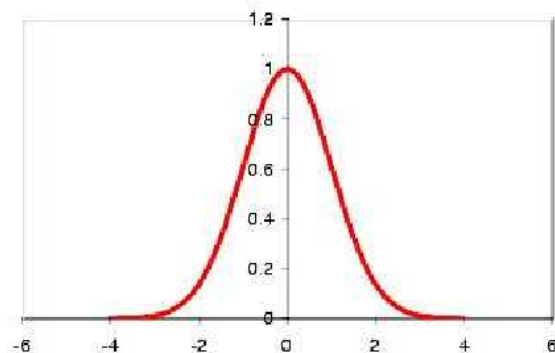
– la fonction de Gauss

$$Y = e^{\left(\frac{-X^2}{2}\right)}$$

– la fonction à seuil

$$Y = 0 \text{ si } X \leq 0 \text{ et } Y = 1 \text{ si } X > 0$$

Illustration graphique, voir figures 3.4, 3.5, 3.6, 3.7.

FIGURE 3.4 – La fonction logistique : $Y = \frac{2}{(1+e^{(-2 \times X)-1})}$ FIGURE 3.5 – La tangente hyperbolique : $Y = \frac{2}{(1+e^{(-2 \times X)-1})}$ FIGURE 3.6 – La fonction de Gauss : $Y = e^{(-\frac{x^2}{2})}$

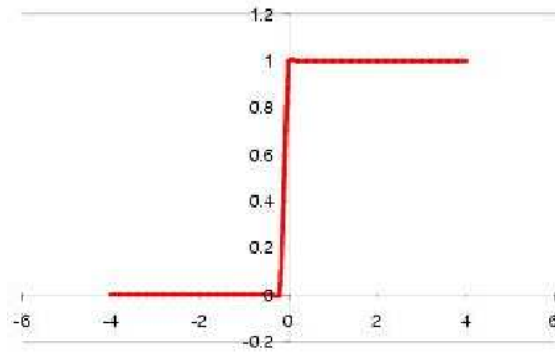


FIGURE 3.7 – Une fonction à seuil : $Y = 0$ si $X \leq 0$ et $Y = 1$ si $X > 0$

3.2 Les étapes de la conception d'un réseau de neurones

Pour construire un réseau de neurones pour l'apprentissage, la première chose à faire n'est pas de choisir le type de réseau mais de bien choisir ses échantillons de données d'apprentissage, de tests et validation. Ce n'est qu'ensuite que le choix du type de réseau interviendra. Afin de clarifier un peu les idées, voici chronologiquement les quatre grandes étapes qui doivent guider la création d'un réseau de neurones.

3.2.1 Choix et préparation des échantillons

Le processus d'élaboration d'un réseau de neurones commence toujours par le choix et la préparation des échantillons de données. Comme dans les cas des analyses de données, cette étape est cruciale et va aider le concepteur à déterminer le type de réseau le plus approprié pour résoudre son problème. La façon dont se présente l'échantillon conditionne le type de réseau, le nombre de cellules d'entrée, le nombre de cellules de sortie et la façon dont il faudra mener l'apprentissage, les tests et la validation.

3.2.2 Elaboration de la structure du réseau

La structure du réseau dépend étroitement du type des échantillons. Il faut d'abord choisir le type de réseau : un perceptron standard, un réseau de *Hopfield* [76], un réseau à décalage temporel (*TDNN*) [115, 148], un réseau de *Kohonen* [22, 88], un *ARTMAP*, etc... Dans le cas du perceptron par exemple, il faudra aussi choisir le nombre de neurones dans la couche cachée. Plusieurs méthodes existent et on peut par exemple prendre une moyenne du nombre de neurones d'entrée et de sortie, mais rien ne vaut de tester toutes les possibilités et de choisir celle qui offre les meilleurs résultats.

Propagation en avant

Dans ce cas, la structure du réseau de neurones peut être celle d'un réseau multicouche ou celle d'un réseau à connexion à locale [43]. Mais le principe reste celui de la propagation des activations de l'entrée vers la sortie comme le montre la figure 3.8 ci-après.

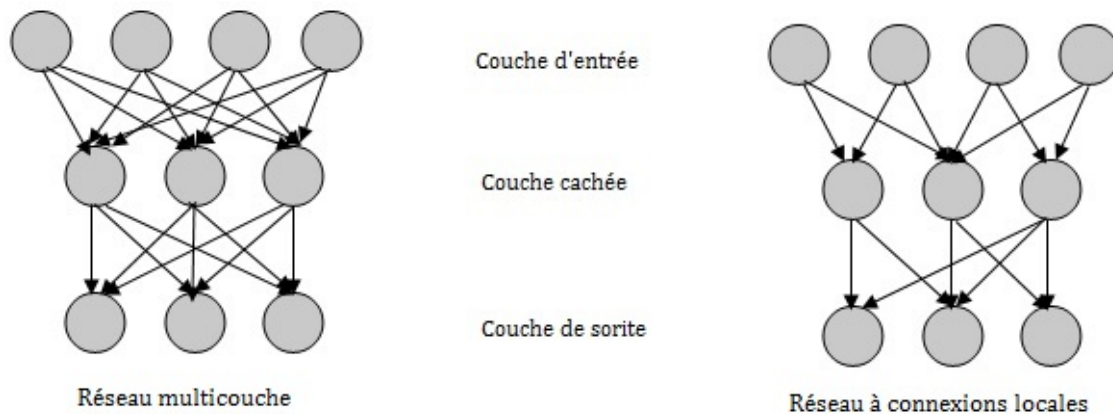


FIGURE 3.8 – Structure d'un réseau à propagation avant classique

Modèle récurrent

Pour le cas des réseaux de neurones qui utilisent une structure d'interconnexion à modèle récurrent, la propagation des activations se fait de deux façons :

- synchrone : c'est à dire que toutes les unités sont mises à jour simultanément ;
- asynchrone : les unités sont mises à jour séquentiellement

La figure 3.9 ci-dessus illustre le mode synchrone et asynchrone d'un modèle récurrent de réseau de neurones.

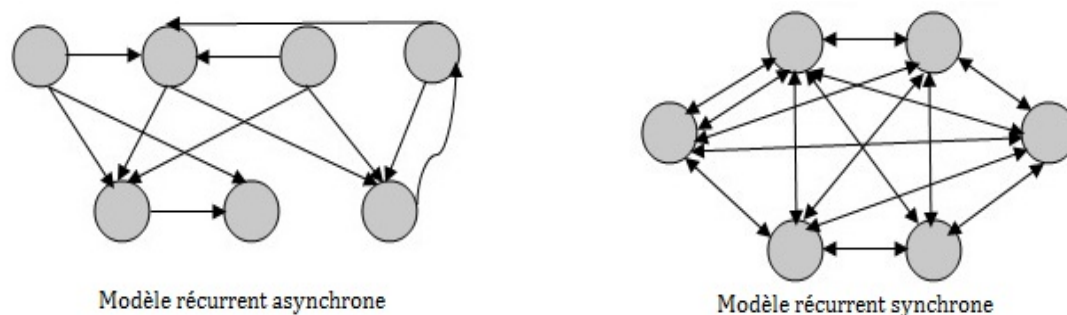


FIGURE 3.9 – Structure d'un réseau à propagation avant

3.2.3 Apprentissage

La notion d'*apprentissage*, bien que connue déjà depuis *Sumer* [105], n'est pas modélisable dans le cadre de la logique déductive. Celle-ci en effet procède à partir de connaissances déjà établies dont on tire des connaissances dérivées. Or il s'agit ici de la démarche inverse par observations limitées, tirées des généralisations plausibles. La notion d'apprentissage recouvre deux réalités successives :

1. *mémorisation*

le fait d'assimiler sous une forme dense des exemples éventuellement nombreux

2. *généralisation*

le fait d'être capable grâce aux exemples appris, de traiter des exemples distincts encore non rencontrés mais similaires.

Ces deux points sont partiellement en opposition. Si on privilégie l'un, on élaborera un système qui ne traitera pas forcément de façon très efficace l'autre. Dans le cas des systèmes d'apprentissage statistique, utilisés pour optimiser les modèles statistiques classiques, réseaux de neurones et automates de *Markov* [2, 59, 60], c'est la généralisation qui est l'objet de toute l'attention.

On distingue deux types d'apprentissage :

- *Apprentissage supervisé.*

Un apprentissage est dit *supervisé* lorsque l'on force le réseau à converger vers un état final précis (connu à l'avance), en même temps qu'on lui présente un motif.

- *Apprentissage non supervisé.*

A l'inverse, lors d'un apprentissage *non supervisé*, le réseau est laissé libre de converger vers n'importe quel état final lorsqu'on lui présente un motif.

Les distinctions observées ci-dessus amènent à la définition de certains algorithmes d'entraînement.

3.5.3.1 Algorithmes

La large majorité des réseaux de neurones possède un algorithme "*d'entraînement*" qui consiste à modifier les poids synaptiques en fonction d'un jeu de données présentées en entrée du réseau. Le but de cet entraînement est de permettre au réseau de neurones "*d'apprendre*" à partir des exemples. Si l'entraînement est correctement réalisé, le réseau est capable de fournir des réponses en sortie très proches des valeurs d'origines du jeu de données d'entraînement. Mais tout l'intérêt des réseaux de neurones réside dans leur capacité à généraliser à partir du jeu de test.

On distingue entre autres les algorithmes d'apprentissage suivants :

– **Sur apprentissage :**

Il arrive qu'à force de faire apprendre un réseau de neurones toujours sur le même échantillon, celui-ci devienne inapte à reconnaître autre chose que les éléments présents dans l'échantillon. On parle alors de "*sur apprentissage*" : le réseau est devenu trop spécialisé et ne généralise plus correctement.

– **L'élagage :**

(ou *pruning*, en anglais) est une méthode qui permet d'éviter le sur apprentissage tout en limitant la complexité du modèle. Elle consiste à supprimer des connexions (ou synapses), des entrées ou des neurones du réseau une fois l'apprentissage terminé. En pratique, les éléments qui ont la plus petite influence sur l'erreur de sortie du réseau sont supprimés. Les deux algorithmes d'élagage les plus utilisés sont :

a) *Optimal Brain Damage (OBD)* de Y. Le Cun et al. [100]

b) *Optimal Brain Surgeon (OBS)* de B. Hassibi et D.G. Stork. [30]

– **La rétro propagation :**

Elle consiste à rétro propager l'erreur commise par un neurone à ses synapses et aux neurones qui y sont reliés. Pour les réseaux de neurones, on utilise habituellement la *rétro propagation du gradient de l'erreur*, qui consiste à corriger les erreurs selon l'importance des éléments qui ont justement participé à la réalisation de ces erreurs : les poids synaptiques qui contribuent à engendrer une erreur importante se verront modifiés de manière plus significative que les poids qui ont engendré une erreur marginale [101, 83, 92].

3.5.3.2 Principe de l'algorithme de rétro propagation

(1) Initialiser les poids de connexions par des valeurs aléatoires (généralement petites, dans l'ordre de 0 et 1) ;

(2) Choisir un pattern et l'appliquer à la couche d'entrée ;

(3) Propager le signal à travers le réseau :

$$v_i^m = g(h_i^m) = g\left(\sum_j w_{ij}^m \times v_j^{m-1}\right)$$

jusqu'au calcul de v^N ;

(4) Calculer les deltas des sorties :

$$\delta_i^N = g'(h_i^N) \times (y_i^m - v_i^m);$$

(5) Calculer les deltas sur les autres couches :

$$\delta_i^{m-1} = g'(h_i^{m-1}) \sum_j w_{ji}^m \delta_j^m;$$

$$m = N, N - 1, \dots, 2$$

(6) Modifier les poids de connexions en utilisant :

$$\Delta w_{ij}^m = \eta \delta_i^m v_j^{m-1};$$

$$w_{ij}^m = w_{ij}^{m-1} + \Delta w_{ij}^m;$$

$$\eta = \frac{1}{\eta};$$

(7) Aller à l'étape 2 si encore des patterns ;

(8) Itération suivante.

3.2.4 Validation et tests

Alors que les tests concernent la vérification des performances d'un réseau de neurones hors échantillon et sa capacité de généralisation, la validation est parfois utilisée lors de l'apprentissage. Une fois le réseau calculé, il faut toujours procéder à des tests afin de vérifier que le réseau réagit correctement. Pour les tests dans le cas général, une partie des données est simplement écartée de l'échantillon d'apprentissage et conservée pour les tests hors échantillon. On peut par exemple utiliser 60% de l'échantillon pour l'apprentissage, 20% pour la validation et 20% pour les tests. Dans les cas de petits échantillons, on ne peut pas toujours utiliser une telle distinction, simplement parce qu'il n'est pas toujours possible d'avoir suffisamment de données dans chacun des groupes ainsi créés. On a alors parfois recours à des procédures comme la cross-validation pour établir la structure optimale du réseau.

Il existe plusieurs types de réseaux de neurones à ce jour, mais la mise sur pied de leurs différentes structures respecte toute l'ordre des étapes définis ci-dessus.

3.3 Les différents types de réseaux de neurones

L'ensemble des poids des liaisons synaptiques détermine le fonctionnement du réseau de neurones. Les motifs sont présentés à un sous-ensemble du réseau de neurones : la couche d'entrée. Lorsqu'on applique un motif à un réseau, celui-ci cherche à atteindre un état stable. Lorsqu'il est atteint, les valeurs d'activation des neurones de sortie constituent le résultat. Les neurones qui ne font ni partie de la couche d'entrée ni de la couche de sortie sont dits *neurones cachés*.

Les types de réseaux de neurones diffèrent par plusieurs paramètres :

- la topologie des connexions entre les neurones,
- la fonction de combinaison utilisée (somme pondérée, distance pseudo euclidienne, . . .),
- la fonction d’activation utilisée (sigmoïde, fonction linéaire, fonction de Gauss, . . .),
- l’algorithme d’apprentissage (rétro propagation du gradient de l’erreur, . . .).

De nombreux autres paramètres sont susceptibles d’être mis en œuvre dans le cadre de l’apprentissage, par exemple : la méthode de dégradation des pondérations (weight decay), permettant d’éviter les effets de bord et de neutraliser le sur apprentissage.

On distingue donc selon le cas les réseaux de neurones à apprentissage supervisés d’une part, et les réseaux de neurones à apprentissage non supervisés d’autre part.

3.3.1 Réseau de neurones à apprentissage supervisés

Cette section présente certaines familles de réseaux parmi les plus populaires, toujours mieux adaptés à la recherche de solutions de problèmes particuliers :

1. Le perceptron multicouche

Il est apparu en 1986 et introduit par *Rumelhart* [140] ; c’est un réseau à apprentissage supervisé. Le perceptron multicouche [57] est sans doute le plus simple et le plus connu des réseaux de neurones. La structure est relativement simple : une couche d’entrée, une couche de sortie et une ou plusieurs couches cachées. Chaque neurone de la couche cachée est relié à tous les neurones des autres couches, comme le montre la figure 3.10.

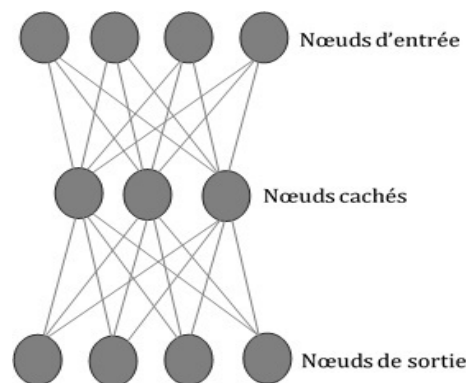


FIGURE 3.10 – Perceptron à trois couches

– Les nœuds d’entrée

La première couche est appelée couche d’entrée. Elle recevra les données source que l’on veut utiliser pour l’analyse. Sa taille est donc directement déterminée par le nombre de variables en entrée du réseau.

– *Les nœuds cachés*

La seconde couche est une couche cachée, en ce sens qu'elle n'a qu'une utilité intrinsèque pour le réseau de neurones et n'a pas de contact direct avec l'extérieur. Les fonctions d'activation sont en général non linéaires sur cette couche mais il n'y a pas de règle à respecter. Le choix de sa taille n'est pas implicite et doit être ajusté. Il est souvent préférable pour obtenir de bons résultats, d'essayer le plus de tailles possibles. Ces nœuds cachés peuvent être repartis sur plus d'une couche, la couche cachée aura alors plus d'une couche dans l'ensemble du réseau de neurones considéré.

– *Les nœuds de sortie*

La dernière couche est appelée couche de sortie. Elle donne le résultat obtenu après traitement par le réseau des données entrées dans la première couche. Sa taille est directement déterminée par le nombre de variables qu'on veut en sortie.

2. Adaline (ADaptive LInear NEuron)

Dans le cas d'Adaline qui est un réseau à apprentissage supervisé sans rétro propagation, on effectue l'apprentissage en utilisant les sorties des neurones avant un passage à travers la fonction d'activation [151]. De ce fait, on utilise uniquement la somme pondérée des entrées avec les poids.

3. Machine de Cauchy

- *Adaptive Heuristic Critic (AHC)*[151]
- *Time Delay Neural Network (TDNN)*[115, 148]
- *Associative Reward Penalty (ARP)*[151]
- *Avalanche Matched Filter (AMF)*[151]
- *Backpercolation (Perc)*[151]
- *Adaptive Logic Network (ALN)*[151]
- *Cascade Correlation (CasCor)*[83]
- *Probabilistic Neural Network (PNN)* [35]

4. Le réseau de Hopfield

Ces réseaux sont des réseaux récursifs, un peu plus complexes que les perceptrons multicouches. Chaque cellule est connectée à toutes les autres et les changements de valeurs de cellules s'enchaînent en cascade jusqu'à un état stable. Ces réseaux sont bien adaptés à la reconnaissance de formes [22].

3.3.2 Réseau de neurones à apprentissage non supervisé

Dans ce type d'apprentissage non supervisé, les neurones sont en compétition pour être actifs. Ils sont à sortie binaire et on dit qu'ils sont actifs lorsque leur sortie vaut

1. Alors que dans les autres règles plusieurs sorties de neurones peuvent être actives simultanément, dans le cas de l'*apprentissage compétitif*, un seul neurone est actif à un instant donné. Chaque neurone de sortie est spécialisé pour "détecter" une suite de formes similaires et devient alors un détecteur de caractéristiques. La fonction d'entrée est dans ce cas,

$$h = b - d(W, X) \quad (3.4)$$

où b , W et X sont respectivement les vecteurs seuils, poids synaptiques et entrées. Et d est une mesure de distance.

Les réseaux de neurones à apprentissage non supervisé sont nombreux et diversifiés. Il n'est pas possible d'énumérer la liste complète à ce jour. Les plus connus sont les suivants :

1. Le réseau de Kohonen

Les réseaux de *Kohonen* [89] dont on parle généralement sont des réseaux à apprentissage non supervisé avec rétro propagation. Ils décrivent en fait trois familles de réseaux de neurones :

- *Quantification vectorielle (QV)*

La quantification vectorielle (ou vector quantization) a été introduite par *Grossberg*, en 1976 [68]. Elle utilise l'apprentissage non supervisé et permet de retrouver des groupes sur un ensemble de données.

- *Self organization map (SOM)*

Les *SOM* sont des réseaux à apprentissage non supervisé avec rétro propagation, issus des travaux de *Kohonen* (1991) [91]. Ces réseaux sont très utilisés pour l'analyse de données. Ils permettent de cartographier en deux dimensions et de distinguer des groupes dans les ensembles de données.

- *Learning Vector Quantization (LVQ)*

Les réseaux utilisant la méthode *LVQ* sont des réseaux à apprentissage supervisé qui ont été proposés par *Kohonen* (1986) [90]. La *LVQ* est une méthode qui est bien adaptée à la classification de données par "*recherche du plus proche voisin*" [77].

2. Discrete Hopfield (DH)[76]

3. Continuous Hopfield (CH) [76]

4. Apprentissage compétitif [100]

5. etc...

Les différents réseaux de neurones présentés ci-dessus varient suivant le modèle d'apprentissage, et produisent des résultats tout au même équivalent. Dans les sections qui

suivent, sont exposés les principaux apports des techniques neuronales dans le traitement d'images.

3.4 Les réseaux de neurones pour les pré-traitements

Le traitement du signal est une des applications majeures des réseaux de neurones. En particulier, plusieurs travaux ont étudié leur utilisation dans le cadre de la reconnaissance de motifs [24, 133]. Pour les transformations linéaires, ils apportent l'avantage du parallélisme [100]. Leurs algorithmes d'apprentissage permettent de définir des transformations non linéaires et des prédicteurs scalaires ou vectoriels non linéaires. Enfin, en tant que classificateurs, certaines architectures peuvent être utilisées pour la segmentation [83].

3.4.1 Transformation

Plusieurs auteurs ont montré l'efficacité des *réseaux de neurones cellulaires (CNN)* pour réaliser sur des images des traitements n'impliquant qu'un nombre réduit de pixels voisins [47, 118, 122]. Cette technique fut proposée par *Chua et Yang* [37]. Un *neurone cellulaire* est une unité de calcul dont l'état à un instant donné dépend des sorties précédentes des cellules de son voisinage et de leurs entrées. Les paramètres d'un neurone cellulaire sont les poids appliqués à ces valeurs. Les neurones cellulaires sont repartis sur un maillage (généralement maillage carré $2D$) et ils ont tous les mêmes poids. La relation entre état et entrées s'établit sous la forme d'une équation différentielle :

$$\frac{de_i}{dt} = -\frac{1}{\tau}e_i + \sum_{V_j} a_j i_j + \sum_{V_j} b_j s_j + e_0 \quad (3.5)$$

où e_i désigne l'état du neurone i , i_j et s_j désignent l'entrée et la sortie du neurone j et V_i désigne le voisinage du neurone i . Les paramètres a_j et b_j sont les poids (matrices A et B) et e_0 est une constante. La sortie d'un neurone est déterminée par une fonction croissante de l'état. Le comportement d'un réseau est donc dynamique [105]. Bien que l'équation (3.5) d'état n'implique que des calculs locaux, un neurone cellulaire peut être influencé par propagation par n'importe quel autre élément du réseau [105]. L'article de *Dony et Haykin* [47] résume bien l'intérêt des réseaux de neurones plus classiques pour les transformations linéaires et en particulier la *KLT* avec réduction de dimension : seules les l premières composantes sont gardées après transformation des vecteurs k -dimensionnels. Les neurones sont linéaires, donc une seule couche est suffisante. Les l neurones de sortie ont k entrées et calculent chacune une projection.

La règle d'apprentissage d'*Oja* [123] permet de déterminer la première composante principale. Un seul neurone calcule le produit scalaire

$$y = W.x = \sum_{j=1}^k w_j.x_j \quad (3.6)$$

L'algorithme d'apprentissage utilise itérativement la règle suivante :

$$w_j(n+1) = w_j(n) + \eta[y(n) + x_j(n) - w_j(n).y^2(n)] \quad (3.7)$$

Où η est le coefficient d'apprentissage.

Des généralisations de cet algorithme (l'algorithme de *Sanger* [142] par exemple) permettent, en utilisant un réseau composé de l unités de traitement linéaires, d'obtenir les l premiers vecteurs propres de la matrice de covariance [83]. L'avantage des réseaux de neurones est une moins grande complexité de l'algorithme de recherche des composantes principales et une possible adaptabilité à des variations des statistiques des images [83].

3.4.2 Prédiction

La définition d'un modèle prédictif $\tilde{y} = f(x)$ est un problème d'approximation de fonction [118, 132]. Une des applications des réseaux de neurones, en particulier les perceptrons multicouches (*MLP*) et l'apprentissage par rétro propagation permettent de construire des modèles prédictifs non linéaires [75, 161]. Cette approche est décrite par *Rizvi* et *Nasrabadi* dans [131] pour la *QV* prédictive d'images. Selon les autres auteurs l'utilisation de la non linéarité permet de mieux prédire les contours [18]. Par conséquent, la quantification vectorielle qui suit la prédiction est plus efficace. Cette technique a également été étudiée dans le cas scalaire [133], et rapporte que le modèle prédictif non linéaire obtenu est plus performant (réduction de l'entropie en sortie) que le modèle prédictif linéaire optimal.

3.4.3 Segmentation

La classification est également une application classique des réseaux de neurones. *Balsi* et *David* [12] ont étudié l'utilisation des réseaux de neurones cellulaires pour la séparation des pixels d'une image selon qu'ils appartiennent à une zone texturée ou non, ici les réseaux de neurones cellulaires sont utilisés pour le parallélisme des calculs. Dans [18], le réseau de neurones est utilisé pour la détection automatique de régions d'intérêts dans des images médicales. L'architecture utilisée est celle des réseaux de neurones probabilistes. Ici, c'est la faculté d'apprentissage des réseaux de neurones et la possibilité d'ajouter de nouvelles connaissances qui sont mises en avant.

3.5 Les réseaux de neurones pour la quantification

Définition

Un *réseau de neurones compétitif* est un réseau de neurones qui possède une couche d'unités complètement connectées aux entrées. Chaque neurone calcule la distance entre son vecteur d'entrée et son vecteur poids, et produit en sortie le neurone "gagnant", c'est à dire celui dont la distance est minimale.

Lorsque la quantification n'est pas uniforme (c'est à dire que les blocs à quantifiés ne sont pas les mêmes), cette opération peut être coûteuse en calculs, les réseaux de neurones cellulaires et les réseaux de neurones compétitifs (telque le réseau de *Kohonen* [88]) peuvent être utilisés pour leur parallélisme.

3.5.1 Codage en quantification scalaire et vectorielle

La quantification scalaire uniforme d'une valeur x ne demande qu'une seule opération [92]. Cependant, si les valeurs codes (valeurs à quantifiées) ne sont pas réparties uniformément, il faut comparer chacune d'elles à x .

Afin de paralléliser ces calculs, *Moreira* et *Pineda* [118] proposent d'utiliser les réseaux de neurones cellulaires, N valeurs codes correspondant à $N - 1$ seuils de décision. La quantification consiste à appliquer successivement $N - 1$ seuillages des valeurs de pixels. Avec un neurone par pixel, l'ensemble de l'image est traitée en parallèle.

La quantification vectorielle (QV) est l'opération réalisée par un réseau de neurones compétitifs ; une telle architecture montre comment implémenter de manière parallèle le codage en quantification vectorielle [97, 131, 133].

D'autre part, la majorité des techniques de QV utilisent une répartition non uniforme des vecteurs codes, le choix de ces vecteurs doit minimiser l'erreur de reconstruction. Les algorithmes d'apprentissage neuronal compétitif apportent une solution à la construction du dictionnaire [83].

Le codage d'un vecteur x de dimension k consiste à lui associer un entier i compris entre 1 et N . La distorsion introduite est généralement mesurée par l'*erreur quadratique moyenne* :

$$d(x, y) = \sum_{j=1}^k (x_j - y_{ij})^2. \quad (3.8)$$

Afin de minimiser l'erreur quadratique moyenne, le codage suit la *règle du plus proche voisin* : on choisi i tel que :

$$\forall j, d(x, y_i) \leq d(x, y_j). \quad (3.9)$$

Donc le codage de chaque vecteur demande le calcul de N distorsions vectorielles afin d'en identifier le minimum. Les calculs de distance peuvent se faire en parallèle pour un codage rapide [83].

3.5.2 Construction du dictionnaire pour la QV

Algorithmes

Il existe différents algorithmes d'apprentissage pour déterminer la position des vecteurs ; ils dérivent tous de l'apprentissage compétitif simple qui se décrit de la manière suivante :

- (1) Initialiser le dictionnaire ;
- (2) Tirer un vecteur d'apprentissage x ;
- (3) Rechercher le neurone i vainqueur de la compétition tel que :

$$d(X, w_i) \leq d(X, w_j); \forall j; \quad (3.10)$$

- (4) Mobiliser son vecteur poids : $y_i \leftarrow y_i + \epsilon(x - y_i)$;
- (5) Re boucler en 2 tant que le critère d'arrêt n'est pas atteint.

Avec d étant une mesure de distance, et w_i le meilleur représentant du vecteur X pris en entrée. Le paramètre ϵ est un coefficient d'apprentissage, il est compris entre 0 et 1 et décroît au cours de l'apprentissage. Le critère d'arrêt est généralement d'atteindre un nombre donné d'itérations fixé à l'avance [83].

Le dictionnaire initial est déterminant pour la réussite de l'algorithme [18]. Si un vecteur code est initialisé dans une zone de faible densité de probabilité, il interviendra peu dans l'apprentissage et sera ensuite peu utilisé lors du codage. Afin de résoudre le problème des unités sous-utilisées, l'apprentissage compétitif sensible à la fréquence [18, 133], introduit la fréquence d'utilisation des neurones dans l'apprentissage. L'étape 3 précédente utilise une mesure de distorsion modifiée : $d'(x, y_i) = F(F_i)d(x, y_i)$, où F_i désigne le nombre d'utilisations du neurone i depuis le début de l'apprentissage, et F est une fonction croissante. Ainsi, un neurone fréquemment choisi sera désavantagé dans la compétition.

Plusieurs auteurs soulignent la rapidité de construction du dictionnaire avec un apprentissage compétitif et la possible adaptation aux variations statistiques des images [83, 133, 18].

Ainsi la transformation d'images utilisant la quantification vectorielle produit des résultats proches de ceux de la *KL - transformation* lorsque le dictionnaire est correctement construit, et avec la taille des blocs d'entrée optimum [54, 98, 18]. Il existe

cependant d'autres algorithmes neuronaux, basés sur la règle de Hebb [84] permettant d'effectuer le même travail.

3.6 Réseaux de neurones basés sur l'apprentissage de Hebb

La structure générale du réseau de neurones consiste en une couche d'entrée et une couche de sortie qui est représentée à la figure 3.11 ci-après.

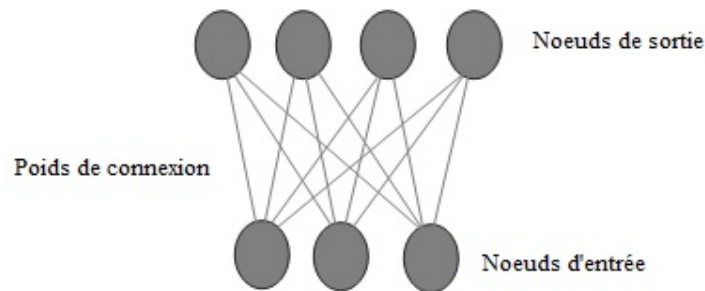


FIGURE 3.11 – Structure d'un réseau de neurones basée sur l'apprentissage de Hebb

Les règles d'apprentissage de *Hebb* viennent des considérations de *Hebb* [72] selon lesquelles si deux neurones sont très actifs au même moment (qui est illustré par des grandes valeurs de leurs sorties et de leurs entrées), les poids de connexion entre ces deux neurones augmenteront.

D'où pour les valeurs de sortie exprimées comme suit :

$$[h] = [W]^T \cdot [X], \quad (3.11)$$

la règle d'apprentissage peut être décrite de la manière suivante :

$$W_i(t+1) = \frac{W(t) + \alpha h_i(t) \cdot X(t)}{\|W_i(t) + \alpha h_i(t) \cdot X(t)\|} \quad (3.12)$$

Où $W_i(t+1) = (W_{i1}, W_{i2}, \dots, W_{iN})$ est le *ième* nouveau vecteur de couple de poids au temps $(t+1)$; avec $1 \leq i \leq M$ et M est le nombre de neurones en sortie. α est le taux d'apprentissage. $h_i(t)$ la *ième* valeur en sortie. $X(t)$ le vecteur en entrée correspondant à chaque bloc de l'image. $\|\cdot\|$ est la norme euclidienne utilisée pour normaliser les poids mis à jour et équilibrer l'apprentissage.

De l'apprentissage de Hebb ci-dessus, une règle d'apprentissage *linéarisée* est développée par *Oja* [123] en étendant l'équation (3.12) à des séries à partir desquelles la mise à jour de tous les couples de poids est construite ci-dessous.

$$W_i(t+1) = W_i(t) + \alpha[h_i(t).X(t) - h_i^2(t).w_i(t)] \quad (3.13)$$

Pour obtenir les M principaux composants, *Sanger* [142] étend le modèle ci-dessus à une règle d'apprentissage qui enlève les principaux composants attendus à travers l'orthogonalisation de *Gram - Schmidt* [66] et fait converger les couples de poids vers les principaux composants dérivés.

Les performances de ce réseau de neurones dépendent du nombre de neurones en sortie (c'est à dire la valeur de M). La qualité de la reconstruction de l'image est déterminée par l'erreur quadratique moyenne :

$$e_{ms} = E((x - \bar{x}^2)) = \sum_{j=M+1}^N \lambda_j \quad (3.14)$$

avec des vecteurs de dimensions N en entrée et M neurones en sortie. Et les λ_j représentent les valeurs propres de la matrice covariance du vecteur x en entrée.

3.7 Conclusion

Les réseaux de neurones sont donc partis du modèle simple du *neurone formel* proposé par *Pitts* et *McCulloch* dans leur article "*what the frog's eye tells to the frog's brain*" [105], jusqu'au modèle du perceptron multicouche proposé par *Rumelhart* [139]. Ils sont utilisés dans de nombreux domaines d'application (traitement du signal vocal, IRD sonore, ...) [18] et varient suivant leur structure et le mode d'apprentissage. Les différentes étapes de conception d'un réseau de neurones (choix et préparation des échantillons, élaboration de la structure du réseau, apprentissage, validation et tests) leur donnent un caractère particulier qui leur permet de s'adapter aux données à traiter et donc d'être potentiellement applicables dans plusieurs domaines d'activités. Par ailleurs, l'utilisation des réseaux de neurones comme classificateurs ont prouvé leur efficacité dans plusieurs champs d'application. Leurs capacités à s'adapter aux données à traiter jumelées à la possibilité d'effectuer les calculs en parallèle qu'ils offrent, les rendent très appropriés pour le traitement intelligent du signal ceci dans divers champs d'application en général et dans celui de la reconnaissance de motifs en particulier. Par ailleurs, plusieurs techniques utilisant les réseaux de neurones ont été proposées pour réaliser des opérations d'identification ou d'authentification d'individus à partir d'empreintes digitales qu'il serait nécessaire d'examiner. L'un des aspects majeurs sur lequel

ce travail insiste est l'utilisation des techniques neuronales pour réaliser l'authentification d'un individu qui souhaite accéder à un système sécurisé privé. Et, puisque l'authentification appartient à la classe de problèmes de la classification, c'est l'apprentissage supervisé que nous utiliserons pour la suite de ces travaux.

La reconnaissance d'empreintes digitales avec les réseaux de neurones

4.1 Introduction

Les chapitres précédents ont montré que la reconnaissance d'empreintes digitales est une technique biométrique mature qui permet de sécuriser un système informatique par l'authentification ou l'identification des personnes. Elle consiste à prendre une image de l'empreinte en entrée et de lui appliquer différentes opérations de comparaison avec un ensemble de données préalablement classifiées. Le principe d'identification ou d'authentification reste donc sensiblement le même pour toutes les différentes techniques employées ; il s'agit toujours d'extraire certaines caractéristiques de l'image d'empreinte sous forme d'informations codées et d'y appliquer les opérations de comparaison nécessaires.

D'une manière générale on distingue deux catégories d'algorithmes de reconnaissances d'empreintes digitales : la première catégorie concerne les algorithmes plutôt "*conventionnels*" qui s'appuient sur la position relative des minuties entre elles, alors que la seconde regroupe les algorithmes visant à extraire d'autres particularités de l'empreinte digitale telles que :

- *la direction locale des sillons* [28, 33], ou encore
- *les composantes fréquentielles locales de la texture au cœur de l'image* [78].

Ces techniques ont été utilisées jusqu'ici pour réaliser la reconnaissance d'empreintes digitales simples ou latentes et produisent des résultats plus ou moins satisfaisants en terme de taux de reconnaissance et de taux de faux rejets. Ces méthodes effectuent la reconnaissance en déterminant la position des minuties au sein de l'image pour quantifier les caractéristiques de ressemblance entre deux gabarits par "*point pattern matching*". Une autre façon plus intéressante de réaliser la reconnaissance est de coder non pas les valeurs individuellement les unes après les autres, mais de coder un ensemble de valeurs

simultanément.

Les approches utilisant les Réseaux de Neurones Artificiels (RNA) pour le traitement des images semblent être très prometteuses, en raison notamment de la possibilité d'effectuer les calculs en parallèle qu'ils offrent et leur processus d'apprentissage qui permet au système de s'adapter aux données à traiter [125, 159]. Ceci est particulièrement important, puisque le scanning de l'empreinte digitale peut inclure plusieurs sources différentes de déformation de son image (la rotation, la pression exercée sur le doigt, l'affectation cutanée, la qualité du scanner en lui-même, etc.), permettant ainsi d'étendre le bruit sur une bonne partie de l'image. Par ailleurs lorsque les réseaux de neurones artificiels sont utilisés convenablement dans la reconnaissance d'empreintes digitales, ils produisent des résultats très intéressants en terme de taux de faux rejets, de temps de reconnaissance et d'efficacité dans le traitement de grandes bases de données. C'est le cas des algorithmes utilisant les principes du "deep learning" [17, 103].

Par exemple une application conçue pour gérer l'accès à une résidence privée ou à un système de formation en ligne, implique un besoin d'une petite base de données modifiable de quelques personnes et d'un temps de réponse généralement de l'ordre de quelques secondes. Dans ce scénario, la personne a déjà fait état de son identité et l'empreinte digitale biométrique est utilisée juste pour confirmer cette prétention.

Par contre, une application d'identification d'individus sur des scènes publiques à des fins policières, exige des recherches rapides sur des très grandes bases de données avec un temps de réponse qui peut être plus long. Les réseaux de neurones peuvent être adaptés et formés différemment pour ajuster l'exigence particulière de chaque demande.

Les techniques utilisant les réseaux de neurones artificiels pour la reconnaissance d'empreintes digitales sont nombreuses et diversifiées. Ce travail, s'intéresse aux approches spécifiques à l'authentification d'une part et à celles propres à l'identification d'autre part. Ensuite il présente une nouvelle méthode basée sur l'utilisation du perceptron et de l'algorithme de rétro propagation du gradient de l'erreur, orientée vers l'authentification de personnes qui souhaitent accéder à un système de façon automatique. Mais avant, le concept d'apprentissage profond (ou *deep learning* en anglais) sera présenté.

4.2 *Présentation des méthodes*

D'un point de vue technique, il existe deux problèmes différents dans le domaine de l'analyse des empreintes digitales, à savoir : la classification et la correspondance. La classification consiste à mettre certaines empreintes digitales qui ont des gabarits qui s'apparentent dans des sous-classes prédéfinies. De nos jours, il existe cinq classes

d'empreintes digitales définies par *Henry* [74].

La correspondance qui sous-entend l'authentification et l'identification quant à elle, repose sur d'autres éléments plus subtils (minuties, orientations des stries) et nécessite une analyse beaucoup plus fine de l'image d'empreinte. Ce n'est donc plus l'image entière qui est prise en considération ici, mais un ensemble d'éléments qui pris globalement constitue le gabarit de l'empreinte : il s'agit des minuties et de leurs orientations relatives au moment de faire la correspondance. Cette technique implique donc l'utilisation d'opérations de pré-traitement (binarisation, squelettisation, filtrage, etc.) qui vont permettre le raffinement de l'image afin d'extraire les caractéristiques les plus utiles.

Dans le cadre ce travail, deux méthodes qui traitent exclusivement l'aspect de la correspondance sont présentées ; celui de la classification étant standardisé. Ces méthodes ayant chacune sa spécificité : l'une étant propice à l'application aux petites bases de données (propre à l'authentification) et l'autre étant favorable à la recherche dans un grand ensemble de données (spécifique à l'identification). Mais avant tout cela, cette section présente le concept d'apprentissage profond et quelques pistes d'optimisation d'algorithmes de ce type d'apprentissage.

4.2.1 *Apprentissage profond*

Définition

L'apprentissage profond (ou *deep learning* en anglais) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur, du traitement automatisé du langage, etc.

Les techniques d'apprentissage profond constituent une classe d'algorithmes d'apprentissage automatique :

- Utilisant différentes couches d'unité de traitement non-linéaire pour l'extraction et la transformation des caractéristiques. Chaque couche prend en entrée la sortie de la précédente. Les algorithmes peuvent être supervisés ou non supervisés et leurs applications comprennent la reconnaissance de modèles ou la classification statistique.
- Elles fonctionnent avec un apprentissage à plusieurs niveaux de détails ou de représentations des données. À travers les différentes couches on passe de paramètres de bas niveau à des paramètres de plus haut niveau.

- Ces différents niveaux correspondent à différents niveaux d'abstraction des données.
- Ce nouveau champ d'étude a pour objectif d'avancer davantage vers les capacités d'intelligence artificielle. Ces architectures permettent aujourd'hui de donner du sens à des données sous forme d'image, de son ou de texte.

L'apprentissage profond utilise des couches cachées de réseaux de neurones artificiels, des "*Restricted Boltzmann machines*", et des séries de calculs propositionnels complexes. Les algorithmes d'apprentissage profond s'opposent aux algorithmes d'apprentissage peu profonds du fait du nombre de transformations réalisées sur les données entre la couche d'entrée et la couche de sortie, où une transformation correspond à une unité de traitement définie par des poids et des seuils.

Contexte

Le "*deep learning*" fait partie d'une famille de méthodes d'apprentissage automatique fondées sur l'apprentissage de modèles de données. Une observation (comme une image) peut être représentée de différentes façons par un vecteur de données en fonction de :

- l'intensité des pixels dont elle est constituée,
- ses différentes arêtes,
- les différentes régions de forme particulière,
- etc.

Certaines représentations, et une bonne capacité d'analyse automatique des différentiations rendent la tâche d'apprentissage plus efficace.

Une des perspectives des techniques de l'apprentissage profond est le remplacement de travaux qui aujourd'hui sont encore relativement laborieux par des modèles algorithmiques d'apprentissage supervisé, non-supervisé (c'est-à-dire ne nécessitant pas de connaissances spécifiques du problème étudié) ou encore par des techniques d'extraction hiérarchique des caractéristiques.

Les recherches dans ce domaine [103, 144] s'efforcent de construire de meilleures représentations du réel et de créer des modèles capables d'apprendre ces représentations à partir de données non labellisées à grande échelle. Certaines de ces représentations s'inspirent des dernières avancées en neurosciences qui sont grosso modo des interprétations du traitement de l'information et des modèles de communication du système nerveux, de la même façon que le système nerveux tente d'établir des connexions en fonction des messages reçus, de la réponse neuronale consécutive et du poids des connexions entre les neurones du cerveau.

Les différentes architectures du "*deep learning*" telles que les "*deep neural networks*", les "*convolutional deep neural networks*", et les "*deep belief network*" ont des champs

d'application tels que la vision par ordinateur, la reconnaissance automatique de la parole, le traitement automatique du langage naturel, la reconnaissance audio et la bio-informatique où elles ont démontré qu'elles étaient capables de produire d'excellents résultats pour différentes problématiques [144].

Pourquoi l'apprentissage profond ?

L'apprentissage profond consiste à entraîner des modèles composés de plusieurs transformations non-linéaires successives. Un RNA avec plusieurs couches cachées en est donc un exemple. Pourtant, nous avons vu qu'un RNA avec une couche cachée est déjà un approximateur universel, quel est alors l'avantage d'augmenter le nombre de couches ?

Il y a plusieurs arguments en faveur de tels réseaux. Premièrement, des résultats mathématiques montrent que certaines classes de fonctions, représentables de manière compacte avec un réseau de profondeur d , nécessiteraient un nombre exponentiel de paramètres avec un réseau de profondeur $d - 1$ [17, 70]. Ensuite, on peut noter que l'architecture des réseaux profonds permet la réutilisation de paramètres ou de caractéristiques extraites, propriété désirable pour la modélisation de fonctions complexes. Et finalement, le cerveau humain, plus particulièrement le cortex visuel, est construit comme une architecture profonde, avec plusieurs sous-régions fonctionnelles (V1, V2, MT) arrangées en niveaux d'abstraction de plus en plus élevés.

Une des motivations est donc d'apprendre une hiérarchie de caractéristiques avec ces niveaux d'abstraction croissants. Par exemple, pour une tâche de reconnaissance d'objet, on aimerait que la première couche détecte des parties de bas niveau, telles que les bords ; que la suivante détecte des combinaisons de ses parties ; et ainsi de suite, pour représenter des concepts de plus en plus évolués. Ceci est observé expérimentalement avec les méthodes utilisées en apprentissage profond [106, 173].

En plus d'extraire des caractéristiques de plus en plus abstraites, une autre propriété désirable serait de les démêler, afin que les différents facteurs qui expliquent les variations des données soient séparés. Ceci serait très intéressant pour l'apprentissage multi-tâches, ainsi que pour l'adaptation de domaine [53, 62].

Avant 2006, l'entraînement des réseaux de neurones était limité aux réseaux avec une seule, voir deux couches cachées, à l'exception des réseaux convolutionnels [103] ; les réseaux plus profonds donnant de moins bonnes performances. Ceci était dû aux raisons suivantes :

- les performances computationnelles insuffisantes du matériel informatique ;
- la quantité trop petite de données d'entraînement disponible ;

- les difficultés inhérentes à l'apprentissage profond.

Concernant les deux premiers points, la situation ne cesse de s'améliorer. Qu'en est-il du dernier ? Les principales difficultés sont les problèmes des minima locaux et celui des gradients diminuant/augmentant [17, 144]. Ce dernier concerne les réseaux récurrents, c'est-à-dire des réseaux admettant des boucles dans leurs connections. Une façon de les entraîner consiste à rétro-propager les gradients à travers le temps. Il peut être montré que la norme de ces gradients tend vers 0 ou vers l'infini à une vitesse exponentielle. Or, un réseau récurrent peut être vu comme un réseau profond avec une infinité de couches. Cela suggère que la rétro-propagation des erreurs est rendue difficile par la profondeur [103]. La suite présente plusieurs solutions qui ont permis de surpasser ces difficultés.

Quelques solutions

Les solutions utilisées pour apprendre de telles architectures sont variées, quelques-unes, notables sont mentionnées ici. Comme cela a déjà été souligné, les réseaux de neurones convolutionnels [104] ont été les premières architectures profondes entraînées efficacement. Ils sont inspirés d'observations du traitement de l'information dans le cortex visuel où les neurones sont organisés en colonnes corticales. Le réseau n'admet que des connections locales et, de plus, les paramètres de ces connections sont partagés au sein d'une couche donnée : cela réduit considérablement le nombre de paramètres du modèle. Mathématiquement, la multiplication matricielle est remplacée par une convolution : les caractéristiques sont extraites sur tout le champ visuel plutôt qu'être spécifique à une région de ce dernier. En addition, une couche d'agrégation, ou "*pooling*", est souvent utilisée pour réduire la taille des couches cachées (dans le cas convolutionnel appelées cartes de caractéristiques), cela permet aussi d'obtenir une invariance à des petites translations. La réduction du nombre de paramètres et le choix de l'architecture, appropriée au type de données à traiter, permet de faciliter l'apprentissage ; même dans le cas d'architectures profondes. Ce type de réseaux est très performant pour les tâches de traitement d'images et est également utilisé pour le traitement de vidéo, d'audio et de texte [103, 106]. Plus généralement, il est possible de l'appliquer pour toutes les données admettant une composante séquentielle et/ou temporelle et/ou spatiale.

Une autre orientation de la recherche pour faciliter l'entraînement des réseaux profonds consiste à améliorer l'algorithme d'optimisation, plutôt que d'utiliser la simple descente de gradient stochastique. Les méthodes de second ordre [173] utilisent l'information de courbure de la fonction de coût pour sélectionner automatiquement le taux d'apprentissage optimal (si la fonction objective est quadratique) pour chaque paramètre. Cela nécessite d'estimer la fonction de coût, ce qui est coûteux en terme de

calcul. De plus, cette dernière doit être définie positive, en d'autres termes, la fonction de coût doit être localement convexe, ce qui n'est pas toujours le cas.

La méthode des gradients naturels, elle, utilise la géométrie de la variété définie par les paramètres du modèle. Il est également possible d'utiliser une mesure d'incertitude sur les gradients, par exemple par le biais de la matrice de covariance de ces derniers, pour les normaliser [53]. Ce ne sont que quelques exemples de méthodes utilisées pour améliorer l'optimisation des réseaux profonds, une énumération exhaustive nécessiterait un chapitre entier. Glorot Xavier a prouvé qu'il est possible de faciliter l'optimisation de réseaux profonds, même dans le cas de la descente de gradient stochastique simple, en faisant juste attention à l'initialisation aléatoire des paramètres et au choix de la fonction d'activation [62].

L'utilisation de régularisations appropriées est aussi une solution permettant de faciliter l'apprentissage des architectures profondes. Un exemple récent est la méthode dite "*Dropout*" [158]. Il s'agit, pendant l'apprentissage, de masquer aléatoirement une fraction des unités cachées du réseau. Ce bruit permet de mieux utiliser la capacité du modèle car l'information doit être distribuée sur les unités pour résoudre la tâche. Cela peut-être vu comme l'apprentissage d'un ensemble de classifieurs qui partagent leurs paramètres. Pour la prédiction, on utilise le modèle complet, mais on multiplie les poids par la fraction de bruit utilisé pendant l'apprentissage, afin de conserver la valeur moyenne des activations des neurones identiques à celles durant l'apprentissage.

Bengio et al. [17] ont montré qu'il est possible d'entraîner des réseaux profonds en utilisant un apprentissage non-supervisé couche-par-couche, ou pré-entraînement, pour initialiser les paramètres. Une étude montre que l'apprentissage non-supervisé agit comme un régularisateur et qu'il permet de placer les paramètres dans un meilleur bassin d'attraction, associé à un minimum local avec une meilleure capacité de généralisation [53]. Ceci est à l'origine de la renaissance des architectures profondes. Cependant, avec les dernières avancées, comme la méthode "*Dropout*" et l'utilisation des unités rectificatives, le pré-apprentissage ne semble plus nécessaire dans le cas supervisé avec beaucoup d'exemples d'entraînements. Cependant, dans le cas semi-supervisé ou en transfert de l'apprentissage, cette méthode reste centrale. Qian et al., dans son article [129], a proposé un modèle de CNN qui, appliqué à la vérification d'empreintes digitales, par le biais du système Fingerprint Verification for Convolutional Neural Network (FVCNN), produit des résultats intéressants bien que perfectibles.

4.2.2 Méthode basée sur les réseaux de neurones probabilistes

Les fondements scientifiques de la méthode élucidée ci-dessous ont été introduits par Baldi et Chauvin en 1993 [11]. Cette méthode qui traite exclusivement l'aspect de la

correspondance consiste en trois étapes principales, à savoir : une étape de construction de la base de données, une étape de pré-traitement et une étape de décision.

L'étape de construction de la base de données a pour principal objectif de construire un ensemble de données qui servira de base d'application et d'exploitation des différentes opérations à effectuer dans les deux étapes suivantes. La phase de pré-traitement aligne essentiellement les deux images et extrait de chacune d'elle une *région centrale*. Les deux régions centrales sont utilisées pour alimenter l'étape de décision, qui est la partie du réseau de neurones la plus appropriée de l'algorithme et qui est soumise à la formation des exemples. Pendant que l'étape de pré-traitement est assez standard (c'est à dire utilise les algorithmes de pré-traitement existants [5, 78, 116]), la phase de décision proposée par *Baldi* et *Chauvin* quant à elle est assez novatrice et est basée sur des réseaux de *neurones probabilistes* qui permettent d'établir la correspondance entre deux gabarits A et B en évaluant la probabilité P que A et B soient identiques [36, 69]. Le réseau est formé par la descente du gradient en utilisant un ensemble d'apprentissage de n paires d'images venant de plusieurs doigts différents. L'ajout de quelques doigts supplémentaires pour effectuer la phase d'utilisation vient consolider la robustesse du réseau de neurones créé et testé préalablement.

4.1.2.1 Construction de la base de données

Bien qu'il existe dans le monde entier plusieurs bases de données d'empreintes digitales (Automated Fingerprint Identification System (*AFIS*) d'*INTERPOL* avec plus de 150 000 empreintes ; l'Autorité Unique d'Identification de l'Inde encore connu sous le nom *Aadhaar*, avec plus 734 millions d'empreintes digitales, de visage et d'iris ; etc. . .), celles-ci ne sont pas généralement disponibles pour l'utilisation publique. De plus, se pose aussi la question des approches connexionistes, car en effet la plupart des bases de données contiennent seulement une seule image par doigt. Ceci peut constituer un facteur négatif dans la construction du réseau de neurones qui, pour être efficace lors de la phase d'utilisation nécessite que plusieurs versions (parfois bruyantes) de la même image soient traitées au préalable. Pour construire un réseau de neurones pour l'apprentissage et l'utilisation, il faut d'abord constituer une base de données d'images d'empreintes digitales numérisées. De telles images peuvent être obtenues de différentes façons, par exemple par le scanner numérique (voir section 1.3) avec des empreintes digitales signées ou par des techniques holographiques plus sophistiquées [8, 130].

Mais de façon générale, les auteurs recommandent à chaque utilisateur de construire sa propre base de données avec des personnes coopératives et sensibilisées sur l'importance de l'opération engagée. L'un des moyens les plus utilisés pour obtenir une image d'empreinte est la camera *CCD* dont le principe peut être décrit à la figure 4.1 ci-dessous.

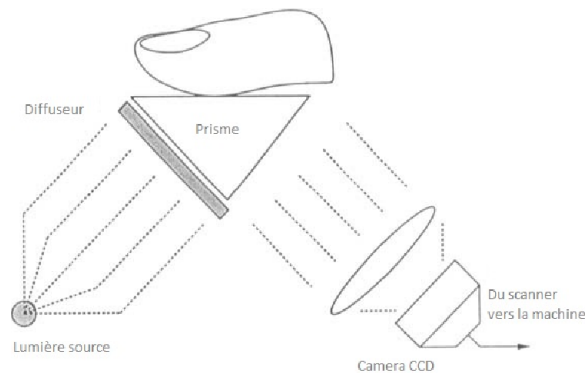


FIGURE 4.1 – Processus d’acquisition d’une image à l’aide d’une camera CCD [169].

Certaines images sont latentes du fait que le prisme du scanner utilisé par les experts n’avait pas été nettoyé [11, 71]. Ce qui laisse apparaître une différence parfois significative sur deux images provenant du même doigt.

Proposition

Soient deux images d’empreintes digitales A et B , la décision que ces deux images sont correspondantes (ou pas) sera dénotée par $M(A, B)$ (ou $\bar{M}(A, B)$). L’objectif de cette technique est alors de concevoir un algorithme basé sur les réseaux de neurones qui prend en entrée une paire d’images d’empreintes digitales (A, B) et donne en sortie un nombre $p = p(M) = p(M(A, B))$ compris entre 0 et 1 correspondant à la probabilité que les deux empreintes soient la même.

4.1.2.2 Opérations de pré-traitement

Tout algorithme de reconnaissance d’empreintes digitales devrait commencer par des opérations de pré-traitement qui permettent d’éliminer le bruit présent sur l’image d’origine. Ces opérations consistent entre autres à : la diminution du contraste, la segmentation, la squelettisation, la compression [111], etc. Le but du pré-traitement ici, est d’extraire de chacune des deux images en entrée une pièce centrale appelée "*région centrale*" et de les aligner. Seul les deux régions centrales sont à leur tour passées à l’étape de décision. Le processus de pré-traitement en lui même consiste en plusieurs étapes : d’abord filtrer le bruit des hautes fréquences, ensuite pallier aux effets de la translation présent dans les images et procéder à la segmentation, enfin aligner et compresser les régions centrales. Pour faciliter la description de la méthode, une des images en entrée sera appelée "*image de référence*" et l’autre "*image de test*", bien qu’il n’y a aucune différence intrinsèque entre les deux.

i) Filtrage passe bande-bas

Pour se débarrasser des nombreuses pointes de haute fréquence qui semblent être présentes dans les images originales, les auteurs ont remplacé chaque pixel qui dévie significativement des valeurs de ses quatre voisins par la moyenne correspondante.

ii) Segmentation

Pour chaque image, l'algorithme dessine d'abord un bloc rectangulaire serré autour de chaque empreinte digitale utilisant un algorithme de détection de bord et détermine le centre géométrique du bloc [11]. La région centrale de l'image de référence est alors définie pour être $n \times n$ pièces carrées centrales qui occupe la région immédiatement au-dessous du centre précédemment décrit. Pour l'image de test, au lieu de cela les auteurs choisissent une pièce semblable mais plus grande de taille $m \times m$ (l'extension de la pièce précédente par 20 pixels dans chaque direction). Cette plus grande pièce est nommée la *fenêtre* [11, 147].

iii) Alignement

La région centrale de l'image de référence est glissée pixel par pixel à travers la vitrine de l'image test (par 20 pixels en haut, en bas, à gauche et à droite) et la corrélation correspondante est calculée à chaque étape, jusqu'à ce que la position où la corrélation est maximale soit trouvée. Cette étape en dehors de celle de l'entraînement du réseau est la plus coûteuse de l'algorithme en terme de nombre de calculs effectués. La région centrale de l'image test est alors déterminée en sélectionnant le bloc central de $n \times n$ pixels correspondant à la position maximale de la corrélation calculée.

iv) Compression et normalisation

Finalement, chacune des deux $n \times n$ régions centrales est réduite à un tableau de taille $k \times k$ (avec $k < n$ et k étant une puissance de 2) par la convolution discrète avec une fonction gaussienne tronquée de taille 5×5 [11, 42, 147]. Cette région centrale compressée de taille $k \times k$ contient une image de basse résolution, qui correspond grossièrement à 10 minutes dans l'image originale. Les valeurs de pixels résultantes sont commodément normalisées entre 0 et 1 [11, 83]. Dans l'implémentation, tous les paramètres (et particulièrement la taille des blocs rectangulaires) sont ajustables. Pour éviter les effets de frontière, un large cadre de x pixels ($x = 2$ ici) est habituellement ajouté autour des blocs rectangulaires, ce qui pourrait expliquer des tailles impaires de certains blocs. Il est également naturel de se demander à ce stade si la décision quant

à la correspondance de deux gabarits basée uniquement sur le calcul d'une corrélation maximale des blocs de pixels voisins ne devrait pas être contrôlée par un seuil prédéfini.

En effet, il est important de noter que le calcul empirique de cette corrélation maximale n'est pas suffisant pour garantir totalement la correspondance, ceci est dû au bruit qui pourrait être présent dans l'un des blocs de l'image. En particulier, les corrélations tant de correspondance que de non correspondance sont souvent très élevées (au dessus de 0.9) et nous observons parfois des cas où la corrélation de non correspondance est supérieure à celle de correspondance des paires d'images. Il est donc essentiel d'avoir une étape de décision non-linéaire après le pré-traitement. Finalement, on devrait remarquer qu'aussi bien pendant la phase d'apprentissage que celle du test, le pré-traitement doit être appliqué une seule fois à chaque paire d'images. Principalement, seules les régions centrales ont besoin d'être passées à travers le réseau de neurones durant la phase d'apprentissage. Bien que le pré-traitement ne soit pas soumis à l'apprentissage, il peut être mis en œuvre, pour la plupart de sa partie, d'une façon parallèle compatible avec une architecture neuronale globale pour l'algorithme entier.

4.1.2.3 Étape de décision du réseau de neurones

Cette étape est la plus importante pour un algorithme basé sur les réseaux de neurones. Comme dans certains articles, [100, 84] le réseau de neurones utilisé ici a une structure pyramidale avec les deux régions centrales compressées et alignées en entrée du réseau et un seul résultat en sortie appelé p . Le niveau le plus bas de la pyramide correspond à une convolution des régions centrales avec plusieurs filtres. Les couches suivantes sont nouvelles. La décision finale qu'elles calculent provient d'un modèle bayésien probabiliste pour l'estimation de p , basé sur la sortie du filtre de convolution [11]. L'étape de filtrage et celle de décision sont effectuées simultanément lors de l'entraînement.

i) Convolution

Les deux régions centrales sont d'abord contrôlées avec un ensemble de filtres réglables. Dans cette implémentation, seulement deux filtres différents sont utilisés, mais une extension à un nombre plus élevé est simple. Ici, chaque filtre a un champ réceptif de taille $q \times q$ et les champs réceptifs de deux filtres voisins du même type ont un chevauchement de 2 pixels pour se rapprocher d'une opération de convolution continue. La sortie de tous les filtres d'un type donné forme alors un tableau de taille $r \times r$. Ainsi chaque région centrale de taille $k \times k$ (ici $k = 32$) est transformée en plusieurs tableaux de taille $b \times b$ (un tableau pour chaque type de filtre). La sortie du filtre de type j à la

position (x, y) dans un de ces tableaux est donnée (par exemple pour A) par :

$$z_{x,y}^j(A) = f \left(\sum_{r,s} w_{x,y,r,s}^j I_{r,s}(A) + t^j \right) \quad (4.1)$$

Où :

- $I_{r,s}(A)$ est l'intensité du pixel dans la région centrale compressée de l'image A à la position (r, s) ;
- f est une fonction sigmoïde [$f(x) = (1 + e^{-x})^{-1}$] ;
- $w_{x,y,r,s}^j$ est le poids de connexion entre le pixel à la position (r, s) dans la région centrale compressée et le pixel à la position (x, y) dans le tableau du filtre en sortie ;
- et t^j est le biais.

La somme de l'équation 4.1 est calculée sur un bloc $q \times q$ correspondant au champ réceptif du filtre à la position (x, y) . Le seuil et le modèle de poids $q \times q$ sont les caractéristiques du type filtre (prétendu "partage de poids"), ils peuvent donc être considérés comme des paramètres d'une translation invariante de convolution de noyau. Ils sont partagés entre une image mais aussi à travers les images A et B . Ainsi,

$$w_{x,y,r,s}^j = w^j(x - r, y - s);$$

Et dans ce cas, chaque type de filtre est caractérisé par $q \times q + 1$ paramètres entraînaibles. Dans la suite, la notation de l'emplacement du filtre en sortie sera simplifiée par la position $(x, y) = i$. Pour chaque filtre de type j , l'on peut maintenant former un tableau $\Delta z^j(A, B)$ composé de toutes les différences carrées ainsi qu'il suit :

$$\Delta_i^j(A, B) = [z_i^j(A) - z_i^j(B)]^2 \quad (4.2)$$

Et $\Delta z = \Delta z(A, B)$ indique le tableau de tous les $\Delta_i^j(A, B)$ pour toutes les positions i et tous les types de filtres j .

ii) *Décision*

L'objectif de cette partie du réseau est d'estimer la probabilité :

$$p = p[M(A, B)/\Delta z(A, B)] = p(M/\Delta z);$$

d'une correspondance entre A et B , étant donnée la preuve Δz provenant du filtre de convolution. L'étape de décision peut être vue comme une classification bayésienne binaire. Baldi [11] propose quatre ingrédients clés pour la décision du réseau. La structure du réseau de neurones utilisé par Baldi peut être décrit par la figure 4.2 ci-après.

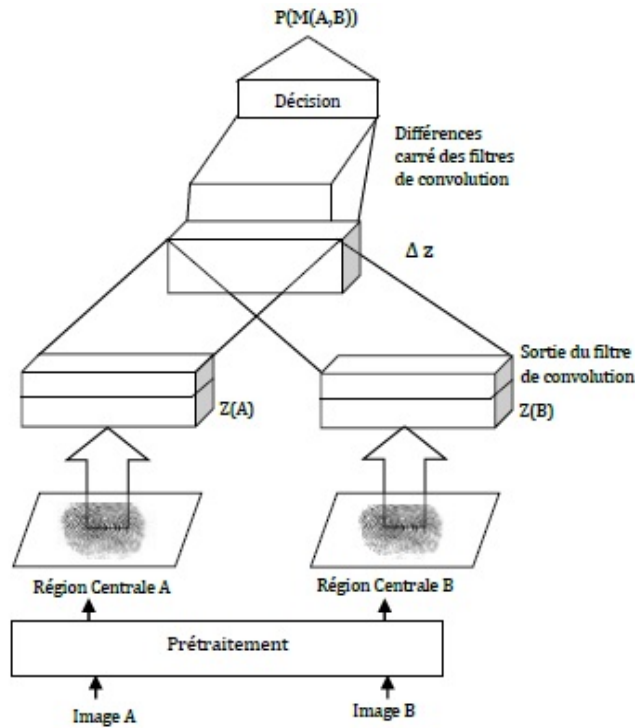


FIGURE 4.2 – Structure du réseau de neurones probabiliste.

(1) Étant donné que la sortie du réseau sera interprétée comme une probabilité, l'erreur quadratique minimum habituellement utilisée dans les algorithmes de rétro-propagation semble ne pas être appropriée comme mesure de performance pour ce réseau. Pour les distributions de probabilité, l'entropie croisée entre la probabilité estimée en sortie p et celle obtenue P , additionnée à tous les modèles est une information bien connue comme mesure de divergence [20].

$$H(P, p) = \sum_{(A,B)} P \log \frac{P}{p} + Q \log \frac{Q}{q} \quad (4.3)$$

Où, pour chaque paire d'image, $Q = 1 - P$ et $q = 1 - p$. H est aussi perçu comme une fonction discrète et peut être prise pour la valeur attendue du ratio de vraisemblance des deux distributions. La distinction est non négative, convexe en chaque argument et égale à 0 si et seulement si ces arguments sont égaux.

(2) En utilisant une formule d'inversion de Bayes de la dépendance des paires (A,B) on a :

$$p(M/\Delta z) = \frac{p(\Delta z/M)p(M)}{p(\Delta z/M)p(M) + p(\Delta z/\bar{M})p(\bar{M})} \quad (4.4)$$

L'effet des préalables $p(M)$ et $p(\bar{M})$ devrait être sans apport dans les cas d'une grande base de données (notre base de données est assez large pour que les décisions soient ren-

dues en fonction des données). Dans les simulations les valeurs choisies sont : $p(M) = 0.1$ et $p(\bar{M}) = 0.9$. La valeur observée de $p(M)$ est approximativement 16% dans la phase d'apprentissage et de 4% dans la phase de test avec la base de données entière.

(3) Considérons l'hypothèse que :

$$p(\Delta z/M) = \prod_{i,j} p(\Delta z_i^j/M) \tag{4.5}$$

$$p(\Delta z/\bar{M}) = \prod_{i,j} p(\Delta z_i^j/\bar{M}) \tag{4.6}$$

Strictement, cela n'est pas vrai, spécialement pour les blocs voisins. Toutefois, dans le centre de l'empreinte digitale où il y a plusieurs variabilités dans les zones périphériques, c'est une approximation raisonnable qui peut conduire à une architecture simple du réseau de neurones avec des bons résultats en terme de faux de rejet.

(4) Pour définir entièrement notre réseau de neurones, l'on a toujours besoin de choisir un modèle pour la distribution conditionnelle $p(\Delta z_i^j/M)$ et $p(\Delta z_i^j/\bar{M})$. Dans le cas d'une correspondance, la probabilité $p(\Delta z_i^j/M)$ devrait être une fonction décroissante de Δz_i^j . Il est donc naturel de proposer un modèle exponentiel de la forme [39, 50] :

$$p(\Delta z/\bar{M}) = C_{ij} s_{ij}^{\Delta z_i^j} \tag{4.7}$$

Où $0 < s_{ij} < 1$ et, pour une normalisation convenable, la constante C_{ij} doit prendre la valeur $C_{ij} = \log s_{ij} / s_{ij} - 1$. Dans ce qui suit, un modèle binomial moins général mais un peu plus simple sera néanmoins utilisé

$$p(\Delta z_i^j/\bar{M}) = p_j^{1-\Delta z_i^j} (1 - p_j)^{\Delta z_i^j} = p_j \left(\frac{1 - p_j}{p_j} \right)^{\Delta z_i^j} \tag{4.8}$$

$$p(\Delta z_i^j/M) = q_j^{\Delta z_i^j} (1 - q_j)^{1-\Delta z_i^j} = (1 - q_j) \left(\frac{q_j}{1 - q_j} \right)^{\Delta z_i^j} \tag{4.9}$$

Avec $0.5 \leq p_j, q_j \leq 1$. Ceci est encore uniquement une approximation utilisée pour sa simplicité et parce que les différences caractéristiques Δz_i^j sont généralement comprises entre 0 et 1. Ici, les paramètres ajustables p_j et q_j dépendent uniquement du type de filtre, dans un cas plus général ils pourraient dépendre aussi de la position des pixels.

Bilan

En résumé, les paramètres ajustables du réseau de neurones sont $w_{x,y,r,s}^j$, t^j , p_j et q_j . A première vue, cela peut sembler être trop grand, dû au fait que le réseau est entraîné

en utilisant un ensemble de paires d'images. En réalité chacun des paramètres partagés correspond à un poids et à un biais de chacun des filtres de convolution et est entraîné sur un assez grand ensemble de données [55, 135, 162]. Les paramètres sont initialisés aléatoirement (par exemple les $w_{x,y,r,s}^j$ sont prélevés indépendamment d'une distribution gaussienne de moyenne 0 et de déviation 0.5). Ils sont alors séquentiellement ajustés, après la présentation de chaque paire d'exemple, en appliquant le gradient sur la mesure d'erreur de l'entropie croisée H . La formule d'ajustement de chacun de ces paramètres peut dériver des équations (4.3) et (4.8).

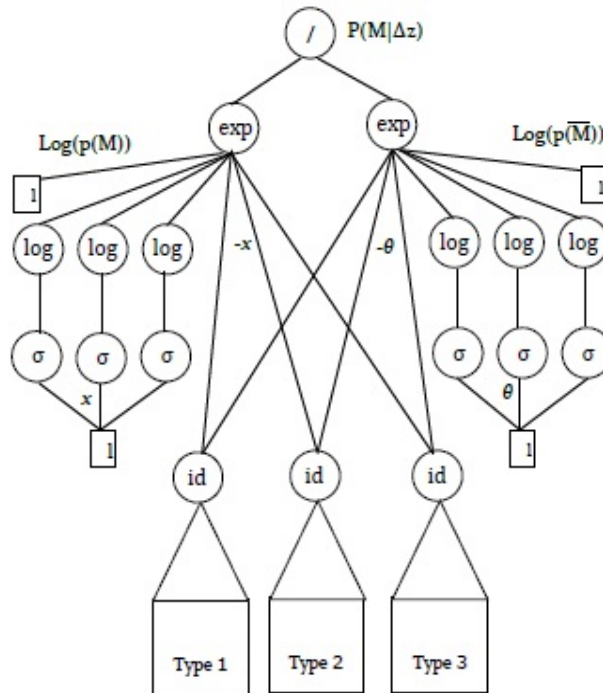


FIGURE 4.3 – Réseau de Baldi pour la décision.

Il est également à noter que les modèles de classification modifiables définis par les équations (4.3) et (4.8) ne sont pas des réseaux de neurones dans le sens strict du terme en ce qui concerne les couches et une fonction sigmoïde. C'est plutôt un modèle non linéaire avec des paramètres ajustables qui peut être décrit par la figure 4.3. Le nombre d'éléments et leurs types dépendent néanmoins de comment on décide de décomposer l'étape algébrique impliquée dans le calcul de la sortie final p .

4.1.2.4 Performances de la méthode probabiliste

Cette technique qui est basée sur l'utilisation des réseaux de neurones probabilistes présente les caractéristiques suivantes :

Avantages

- Utilisation d'une région centrale de l'image avec alignement et compression des données.
- Calcul des probabilités pour effectuer la correspondance entre deux patterns fournis en entrée du réseau après plusieurs filtrages successifs.
- L'algorithme est adapté pour l'authentification d'un individu qui souhaite accéder à un système dont son empreinte est préalablement connue.
- L'algorithme est propice pour les systèmes utilisant une petite base de données.

Limites

- Utilisation des formules complexes pour le calcul des probabilités.
- Taux de faux rejets relativement élevés.
- Utilisable pour un nombre d'images limitées y compris pour la phase d'apprentissage.

4.2.3 Méthode basée sur le réseau de neurones localement connectés

Un bon système d'identification doit avoir une haute exactitude, une phase d'apprentissage courte, un taux d'erreur bas et une grande robustesse pour traiter facilement le cas des objets non encore rencontrés. L'identification d'empreintes digitales à l'aide des réseaux de neurones est basée sur deux étapes principales : l'entraînement ou apprentissage du réseau et le test [83, 92, 93]. Pour cette méthode, le processus de reconnaissance d'une empreinte digitale consiste en deux phases importantes : l'extraction des minuties et la classification du gabarit [44, 119, 149].

Dans la première étape, une image est passée en entrée du système et est réduite à un vecteur de traits caractéristiques, ceci à travers des opérations diverses (transformations, calcul des moments, etc.).

Dans la seconde étape, une classification ou comparaison directe qui aboutira à la reconnaissance ou pas du pattern est effectuée entre le vecteur caractéristique en entrée et ceux des empreintes constituant la base de données. Cette méthode basée sur le réseau de neurones localement connectés est plus appropriée aux applications de vérification de sécurité et d'identification.

La particularité de cette technique est l'utilisation des vecteurs caractéristiques qui sont invariants aussi bien à la translation qu'à la rotation de l'image d'empreinte digitale [149]. Cette méthode se décline en trois étapes principales à savoir : la construction de la base de données, la présentation de la structure du réseau de neurones et enfin la

présentation des algorithmes utilisés pour effectuer la correspondance.

4.1.3.1 Construction de la base de données

Comme mentionné plus haut, il est préférable de construire sa propre base de données. Cependant, il existe certaines bases de données dont la pertinence est reconnue mondialement : c'est le cas de la base de données *NIST* [163]. Pour cette méthode, *Thomas* [149] recommande d'utiliser la base de données *NIST* et une technique de classification basée sur deux éléments simples : *la distance euclidienne* et *les k plus proches voisins*. Dans chacun des deux cas la classe correcte de l'image à reconnaître est connue à l'avance pour effectuer l'apprentissage [93].

La base de données *NIST* contient 2000 paires d'images d'empreintes digitales au format 512×512 et utilise 8 bits pour représenter 1 pixel. Ces images ont été produites par scannage des empreintes digitales d'individus avec un scanner CCD (voir section 1.3.2).

4.1.3.2 Structure du réseau de neurones

Généralement, le réseau de neurones connectés localement exécute un mappage d'un pattern vers un autre de dimension inférieure dans lequel la translation et la rotation sont invariantes : c'est *la représentation spatiale de l'image*. Le vecteur des traits caractéristiques résultant de ce mappage prend en compte les caractéristiques locales, globales et relationnelles du pattern en entrée, spécifiquement comme étant une fonction de densité des pixels. Une fois que cette représentation spatiale est effectuée, une technique de classification standard peut être utilisée pour réaliser la reconnaissance. Bien que ce type d'algorithme s'implémente aisément sur plusieurs architectures, il fournit des résultats très intéressants lorsqu'il est programmé directement sur le matériel comme un système embarqué [51, 81, 149, 166]. La structure générale du réseau précédemment décrit peut être définie à la figure 4.4 ci-après.

i) Pré-traitement

Le pattern en entrée peut être stocké numériquement à la position de chaque pixel ou peut être capté directement sous forme de pixel par l'utilisation d'un scanner numérique. Généralement, le pattern en entrée constitue l'entrée d'alimentation du neurone¹ connecté de façon spatiale avec les pixels en entrée.

1. Le neurone dont on fait allusion ici n'est pas pris comme étant un neurone formel tel qu'indiqué dans la littérature, mais plutôt comme un ensemble de pixels constituant un pattern.

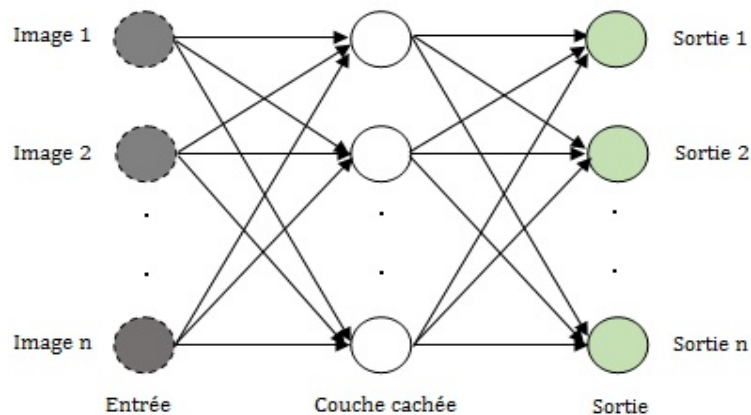


FIGURE 4.4 – Réseau de neurones connectés localement.

Le réseau utilise alors les pixels de l'image en entrée comme l'entrée d'alimentation, tandis que les connexions locales du voisinage fournissent les connexions d'entrée. Le réseau de neurones entier concourt pour une valeur fixe appelée *l'énergie globale*, qui est distribuée en fonction du temps à travers le réseau comme une fonction de la position spatiale relative des entrées d'alimentation et de connexion. Chaque neurone ajoute son entrée d'alimentation et de connexion, puis calcule le ratio de son total à celui du total de tous les autres neurones dans le réseau pour déterminer sa contribution au calcul de *l'énergie globale*, qui devient son nouveau degré d'activation [149]. Ce processus est répété jusqu'à ce que le degré d'activation de chaque neurone se positionne à une valeur fixe.

L'activation d'un neurone à une position spatiale est donc représentée par une fonction prenant en compte aussi bien l'entrée d'alimentation du pixel correspondant que les poids de connexion de ses pixels voisins.

Les positions sans aucune entrée d'alimentation subissent l'influence de leurs pixels voisins et obtiennent une activation non nulle. Afin d'éviter une valeur non désirée, un seuil est fixé dans le calcul de l'énergie globale du réseau. De plus, les blocs de pixels voisins qui s'apparentent seront réduits en un seul bloc [100, 46, 73] afin d'éliminer les informations superflues. La figure 4.5 illustre la structure d'un neurone d'entrée avec ces connexions.

Puisque le réseau est récurrent et représenté par une équation dynamique, son activation sera faite de façon itérative. La mise à jour d'un simple neurone sera effectuée comme suit :

1. Calculer la somme pondérée du voisinage local d'un neurone relié à ses voisins par des poids de connexion, afin de le rendre invariant à la translation et à la rotation ;
2. Ajouter la valeur du pixel d'alimentation en entrée à la somme pondérée obtenue

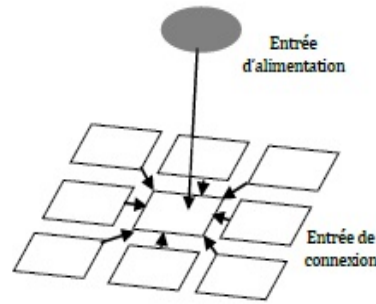


FIGURE 4.5 – Neurones d'entrée et de connexion.

à l'étape 1 précédente pour former *la somme locale*.

3. Une fois que les étapes 1 et 2 ont été effectuées pour le réseau de neurones entier, revenir et diviser *la somme locale* par toutes les sommes obtenues après chaque opération sur un neurone, et multiplier le résultat obtenu par l'énergie globale pour obtenir la nouvelle activation locale.

La somme locale est donc calculée pour chaque neurone à travers le réseau de neurones entier et le ratio entre ces sommes ; le total de ces sommes est utilisé pour déterminer la part locale de l'énergie totale fixe. Chaque neurone obtient ainsi un pourcentage sur l'énergie totale basée sur la valeur obtenue du ratio entre sa somme et la somme totale calculée. Mathématiquement l'activation d'un neurone dans le réseau est définie par :

$$\alpha_{ij}(n+1) = \frac{I_{ij} + \sum_{kl \in N_r(ij)} [w_{ij;kl} \alpha_{kl}(n)]}{\sum_{ij} (I_{ij} + \sum_{kl \in N_r(ij)} [w_{ij;kl} \alpha_{kl}(n)])} \cdot E \quad (4.10)$$

Où :

- $kl \in N_r(ij)$ sont les coordonnées kl d'un point dans un rayon r de voisinage d'un neurone ij ;
- $\alpha_{kl}(n)$ le niveau d'activation courant du neurone kl dans $N_r(ij)$;
- $w_{ij;kl}$ les poids de connexions entre le neurone ij et le neurone kl ;
- I_{ij} la valeur du pixel de l'image en entrée à la position ij ;
- E l'énergie globale du réseau ;
- et n le nombre d'itération.

ii) *Extraction des traits caractéristiques*

Une fois que le réseau de neurones est établi, chaque neurone a un degré d'activation fixe. Un vecteur caractéristique est alors extrait du réseau en calculant un histogramme

qui s'élève au nombre total de neurones dans chaque tranche d'ensemble d'activation [149]. Le nombre de tranches dans l'histogramme est un paramètre ajustable dépendant de l'application et détermine directement la taille du vecteur caractéristique en sortie. Généralement s'il y a beaucoup de tranches dans le graphe, alors l'histogramme est sensible à des légères variations dans l'image d'entrée. Par contre s'il y a peu de tranches dans le graphe, l'histogramme sera incapable de bien faire la différence entre les images en entrée. De plus, les degrés d'activation minimum et maximum sont aussi des paramètres de l'histogramme qui peuvent être déterminés en fonction du type de pattern en entrée. Par exemple, des motifs de petites tailles avec des larges arrières plans (tels que les caractères manuscrits) auront beaucoup de neurones avec peu de tranches d'histogrammes qui ne représentent pas assez d'informations sur le motif [96]. Dans ce cas, le traitement sera effectué uniquement sur les tranches de l'histogramme qui sont au dessus d'un certain seuil. Inversement, pour des motifs de grandes tailles (tels les empreintes digitales) qui ont une distribution qui s'apparente à la distribution gaussienne, on aura besoin de la totalité des tranches d'activation pour effectuer le traitement.

Chaque degré d'activation du neurone appartient ainsi à une tranche (ou à ensemble de tranches pour rendre le graphe continu). Le vecteur caractéristique est formé avec les mêmes dimensions que le nombre de tranches de l'histogramme et, la valeur dans chaque dimension est égale au nombre de neurones qui s'alignent au degré d'activation de la tranche. Pour compléter le système de classification, une variété de techniques (voisin les plus proches, classe de moyennes, etc.) sera utilisée pour effectuer la classification finale du vecteur caractéristique généré ci-dessus pour la reconnaissance du motif en entrée.

Le processus de création d'un histogramme à partir du réseau d'activation détruit effectivement les informations spatiales dans le motif original ; c'est ce phénomène qui produit l'invariance à la translation et à la rotation. Le réseau de neurones préserve les informations relatives à la relation existante entre les motifs, car l'activation locale est une fonction de la relation spatiale entre un pixel et ses voisins. Les vecteurs caractéristiques produits par le réseau de neurones localement connecté des images d'empreintes digitales sont obtenus des calculs effectués à partir de l'équation 4.10 ci-dessus. Une fois que les opérations de pré-traitement ont été effectuées, les algorithmes de correspondance peuvent alors être appliqués.

4.1.3.3 Présentation des algorithmes de correspondance

La méthode de reconnaissance d'empreintes digitales basée sur les réseaux de neurones localement connectés utilise trois principaux algorithmes pour effectuer la correspondance entre deux gabarits lors de la phase d'apprentissage du réseau ; il s'agit des algorithmes suivants : la correspondance à l'aide des classes de moyennes, la correspon-

dance à l'aide du plus proche voisin et la correspondance à l'aide des k – plus proches voisins.

i) *Algorithme de correspondance à l'aide des classes de moyennes*

Cette technique est très simple, une classe moyenne est calculée à partir des données d'apprentissage pour chaque image d'empreinte digitale utilisée [149]. Une fois que les classes moyennes sont établies, on calcule successivement les distances euclidiennes entre la donnée en entrée et celles de ces classes moyennes là ; c'est la classe moyenne qui obtient la plus petite distance euclidienne qui est déclarée vainqueur et se voit attribuée l'image en entrée.

Cette technique n'est pas très prometteuse, puis-qu'étant basée sur le calcul des moyennes simples des blocs de l'image en entrée. En effet, après les opérations de pré-traitement (qui semblent être orienté sur les blocs de pixels voisins et qui s'appuie principalement sur la texture centrale de l'image) le calcul d'une moyenne simple d'un bloc de pixel peut ne plus donner le même résultat. De plus, pour une même image d'empreinte digitale provenant des capteurs différents, les valeurs des pixels au sein de la texture centrale de l'image peuvent être différentes. D'où la légèreté de cette technique basée sur les classes moyennes.

ii) *Algorithme de correspondance à l'aide du plus proche voisin*

Pour le cas des plus proches voisins ($1 - ppv$), chaque échantillon d'apprentissage est considéré comme un prototype pour sa classe. Le vecteur à classer est comparé à l'ensemble des vecteurs déjà classés ; ceci aussi bien pour la phase d'apprentissage que pour celle du test. Le vecteur à classer est alors attribué à la classe de la forme la plus proche de lui (c'est à dire au vecteur qui obtient la plus petite distance euclidienne de lui).

Cette technique semble être un peu plus rodée que celle précédemment décrite. Car en effet, le test de correspondance n'est plus effectué sur la base du calcul des moyennes simples entre les pixels de l'image, mais en comparant l'image en entrée à des prototypes établis pour chaque classe de données. Seulement chaque échantillon est considéré comme étant un prototype pour sa classe, ce qui veut dire que la comparaison d'un pattern donné nécessite que l'on parcourt tous les éléments de la base de données, cela implique un coût élevé en terme de temps de calcul surtout lorsque l'on aura à faire à une grande base de données.

iii) *Algorithme de correspondance à l'aide des k-plus proches voisins (k-ppv)*

Cette méthode de correspondance utilise k des prototypes les plus proches du vecteur en entrée pour former l'ensemble de comparaison. On affecte au vecteur d'entrée la classe la plus représentée parmi les k -plus proches de lui. Si la classe correcte du vecteur en entrée est connue à priori et que l'on souhaite faire une simple vérification, alors on peut combiner la méthode du *plus proche voisin* à celle des *k-plus proches voisins* pour améliorer les performances de reconnaissance. Dans cet arrangement, si le vecteur de test est proche de la classe correcte, alors la correspondance s'effectue simplement en utilisant la méthode du $1 - ppv$. Si par contre le prototype le plus proche est dans une classe différente, alors on utilise plutôt la méthode des $k - ppv$ pour déterminer si la majorité des k plus proches prototypes est dans la classe correcte. Si c'est le cas, alors l'empreinte est correctement vérifiée, autrement on génère en sortie un faux rejet. Le tableau 4.1 ci-après permet de faire une brève comparaison des différents algorithmes présentés ci-dessus (pour plus de détails, voir section 4.4.2).

Classe moyenne	1-ppv	k-ppv
Calcul des moyennes des pixels d'image	Utilisation d'un prototype pour chacune des données	Utilisation d'un prototype pour chaque classe de données
Temps de calcul évalués en fonction de la taille des images à comparer	Temps de calcul évalués en fonction de la taille des pixels et de leur voisin	Temps de calcul évalués en fonction de la taille des pixels de leurs $k - plus$ proches voisins
Deux images peuvent avoir des valeurs de pixels très proches et ne rien avoir en commun	Le voisinage d'un pixel est pris en compte dans la reconnaissance	Les voisinages de plusieurs pixels sont pris en compte dans la reconnaissance

TABLE 4.1 – Tableau de comparaison des méthodes de correspondance.

Bilan

On a montré que le réseau de neurones localement connectés produit des vecteurs caractéristiques utiles pour la correspondance d'empreintes digitales. Ceci est avantageux pour les performances du système tant en termes de taux de faux rejet que celui de reconnaissance. L'utilisation d'une grande base de données est bénéfique pour cette technique car elle permet de s'assurer que la représentation spatiale ne sera pas trop cohue au fur et à mesure qu'une nouvelle empreinte sera ajoutée à l'ensemble d'apprentissage ou de test. Puisque le réseau de neurones localement connectés a été appliqué à des

sous-images d'empreintes digitales, il est évident que cette technique pourrait bien marcher avec un système qui identifie un ordre de tranches d'images d'empreintes digitales par opposition à une grande empreinte digitale unique [149]. L'utilisation des sous blocs d'images permet d'accroître la sécurité et de réduire le taux de fausses acceptations. Le principal avantage de cette méthode basée sur le réseau de neurones localement connectés est la simplicité de son architecture et la possibilité de réduire les taux d'erreur grâce à son principe de non dépendance à la translation/rotation de l'image.

4.1.3.4 Performances de la méthode basée sur le réseau localement connecté

Ici, on utilise des réseaux de neurones localement connectés pour effectuer la reconnaissance entre un vecteur caractéristique et un ensemble de vecteurs déjà classifiés.

Avantages

- Calcul de l'énergie globale du réseau en fonction de l'ensemble des pixels d'images d'apprentissage.
- Utilisation des prototypes (vecteurs caractéristiques des images d'empreintes suivants les classes prédéfinies) invariants à la translation/rotation pour effectuer la reconnaissance.
- Utilisation des sous-blocs favorables aux filtrages.
- Utilisation d'algorithmes spécifiques (classes moyennes, plus proches voisins) pour effectuer la correspondance.
- L'algorithme est approprié pour l'identification d'un individu dans un grand ensemble de données.

Limites

- Les opérations de pré-traitement sont coûteux en temps de calcul, ce qui contribue à avoir des temps d'apprentissage élevés.
- Utilisation des seuils et des calculs d'activation pour effectuer la correspondance.

4.3 Présentation de la méthode proposée

Les techniques précédemment décrites (réseaux de neurones probabilistes et réseaux de neurones connectés localement) proposent des caractéristiques et des utilisations variées. En prenant en compte les éléments mentionnés par chacune d'elle, une méthode mixte est proposée, qui exploite à la fois les avantages de la méthode probabiliste et celles de la méthode basée sur les plus proches voisins du réseau de neurones connectés

localement. Ainsi, nous proposons un nouvel algorithme avec la structure appropriée qui se décline en quatre (04) étapes comme suit :

Etape 1 : valeurs de caractéristiques essentielles

Premièrement, l'algorithme procède à la détection des valeurs de caractéristiques essentielles qui consistent à extraire une région centrale de chaque image d'empreinte digitale de chacune des bases de données (NIST, BDAL et SOCOFinger) ; après la constitution du gabarit de chacune des images d'empreintes, les prototypes de chaque classe de données sont élaborés en fonction des différentes classes standards de la littérature (boucles à gauche, boucle à droite, tourbillon, arche et arche tentée) ;

Justification de l'étape 1 :

Étant donné que le système travaille directement sur les images d'empreintes digitales, il est primordial qu'il extrait les caractéristiques essentielles de ces dernières (minuties, orientation des stries, etc), afin de disposer des modèles génératifs pour mieux prédire.

Etape 2 : définition de la structure du réseau

Après la détection des valeurs caractéristiques essentielles, nous procédons à la définition de la structure du réseau de neurones utilisé pour effectuer l'authentification (en réalité, la première étape décrite ci-dessus fait partie intégrante de cette deuxième). Ce réseau a une nouvelle structure avec une couche d'entrée, une couche de sortie et une couche cachée constituée de 3 sous couches (deux sous couches de classification et une sous couche de correspondance dont la taille est strictement supérieure à celle des deux précédentes), comme illustré à la figure 4.6 ci-dessous.

La particularité de la structure proposée est qu'à la couche cachée, les sorties du niveau supérieur constituent les entrées du niveau inférieur et ainsi de suite, et des convolutions sont appliqués pour obtenir le résultat à la couche du niveau inférieur. Les deux sous couches de classification sont initialisées avec les prototypes de chaque classe de données ; tandis que la sous couche de correspondance est initialisée avec l'ensemble des gabarits (de chaque classe de données standard) traités de façon parcimonieuse pour les rendre invariants à la translation/rotation (originalité de la structure proposée) [95].

Justification de l'étape 2 :

Yann Le Cun a démontré dans son article [103] que lors de l'apprentissage profond d'un réseau de neurones, plus le nombre de couche cachée est grand, mieux le résultat sera, mais en contrepartie la complexité des calculs est accrue. Il est donc intéressant de trou-

ver un bon compromis en fonction du problème qu'on cherche à résoudre. Dans le cadre de notre travail qui consiste en l'authentification, trois (03) grandes opérations sont en générale nécessaire. Il s'agit de :

- l'identification des minuties et leurs orientations relatives ;
- la constitution des vecteurs caractéristiques en fonction du type de minuties et de l'orientation de celles-ci et enfin ;
- la correspondance des gabarits constitués.

Pourquoi 3 couches cachées ?

Trois (03) couches cachées ont été choisies en raison de ces trois grandes opérations. Chacune de ces trois opérations sera effectuée à chacune des couches cachées et les sorties/résultats des couches supérieures seront utilisées comme entrées des couches suivantes, ceci de façon hiérarchisée. Par ailleurs, il faut en moyenne 12 à 30 minuties pour que la correspondance entre deux gabarits soit établie [28, 58]. Cette contrainte a également orienté dans le choix des 03 couches cachées de la manière suivante : les prototypes de la 1^{re} couche cachée utilisent les 12 premières minuties détectées, ceux de la 2nd couche cachée utilisent les autres minuties restantes et l'ensemble est reconstitué à la couche de correspondance (chose qu'aucune technique n'avait encore exploré jusqu'ici) [94, 95].

Etape 3 : classification des prototypes et correspondance des gabarits

- Une fois la structure du réseau élaborée, nous procédons à la classification proprement dite de l'image en entrée du réseau par comparaison de gabarit avec ceux des prototypes des différentes classes de données, ceci par simples calculs de probabilité. A la première couche de classification, les vecteurs caractéristiques voués aux calculs de comparaison sont établis en prenant en compte uniquement les minuties. Et à la deuxième couche de classification, ces derniers sont élaborés en prenant en compte l'intensité des pixels des images prototypes de chaque classe de données.
- Enfin, le gabarit vainqueur des deux classifications précédentes est comparé à l'ensemble des données de cette classe qui est constituée de vecteurs caractéristiques invariants à la translation/rotation. Cette dernière comparaison permet de réduire les faux rejets et de mettre à jour les prototypes afin de renforcer les éléments de la classe correcte. Lorsque le vainqueur n'est pas de la classe correcte, il est rejeté et les erreurs sont rétro propagées.

Justification de l'étape 3 :

L'utilisation des prototypes plutôt que l'ensemble des données de la base pour effectuer l'authentification permet de réduire les temps de calcul. A chaque fois qu'un prototype est reconnu appartenant à la classe correcte, il est modifié en fonction des éléments de l'image reconnue.

Cette étape permet de classifier l'ensemble des données par classe de données standard définies dans la littérature et facilitera la correspondance des gabarits.

Etape 4 : Prédiction ou reconnaissance

Afin de se rassurer de l'efficacité de l'algorithme proposée par rapport à ceux existants dans la littérature, des tests sont effectués sur trois bases de données : une base de données de renommée internationale *NIST4*, une base de données construite localement *BDAL* (voir 4.3.2) et une base de données construite avec des étudiants africains *SOCOFinger*.

4.3.1 *Structure du réseau de neurones proposée*

La structure du réseau de neurones proposée sera alors celle d'un réseau de neurones localement connectés en profondeur, qui utilise le calcul des probabilités pour classer les sous-blocs de l'image en entrée en tenant compte de la classification d'*Henry* [74]. Elle utilise également un seuil (pour déterminer s'il y a matching entre deux gabarits) et l'algorithme de rétro-propagation de base pour l'entraînement du réseau de neurones.

Le réseau de neurones utilisé dispose alors d'une couche d'entrée, d'une couche cachée contenant elle-même trois couches (deux couches de classification et une couche de correspondance) et enfin d'une couche de sortie (confère Figure 4.6 ci-dessous).

Comme pour les méthodes précédentes, le processus de construction du système de reconnaissance se déroule en deux étapes : l'apprentissage et le test. Pendant la phase d'apprentissage, on utilise un échantillon d'images d'empreintes pour entraîner le réseau de neurones à effectuer la reconnaissance. La phase de test quant à elle implique uniquement des images n'ayant pas été utilisées lors de l'entraînement. La structure générale du réseau de neurones peut donc être observée à la figure 4.6 ci-après.

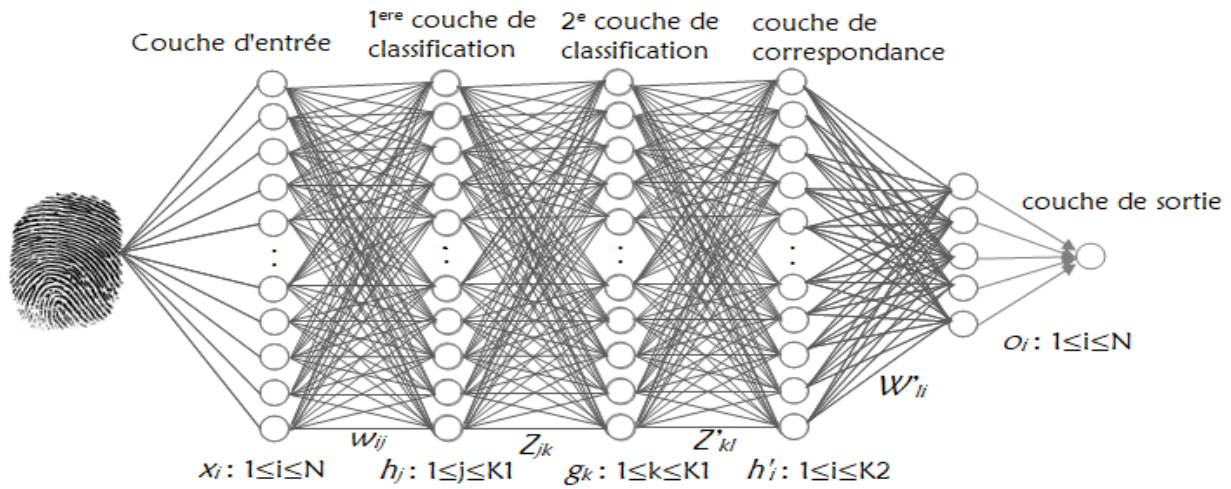


FIGURE 4.6 – Structure du réseau de neurones proposé.

Où :

- x_i représente le vecteur de l'image en entrée ;
- h_j les prototypes de chaque classe (en fonction de la classification d'Henry [74]) de données utilisée pour l'apprentissage ;
- w_{ij} les poids de connexion entre la couche d'entrée et la première couche de classification ;
- z_{jk} les poids de connexion entre la première et la seconde couche de classification ;
- g_k les données de la seconde couche de classification représentées par classe de données ;
- z'_{kl} les poids de connexion entre la seconde couche de classification et la couche de correspondance ;
- h'_i l'ensemble des données de la base constituant la couche de correspondance ;
- w'_{li} les poids de connexion entre la couche de correspondance et la couche de sortie ;
- et o_i est la sortie attendue.

4.3.2 Construction de la base de données

Comme cela a déjà été mentionné dans les sections précédentes, il existe plusieurs bases de données internationales d'empreintes digitales. Seulement la plupart de ces bases de données sont généralement constituées d'une seule occurrence de chaque image d'empreinte digitale, alors que le système a parfois besoin de prendre en compte plusieurs images d'une même empreinte sous différentes positions afin de mieux s'adapter aux données à traiter lors de la phase d'apprentissage. C'est la raison pour laquelle le choix

d'un échantillonnage constitué de trois bases de données est fait :

- une base de données (appelée BDAL) constituée de 500 paires d'images d'empreintes digitales au format 248×338 avec 8 – *bits* par pixel, provenant de 100 personnes différentes (toutes étant entraînées, coopératives et surtout sensibilisées sur l'importance du processus engagé ; ce qui a permis d'éviter d'avoir des images latentes et très bruitées) à raison de 5 images d'empreintes par personne (c'est la même empreinte digitale qui est prise en compte) ;
- la base de données internationale NIST 4 [163] qui est une base de donnée spécifique à la reconnaissance par empreinte digitale ; elle est constituée de 2000 paires d'images d'empreintes digitales au format 512×512 avec 8 – *bits* par pixel.
- et la base de données *SOCOFinger* construite en collaboration avec des étudiants africains du laboratoire "Data Science and Big data development", constituée de 6000 images d'empreintes digitales au format divers avec 8 – *bits* par pixel, provenant de 1200 individus. Cette base de données a fait l'objet d'une publication internationale et est ouverte au grand public depuis 2018.

Les images de ces bases de données sont regroupées avec le souci permanent d'assurer une couverture adéquate dans la variation des 5 classes définies par *Henry* dans son article [74], de la manière suivante : pour la base de données BDAL 100 images par classe, pour la base de NIST 400 images par classe et pour *SOCOFinger* 600 images par classe de données. La figure 4.7 montre quelques images provenant desdites bases de données.



FIGURE 4.7 – Quelques exemples d'images d'apprentissage

4.3.3 Apprentissage

Une fois que le réseau de neurones est construit et que la base de données a été créée, l'apprentissage supervisé du réseau de neurones profond peut être effectué à l'aide de l'algorithme de rétro-propagation.

Raffinement

L'image en entrée, obtenue sous la forme numérique subit quelques opérations afin d'éliminer les informations superflues. Le but premier de ces opérations est de construire un vecteur contenant les informations sur le texture centrale de l'image d'empreinte digitale et de calculer l'orientation relative des pixels au sein de cette image.

Après ces opérations de raffinement de l'image d'empreinte, on procède à la détection des minuties à l'aide de la méthode du *crossing number* décrit à la section 2.2.2 ci-dessus. Une fois le type de minutie établi, on calcule leurs orientations relatives au sein de cette image en utilisant *l'algorithme de l'index de Poincaré* décrit à la section 2.2.1.2.

Enfin, on construit le vecteur caractéristique de la forme : $(type\ de\ minutie, x, y, \theta)$, qui est invariant à la translation et la rotation. Où *type de minutie* représente le type de la minutie du pixel se trouvant à la position (x, y) et θ représente l'angle d'orientation de cette minutie.

Ensuite on applique l'algorithme proposé de la manière suivante :

Algorithme proposé

1. Choisir les prototypes initiaux deux par deux pour chaque classe de données, ceci en fonction de la classification d'Henri.
2. Initialiser les neurones de la première couche de classification par les premiers prototypes de chaque classe de données (en fonction des 12 premières minuties).
3. Initialiser les neurones de la seconde couche de classification par les seconds prototypes de chaque classe de données (en fonction des autres minuties restantes).
4. Initialiser les neurones de la couche de correspondance par les données de la base en fonction des classes prédéfinies.
5. Appliquer un vecteur en entrée.
6. Appliquer un filtre de convolution à ce vecteur et obtenir les features map.
7. Appliquer la mesure de similarité à la première couche de classification.

$$h_j = H(P(M(x_i, w_{ij}))) \quad 1 \leq i \leq N, \quad 1 \leq j \leq K1,$$

8. Sélectionner les features vainqueurs.

$$\text{Softmax}(h_j), \quad 1 \leq j \leq K1$$

9. Appliquer la mesure de similarité à la seconde couche de convolution.

$$g_k = \sqrt{\sum_{j=1}^{K1} (h_j - Z_{jk})^2} \quad 1 \leq k \leq K1,$$

10. Sélectionner les features vainqueurs.

$$\text{Softmax}(g_k), \quad 1 \leq k \leq K1$$

11. Appliquer la mesure de similarité à la couche de correspondance (si le vainqueur est élément de la classe correct).

$$h'_l = \sqrt{\sum_{k=1}^{K1} (g_k - z'_{kl})^2} \quad 1 \leq l \leq K2,$$

12. Sélectionner les features vainqueurs.

$$\text{Softmax}(h'_l), \quad 1 \leq l \leq K2$$

13. Calcul de l'erreur entre la sortie attendue et celle obtenue et vérifier les seuils.

$$o_i = \frac{1}{N} \sum_{i=1}^N (h'_l - w'_i)^2 \quad 1 \leq l \leq K2,$$

14. Modifier les poids de connexion des vainqueurs par :

$$\Delta w_{ij}^l = \alpha(x_k^l - w_{ij}^l),$$

15. Mettre à jour les prototypes des neurones vainqueurs par :

$$m_{n+1}^l = \frac{m_n^l + x_{n+1}^l}{n + 1}$$

16. Retour à l'étape (5) jusqu'à ensemble de données terminé.

Fin Algorithme

Dictionnaire de données

- x_i représente le vecteur en entrée ;
- w_{ij} les poids de connexion entre x_i et le premier prototype vainqueur h_j ;
- $P(M(x_i, h_j))$ la probabilité que l'image en entrée soit égale à celle de la première couche de classification ;
- z'_{kl} les poids de connexion entre le premier prototype vainqueur et le second prototype vainqueur g_k ;
- w'_{ij} les poids de connexion entre l'ensemble des vecteurs h'_i de la classe vainqueur après les deux classification et la sortie attendue o_j ;
- w_{ij}^l poids de connexion entre le neurone i et le neurone j de la couche l ;
- x_k^l neurone k en entrée de la couche l ;
- m_{n+1}^l le prototype mis à jour à la couche l ;
- x_{n+1}^l est le nouveau vecteur en entrée de la couche l qui servira à mettre à jour le vecteur prototype de ladite couche.

Remarques

- Les prototypes de chaque classe sont des vecteurs qui sont invariants à la translation et la rotation et sont sous la forme (*type de minutie*, x , y , θ).
- Le représentant initial ou prototype peut être choisi de façon aléatoire entre les vecteurs d'une classe, mais l'algorithme aura plus de chance de s'exécuter efficacement si on utilise plutôt un vecteur représentatif de cette classe [93].
- Les neurones de la couche de classification sont initialisés en fonction de l'échantillonnage et les prototypes sont formés en calculant la moyenne des vecteurs représentant chaque classe de données.

4.3.4 Utilisation

L'apprentissage a pour but essentiel d'entraîner le réseau de neurones à reconnaître les empreintes à partir d'un ensemble de données préétabli. La phase d'utilisation quant à elle, sert à tester les performances du réseau créé en appliquant des données non encore rencontrées par le réseau.

Le vecteur en entrée à reconnaître est donc passé successivement aux couches de classification et à celle de la correspondance comme illustré aux figures 4.8, 4.9, 4.10 ci-dessous.

En considérant uniquement la couche d'entrée et la première couche de classification, nous avons le sous réseau de neurones présenté à la figure 4.8. Ce réseau a pour objectif final de positionner l'image en entrée en fonction de l'ensemble des classes définies par

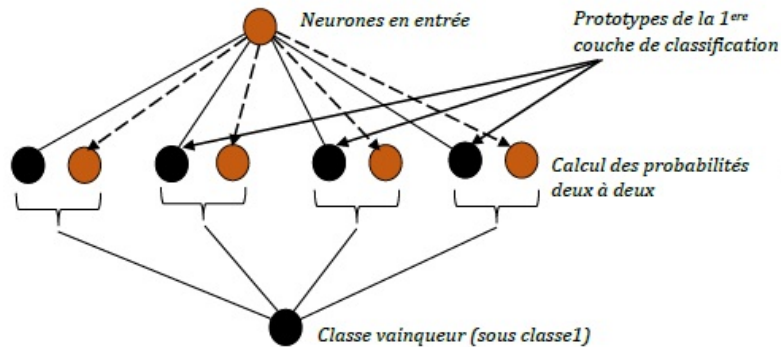


FIGURE 4.8 – Illustration de la première Classification

la classification d'Henry [74]. Les comparaisons sont faites deux à deux entre l'image en entrée et les prototypes de la première couche de classification. Le vainqueur est sélectionné et passé à la deuxième couche de classification pour vérification du résultat obtenu, qui à son tour produit en sortie un neurone vainqueur (confère figure 4.9).

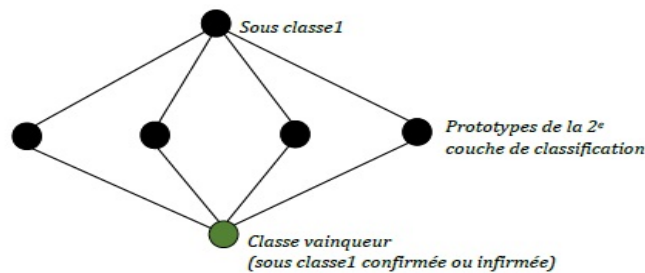


FIGURE 4.9 – Illustration de la deuxième classification

Une fois que la classe de l'image en entrée est connue, on peut alors effectuer la correspondance en sélectionnant uniquement les données appartenant à la classe vainqueur ; ceci, par le calcul d'une simple mesure de similarité entre les vecteurs caractéristiques de l'entrée et ceux de la sortie attendue comme illustré à la figure 4.10.

Si le résultat obtenu après application de la mesure de similarité respecte les conditions d'arrêt fixées par le seuil, alors la correspondance est établie entre les deux gabarits et l'empreinte en entrée est acceptée ; sinon elle est rejetée et enregistrée dans l'ensemble de non correspondance selon que la sortie attendue ne correspond réellement pas avec l'image en entrée ou dans l'ensemble de faux rejets selon que la sortie attendue est en fait une copie de l'image de départ.

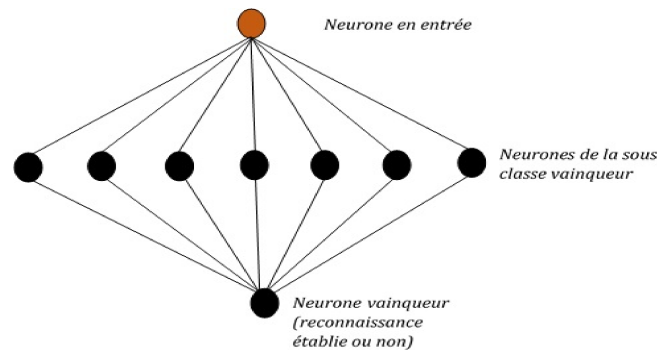


FIGURE 4.10 – Processus d'identification par classe de données

4.4 Résultats expérimentaux et discussions

Cette section expose et discute les résultats de l'implémentation des différents algorithmes décrits précédemment. Mais avant, elle présente le choix et la préparation des échantillons.

4.4.1 Choix et préparation des échantillons

Pour exécuter les différentes étapes (d'apprentissage et de test), les outils ci-après ont été utilisés :

- Un ordinateur portable MacBook Pro (13-inch Mid 2012) CPU 2.5GHz Intel Core i5 Memory 6Go 1600MHz DDR3 ;
- Un ordinateur Intel® Core™ i5 CPU 2.5GHz 2.5GHz et 4Go de RAM ;
- Un ordinateur Intel(R), Core(TM) i3–6006U CPU@2.00GHz 1,99GHz avec 4Go de RAM ;
- Un ordinateur Intel Pentium CPU G645@ 2.90Hz 2.90GHz avec 2Go de RAM ;
- Un ordinateur Intel(R) Core(TM) i3 – 4130T CPU@2.90Ghz 2.90GHz avec 8Go de RAM ;
- Les systèmes d'exploitation Ubuntu 12.10, Windows 8.1 Pro et MacOS High Sierra ;
- Les logiciels Matlab Release 2010a, GNU Octave 3.6.1, Jupyter notebook server : 5.6.0, Python 3.7.0, Anaconda 2.4 ;
- Le compilateur gcc 3.4 avec la librairie Openmpi.

Le protocole de validation utilisé pour l'échantillonnage de la base de données est la "cross-validation" [76], qui stipule qu'un pourcentage de l'échantillon de départ peut être utilisé pour la phase d'apprentissage et le reste consacré à la phase de test/utilisation.

Ainsi, les échantillonnages successifs de 40–60, 50–50 et de 80–20 ont été employés. C'est à dire 40% d'images d'empreintes pour l'apprentissage et 60% pour le test dans le premier cas ; 50% d'images d'empreintes pour l'apprentissage et le test dans le second cas ; 80% pour l'apprentissage et 20% pour le test dans le troisième cas.

Par ailleurs, pour se assurer que notre modèle ne fait pas du sur-apprentissage, la régularisation "*Dropout*" est mise à contribution. Cette technique consiste à désactiver certains neurones à chaque itération pour ne pas que les neurones dans leur globalités soient dépendants les uns des autres. La figure 4.11 suivante illustre ce phénomène :

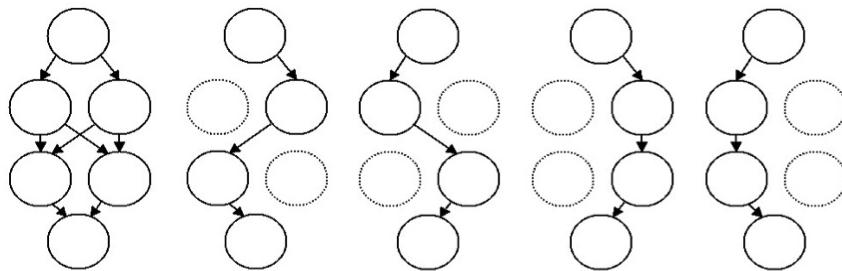


FIGURE 4.11 – Technique du Dropout

Ainsi, un "*Dropout*" est placé à l'entrée de chaque couche cachée. Il sera chargé de désactiver 1% des neurones (il n'est pas en effet recommandé d'aller au delà de 4% de neurones désactivés par risque de faire face à "*Underfitting*").

Après acquisition, les images sont compressées au format 64×64 [92], puis découpées successivement en sous blocs de taille 8×8 et 16×16 afin de réduire le nombre d'opérations à effectuer lors du traitement (ce découpage est effectué par le système lui même après le raffinement de l'image d'empreinte). En identifiant des sous blocs plutôt qu'une image entière, les performances du système seront accrues (le taux de fausse reconnaissance et les temps de calculs seront réduits). Par exemple si le taux de fausse reconnaissance est de 10%, alors la probabilité de reconnaître faussement un bloc de l'image sera de 10%, mais la probabilité de reconnaître faussement deux blocs de l'image sera seulement de 1% (et celle de reconnaître trois blocs sera de 0.1%, etc.) [11, 147].

4.3.1.1 Bases de données

Comme déjà mentionné à la section 4.3.2, trois bases de données sont utilisées pour effectuer les tests : la base de données internationale NIST 4, la base de données BDAL et la base de données SOCOFinger.

Énergie globale des Bases de données

Une analyse des 2000 images centrales de la base de données NIST montre une intensité moyenne de 102 avec un écart type de 26. La valeur minimum de la moyenne d'intensité est de 21 et le maximum est de 197.

D'un autre côté, l'analyse des 500 images de la base de données BDAL fait ressortir une intensité moyenne de 57 avec un écart type de 11. Le bruit gaussien avec un écart-type de 10 a été ajouté à chaque pixel de chaque échantillon afin d'émuler le bruit sonore et plus tard pour augmenter la différence entre les échantillons de la même empreinte digitale.

Enfin, l'analyse des 6000 images de la base de données SOCOFinger fait ressortir une intensité moyenne de 356 avec un écart type de 86.

Une fois que la classification est faite, la phase d'apprentissage à proprement parlé peut être engagée.

4.3.1.2 Protocole de validation et Indicateurs de performances

Le protocole de validation de notre algorithme consiste à comparer les résultats obtenus par notre modèle, à ceux existants à savoir : les Réseaux de neurones probabilistes (Baldi et Maltoni), Réseaux de neurones localement connectés (T.J et Thomas) et les réseaux de neurones convolutionnels (Qian et Yu).

Les performances du système quant à eux, seront évaluées tant par un aspect temporel que par un aspect qui tient compte de la qualité de reconnaissance. On aura donc pour principaux indicateurs de performances [11, 112, 166, 172] :

- le taux de reconnaissance ou d'acceptation :

$$TA = \frac{M}{N},$$

- le taux d'erreur, constitué du : taux de faux rejet (TFR) et du taux de fausse acceptation (TFA)

$$TFR = \frac{R}{N}, \quad TFA = \frac{A}{N}$$

- le temps d'apprentissage,
- le temps de reconnaissance.

Où :

- N est le nombre de caractères présentés au système lors de la phase de test ;
- M le nombre de caractères reconnus ;
- R et A respectivement le nombre de caractères rejetés et acceptés à tort par le système.

Le tableau ci-dessous fait la liaison entre les notions de fausse acceptation et faux rejet à celles de Vrai positif (VP), Vrai négatif (VN), Faux positif (FP) et Faux négatif (FN). Il y a également les indicateurs de performances suivants :

	$\hat{1}$	$\hat{0}$
1	Vrai Positif	Faux Négatif
0	Faux Positif	Vrai Négatif

– Précision

$$Precision = \frac{VP}{VP + FP};$$

– Rappel ou Sensibilité

$$Rappel = Sensibilite = \frac{VP}{VP + FN};$$

– Score

$$Score = \frac{VP + VN}{\sum(VP + FP + VN + FN)}.$$

La phase d'apprentissage a été effectuée successivement avec 40%, 50% et 80% d'images des bases de données BDAL, NIST et SOCOFinger, ceci avec des images découpées successivement en sous-blocs de taille 8×8 et 16×16 . Une exécution de notre algorithme en parallèle à l'aide de la méthode *mpi* est également explorée. La phase de test quant à elle a été effectuée avec le reste d'images des deux bases de données dans chacun des cas sus-évoqués. En guise d'exemple, la figure 4.12 montre quelques images provenant des mêmes personnes qui ont été utilisées pour effectuer la phase de test [163].

4.4.2 Apprentissage avec des sous-blocs de taille 8×8

Dans ce cas, 40%, 50% et 80% de données de chaque base sont successivement utilisés. Soient :

- pour la base de données BDAL : 200, 250 et 400 images également réparties entre les individus de la base de données (excepté pour le cas de 250 images où on a eu à tirer de façon aléatoire 1 image supplémentaire des 50 individus de la base après avoir préalablement sélectionner 2 images par personne).
- pour la base de données NIST : 800, 1000 et 1600 images.
- pour la base de données SOCOFinger : 2400, 3000 et 4800 images également réparties entre les individus de la base de données.

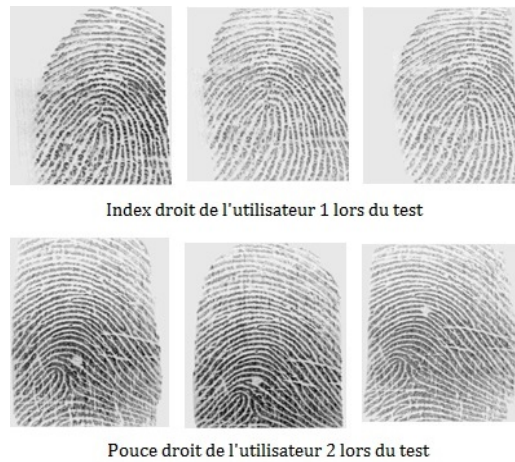


FIGURE 4.12 – Exemples d'images de test provenant du même individu

4.3.2.1 Apprentissage avec la base de données NIST 4

Les résultats sont consignés dans le tableau 4.2 ci-dessous.

4.3.2.2 Apprentissage avec la base de données BDAL

Le tableau 4.4 ci-dessous fait état des résultats obtenus.

4.3.2.2 Apprentissage avec la base de données SOCOFinger

Le tableau 4.4 ci-dessous fait état des résultats obtenus.

4.4.3 Apprentissage avec des sous-blocs de taille 16×16

L'échantillonnage est le même que celui précédemment décrit, seulement ici, les images sont découpées en blocs de 16×16 .

4.3.3.1 Apprentissage avec la base de données NIST 4

Les résultats sont consignés dans le tableau 4.5 ci-dessous.

4.3.3.2 Apprentissage avec la base de données BDAL

Les résultats sont consignés dans le tableau 4.7 ci-dessous.

4.3.3.2 Apprentissage avec la base de données SOCOFinger

Les résultats sont consignés dans le tableau 4.7 ci-dessous.

Apprentissage avec 40% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	33min	24min	27min	31min	39min	11min
Temps de reconnaissance	41s	5s	4s	5s	9s	2s	2s
TA (en %)	93.96	96.18	51.99	89.93	96.27	96.76	96.91
TFR (en %)	5.66	3.35	15.22	9.64	3.32	2.83	2.77
TFA (en %)	0.38	0.47	32.79	0.43	0.41	0.41	0.32
Apprentissage avec 50% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe de moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	41min	30min	34min	39min	47min	14min
Temps de reconnaissance	42s	5s	4s	5s	9s	2s	2s
TA (en %)	94.14	96.44	52.93	91.58	96.57	96.93	97.27
TFR (en %)	5.54	3.23	15.78	8.11	3.11	2.72	2.46
TFA (en %)	0.36	0.33	31.29	0.31	0.32	0.35	0.27
Apprentissage avec 80% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe de moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	66min	48min	54min	62min	78min	21min
Temps de reconnaissance	45s	5s	4s	5s	10s	2s	3s
TA (en %)	94.31	96.64	53.53	92.42	96.74	97.13	97.44
TFR (en %)	5.34	2.92	15.26	7.18	2.96	2.55	2.37
TFA (en %)	0.35	0.44	31.21	0.40	0.30	0.32	0.19

TABLE 4.2 – Résultats de l'apprentissage avec les blocs de taille 8×8

Apprentissage avec 40% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	23min	14min	17min	21min	19min	11min
Temps de reconnaissance	16s	3s	2s	3s	5s	1s	1s
TA (en %)	92.92	95.14	50.95	88.89	95.23	95.72	95.87
TFR (en %)	6.69	4.38	16.25	10.67	4.35	3.86	3.80
TFA (en %)	0.39	0.48	32.80	0.44	0.42	0.42	0.33
Apprentissage avec 50% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe de moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	23min	14min	17min	21min	19min	11min
Temps de reconnaissance	16s	3s	2s	3s	5s	1s	1s
TA (en %)	92.92	95.42	51.91	90.56	95.55	95.91	96.25
TFR (en %)	6.60	4.29	16.84	9.17	4.17	3.77	3.53
TFA (en %)	0.28	0.29	31.25	0.27	0.28	0.32	0.22
Apprentissage avec 80% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe de moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	28min	19min	22min	25min	27min	16min
Temps de reconnaissance	19s	3s	2s	3s	5s	1s	1s
TA (en %)	93.28	95.60	52.44	91.40	95.60	96.09	96.38
TFR (en %)	6.38	.96	16.30	8.22	4.03	3.59	3.41
TFA (en %)	0.34	0.44	31.26	0.38	0.37	0.32	0.21

TABLE 4.3 – Résultats de l'apprentissage avec les blocs de taille 8×8

Apprentissage avec 40% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	39min	30min	33min	37min	45min	16min
Temps de reconnaissance	39s	5s	4s	5s	9s	2s	3s
TA (en %)	94.96	97.18	52.99	90.93	97.27	97.76	97.91
TFR (en %)	4.64	2.33	14.20	8.62	2.30	1.81	1.75
TFA (en %)	0.40	0.49	32.81	0.45	0.43	0.43	0.34
Apprentissage avec 50% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe de moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	46min	35min	39min	44min	52min	26min
Temps de reconnaissance	40s	5s	4s	5s	9s	2s	3s
TA (en %)	95.26	97.56	54.05	92.70	97.69	98.05	98.39
TFR (en %)	4.41	2.10	14.65	6.98	1.98	1.59	1.33
TFA (en %)	0.33	0.34	31.30	0.32	0.33	0.36	0.28
Apprentissage avec 80% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe de moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	69min	51min	57min	65min	81min	38min
Temps de reconnaissance	45s	5s	4s	5s	10s	2s	3s
TA (en %)	95.74	98.07	54.96	93.85	98.17	98.56	98.87
TFR (en %)	3.93	1.51	13.85	5.77	1.55	1.15	0.96
TFA (en %)	0.33	0.42	31.19	0.38	0.28	0.29	0.17

TABLE 4.4 – Résultats de l'apprentissage avec les blocs de taille 8×8

Apprentissage avec 40% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps appren.	Aucun	23min	14min	23min	24min	31min	8min
Temps reconn.	30s	5s	4s	5s	9s	3s	2s
TA (en %)	93.26	95.74	51.29	89.23	95.56	96.06	96.21
TFR (en %)	6.36	4.05	15.92	10.34	4.02	3.53	3.47
TFA (en %)	0.38	0.47	32.79	0.43	0.41	0.41	0.32
Apprentissage avec 50% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps appren.	Aucun	29min	20min	29min	30min	37min	11min
Temps reconn.	31s	5s	4s	5s	9s	3s	2s
TA (en %)	93.44	95.74	52.23	90.88	95.87	96.23	96.57
TFR (en %)	6.24	3.93	16.48	8.81	3.81	3.42	3.16
TFA (en %)	0.32	0.33	31.29	0.31	0.32	0.35	0.27
Apprentissage avec 80% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe de moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps appren.	Aucun	53min	37min	42min	51min	64min	21min
Temps reconn.	32s	5s	4s	5s	8s	3s	3s
TA (en %)	93.61	95.94	52.83	91.72	96.04	96.43	96.74
TFR (en %)	6.04	3.62	15.96	7.88	3.66	3.25	3.07
TFA (en %)	0.35	0.44	31.21	0.40	0.30	0.32	0.19

TABLE 4.5 – Résultats de l'apprentissage avec les blocs de 16×16

Apprentissage avec 40% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	16min	7min	10min	14min	12min	4min
Temps de reconnaissance	14s	3s	2s	3s	5s	1s	1s
TA (en %)	92.22	94.44	50.25	88.19	94.53	95.02	95.17
TFR (en %)	7.39	5.08	16.95	11.37	5.05	4.56	4.50
TFA (en %)	0.39	0.48	32.80	0.44	0.42	0.42	0.33
Apprentissage avec 50% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	18min	10min	12min	16min	15min	6min
Temps de reconnaissance	15s	3s	2s	3s	5s	2s	1s
TA (en %)	94.42	94.72	51.21	89.86	94.85	95.21	95.55
TFR (en %)	7.30	4.99	17.54	9.87	4.87	4.47	4.23
TFA (en %)	0.28	0.29	31.25	0.27	0.28	0.32	0.22
Apprentissage avec 80% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	21min	12min	15min	18min	17min	9min
Temps de reconnaissance	16s	3s	2s	3s	5s	2s	1s
TA (en %)	92.58	94.90	51.74	90.70	94.90	95.39	95.68
TFR (en %)	7.08	4.66	17.00	8.92	4.73	4.29	4.11
TFA (en %)	0.34	0.44	31.26	0.38	0.37	0.32	0.21

TABLE 4.6 – Résultats de l'apprentissage avec les blocs de 16×16

Apprentissage avec 40% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	27min	21min	23min	27min	36min	14min
Temps de reconnaissance	26s	5s	4s	5s	7s	2s	2s
TA (en %)	94.26	96.48	52.29	90.23	96.57	97.06	97.69
TFR (en %)	5.11	2.80	15.35	7.68	2.68	2.39	2.03
TFA (en %)	0.33	0.34	31.31	0.32	0.33	0.26	0.28
Apprentissage avec 50% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	36min	25min	29min	34min	42min	16min
Temps de reconnaissance	19s	5s	4s	5s	7s	2s	2s
TA (en %)	94.56	96.86	53.38	92.01	96.99	97.35	97.69
TFR (en %)	5.11	2.80	15.35	7.68	2.68	2.39	2.03
TFA (en %)	0.33	0.34	31.30	0.32	0.33	0.26	0.28
Apprentissage avec 80% de l'échantillon							
	Méthode classique	Méthode probabiliste	Classe moyennes LCNN	1 – PPV LCNN	K – PPV LCNN (K=8)	FVCNN deep $l = 2$	Méthode proposée (en parallèle)
Temps d'apprentissage	Aucun	58min	42min	47min	56min	69min	26min
Temps de reconnaissance	21s	5s	4s	5s	7s	2s	2s
TA (en %)	95.04	97.37	54.26	93.15	97.47	97.86	98.17
TFR (en %)	4.63	2.21	14.55	6.47	2.25	2.05	1.66
TFA (en %)	0.33	0.42	31.19	0.38	0.28	0.09	0.17

TABLE 4.7 – Résultats de l'apprentissage avec les blocs de 16×16

Commentaires

Dans les trois cas retenus pour l'échantillonnage (à savoir : 40-60, 50-50 et 80 -20) les images sont proportionnellement réparties entre les sous-classes définies par Henry (arche, arche tentée, tourbillon, et boucles) ; le réseau apprend généralement sans problème avec les données des bases *NIST 4*, *BDAL* et *SOCOFInger*. La méthode probabiliste utilise un seuil de décision qui influence le résultat final à mesure qu'il est réglé. Dans ce travail il a été ajusté à un seuil convenable (10^{-7}) afin de réduire au maximum les taux de fausses acceptations, puisque l'algorithme sera utilisé spécifiquement pour l'authentification d'individus dans un système automatisé. Les tableaux 4.2, 4.4, 4.5 et 4.7 montrent que :

(1) Le temps d'apprentissage et de reconnaissance diminuent au fur et à mesure que l'échantillon d'apprentissage baisse. Ce qui est prévisible car les données à traiter sont moins importantes dans un ensemble avec un échantillon réduit que dans un autre avec un échantillon plus élevé.

(2) En utilisant des blocs de vecteurs de taille 16×16 par rapport à des blocs de vecteurs de taille 8×8 un gain important de 6.40 minutes en moyenne peut être atteint dans les temps d'apprentissage. Mais en contre partie, on enregistre une baisse moyenne de 0.6% et une légère hausse de 0.2% respectivement sur le taux d'acceptation et le taux de faux rejet. En effet, lorsqu'on utilise les blocs de taille 8×8 par rapport à ceux de taille 16×16 , la quantité d'informations à gérer est réduite et donc plus facilement traitable en ce qui concerne la détection des stries, des points singuliers, des vallées, ... Cependant, le nombre de sous-blocs à traiter augmente et favorise les temps de traitement élevés.

(3) La méthode probabiliste, le fingerprint verification neural network (FVCNN) et la méthode proposée (qui s'appuie en partie sur la méthode probabiliste) produisent de bons résultats en terme de taux d'acceptation par rapport aux autres techniques (même s'il apparaît évident que c'est la méthode proposée qui produit des meilleurs résultats parce que utilisant des vecteurs invariants à la translation et la rotation). Ceci pourrait s'expliquer entre autres par le fait que la méthode probabiliste et la méthode proposée sont spécifiques pour l'utilisation avec un échantillon restreint de quelques vecteurs caractéristiques et utilisent le calcul des probabilités sur les pixels de ces vecteurs ; tandis que le FVCNN et la méthode proposée s'appuient sur le principe du deep learning.

Discussions

(1) Les images des bases de données (BDAL, NIST et SOCOFinger) utilisées pour effectuer les tests ci-dessus sont proportionnellement réparties entre les sous-classes définies par Henry (arche, arche tentée, tourbillon, et boucles).

(2) L'algorithme de la méthode du réseau de neurones localement connecté basé sur les classes de moyennes des images d'empreintes produit des résultats peu intéressants en terme de taux de reconnaissance. En effet étant donnée qu'ici la mesure de similarité se limite à la simple comparaison des moyennes des pixels des images concernées, on peut trouver plusieurs images ayant des moyennes très proches mais qui n'ont rien en commun en terme de minuties.

(3) Les performances du système en terme de taux de reconnaissance augmentent au fur et à mesure que le nombre de données impliquées dans la phase d'apprentissage augmente. Ce progrès des performances du système est suivi d'une dégradation des temps d'apprentissage et de reconnaissance qui sont également en hausse. En utilisant un échantillonnage de 80% – 20% par rapport à celui de 40% – 60%, un gain moyen de 0.8% peut être atteint dans les taux de reconnaissance. Cependant les temps de calcul restent élevés ; un compromis doit donc être fait en fonction du choix de l'utilisateur.

(4) Par ailleurs, ces mêmes performances (en terme de taux de reconnaissance) diminuent lorsque la taille des blocs d'entrée augmente. En effet, plus la taille des blocs est importante, plus le contenu est varié et donc les chances de trouver des sous-blocs fortement ressemblants s'amointrissent. En utilisant les blocs de pixels de tailles 8×8 par rapport à ceux de tailles 16×16 un profit moyen de 0.6% peut être observé dans les taux de reconnaissance. Cependant on observe une nette augmentation d'au moins 9 minutes en moyenne dans les temps d'apprentissage.

(5) Le FVCNN basée sur le principe de l'apprentissage profond avec 2 couches de convolution offre les meilleurs résultats en terme de taux de fausses acceptations. Ceci pourrait être expliqué par le fait que le réseau utilise la convolution avant d'effectuer la classification.

(6) La méthode proposée basée sur l'utilisation d'un réseau de neurone profond avec trois couches cachées offre les meilleurs résultats en terme de taux de reconnaissance et taux de faux rejets. Ceci peut être expliqué par le fait que le réseau utilise les calculs de probabilités pour effectuer la première classification, ensuite, des vecteurs invariants à la translation/rotation et l'énergie globale des données pour effectuer la correspondance. Cette technique proposée, utilise en effet les avantages des autres techniques offertes par la littérature à savoir le calcul des probabilités avec convolution proposé par Baldi et l'utilisation des vecteurs invariants à la translation/rotation proposé par Thomas.

(7) L'utilisation du parallélisme qui est mis en exergue par la méthode proposée exé-

cutée en parallèle, contribue à réduire les temps d'apprentissage et de reconnaissance. Un bénéfice de presque le quart peut être atteint en exécutant les algorithmes en parallèle. Ce gain pourrait être accru en fonction de l'architecture et du matériel Informatique utilisé.

L'algorithme proposé, effectue l'appariement des minuties en moins de 4s en moyenne. Sur les deux bases de données utilisées, suivant le seuil de décision ϵ fixé (de 12 à 30 minuties), le système global présente un taux de fausse acceptation (TFA) variant de 0.4 à 4.6%, pour un taux de faux rejet (TFR) variant de 0.5 à 4.9% comme le montre la figure 4.13 ci-dessous.

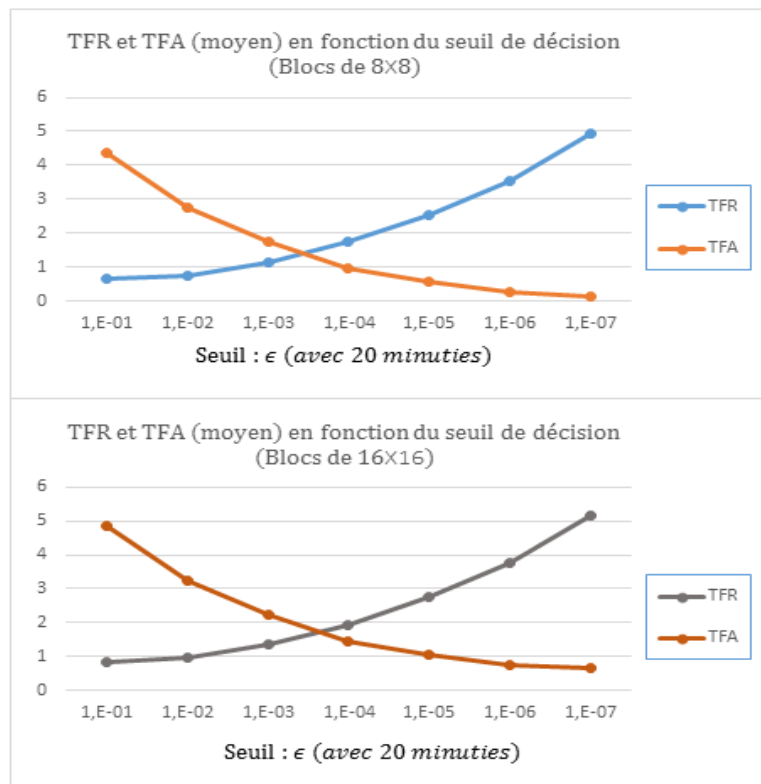


FIGURE 4.13 – TFA et TFR en fonction du seuil de décision

On constate qu'au fur et à mesure que le seuil de décision devient strict, le TFR augmente pendant que le TFA diminue. Ceci s'explique entre autres raisons par le fait que lorsque l'on renforce le seuil de décision, la probabilité de retrouver des gabarits identiques s'amenuise, particulièrement pour les empreintes provenant du même doigt mais recueillies différemment.

D'un autre côté lorsque le seuil de décision est fixé à partir du calcul de l'erreur quadratique entre la sortie attendue et celle obtenue après une itération, l'algorithme

proposé effectue l'appariement des minuties en moins de 2s en moyenne. Le seuil de décision utilisé pour les deux algorithmes est fixé (de 10^{-1} à 10^{-7}) avec 20 minuties en moyenne ; le système présente alors un TFA variant de 0.1 à 4.2%, pour un taux de faux rejet (TFR) variant de 0.4 à 5.1% comme le montre la figure 4.14 ci-dessous. Le même constat que précédemment est fait, à savoir qu'au fur et à mesure que le seuil de décision devient strict, le TFR s'accroît pendant que le TFA diminue.

Cependant, il est intéressant de préciser ici qu'une valeur trop stricte du seuil de décision provoquerait une discordance dans les résultats finaux des TA ; car en effet, pour deux gabarits (provenant du même doigt) présentant le même nombre de minuties détectées mais ayant subis des opérations de traitement différentes, le système générera une non-correspondance. C'est pour cette raison qu'il serait judicieux de fixer les seuils en fonction du résultat recherché.

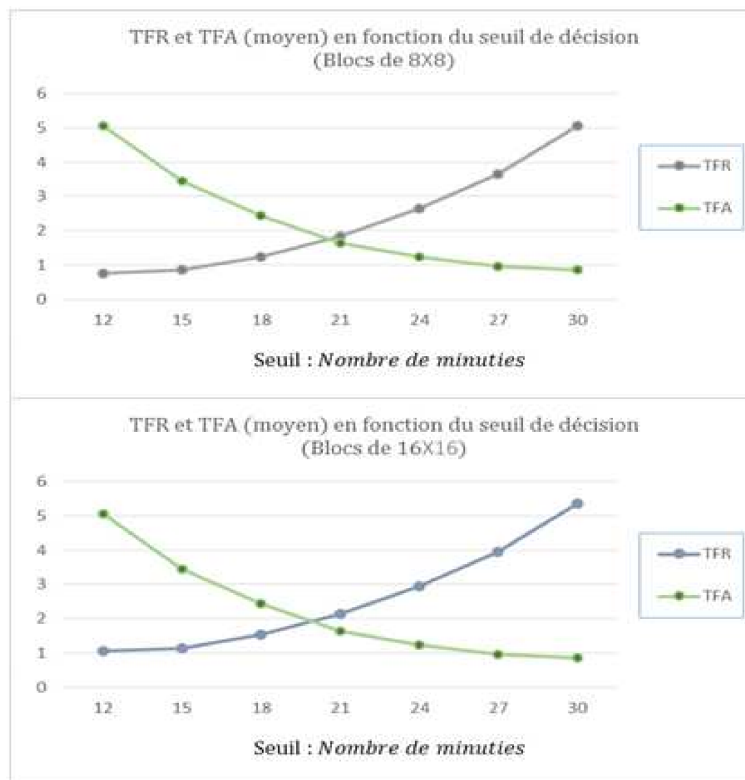


FIGURE 4.14 – TFA et TFR en fonction du nombre de minuties

Recommandations

Dans un contexte d'apprentissage profond, choisir le nombre de couche cachée en fonction du nombre de grandes opérations à réaliser dans le domaine dont il est question, et en fonction des caractéristiques sous-jacentes au problème à résoudre, afin d'optimiser

le choix des couches cachées (lieu où s'effectue la prédiction proprement dite) et d'éviter d'introduire des couches inutiles. Chacune de ces opérations pourraient alors constituer un niveau d'abstraction. Ces niveaux d'abstraction seraient traités de façon individuelle, tout en considérant les sorties/résultats des uns comme entrées des autres et, permettra d'éviter les opérations coûteuses en temps de calcul du "pooling".

Observations

Construction d'un réseau de neurones (RN) en fonction du problème à résoudre :

- réseau de neurones simples : utilisation de plusieurs paramètres pour palier à l'insuffisance du nombre de couche cachée. La qualité du résultat dépend de la qualité et du nombre de paramètres (qui peut croître de façon exponentielle lorsqu'on passe d'un réseau de profondeur d à un réseau de profondeur $d - 1$).
- réseau de neurones profonds : la qualité du résultat dépend des connexions locales, des convolutions et du pooling. Ce qui peut entraîner des temps d'apprentissage beaucoup trop grands.
- réseau de neurones profonds spécifiques (notre proposition) : la qualité du résultat dépend des connexions locales, des opérations inhérentes au domaine d'étude et des caractéristiques sous-jacentes audit domaine. Ce qui permet de se concentrer à l'essentiel et d'éviter des surcoûts liés au temps et la complexité des données à traiter.

Exemple : reconnaissance de l'iris

Après numérisation de l'image de l'œil, cette technique d'identification consiste en quatre grandes étapes à savoir :

- la détermination du centre de la pupille,
- la détermination du contour de l'iris,
- la construction des bandes de tailles égales,
- l'analyse de la texture de l'iris.

Pour résoudre donc un problème lié à l'identification par l'iris, nous recommandons entre autres :

- de construire un réseau profond avec quatre couches cachées localement connectées, et d'initialiser ces couches avec les éléments issus du traitement de l'image de l'œil lors des quatre grandes opérations suscitées pendant la phase d'apprentissage.
- de remplacer les calculs mathématiques sur une région singulière du vecteur caractéristique par des convolutions qui prendront en compte l'ensemble du champ visuel dudit vecteur.

4.5 Conclusion

Les travaux effectués sur la correspondance des empreintes digitales ont conduit à proposer un nouvel algorithme basé uniquement sur les atouts de ces derniers, à savoir : le calcul des probabilités pour établir la correspondance entre des vecteurs caractéristiques invariants à la translation/rotation. Les résultats obtenus démontrent qu'il y a bien adéquation entre les couches du système de reconnaissance. L'un des résultats intéressants de notre travail démontre que plus le seuil de décision est strict, plus la probabilité que le système fournisse un résultat satisfaisant est élevée. Mais en contrepartie, la complexité de l'algorithme est accrue, ce qui entraîne une augmentation des temps d'apprentissage et de reconnaissance. De plus le choix d'un seuil de décision raisonnable est recommandé afin d'éviter l'effet pervers d'un surcoût sur les *TA* dû à un seuil jugé trop strict.

Il est également à noter que la qualité de l'image d'empreinte de départ joue un rôle prépondérant sur la fiabilité des résultats ; en effet, une image non bruitée permettrait d'atteindre des *TA* de l'ordre de 99% ($\pm 2\%$) avec des *TFA* de l'ordre de 0.5% ($\pm 2\%$).

Le choix du domaine d'application reste ouvert et à la charge de l'utilisateur final.

Enfin, l'algorithme proposé pourrait être facilement enrôlé sur une carte à puce [82], afin de parachever le système complet de reconnaissance automatique d'empreinte digitale.

Conclusion Générale

L'objectif de cette thèse était de proposer un algorithme de reconnaissance d'empreintes digitales basée sur les réseaux de neurones artificielles pour l'authentification, dans un contexte de réduction des taux de faux rejets. Après avoir introduit la biométrie et ces différentes caractéristiques à savoir : les empreintes digitales, le visage, la main, l'iris de l'œil, l'empreinte vocale, la dynamique du tracé de la signature et les autres techniques (la rétine, le système de configuration des veines et la dynamique des frappes au clavier, ...), les principes fondamentaux de la reconnaissance d'empreintes digitales ont été présentés tout en décrivant de manière succincte et précise les différentes étapes intervenant dans un système automatique d'identification par empreintes digitales. Les différents types de réseaux de neurones ont également été décrits tout en exhibant leurs principes de base. Enfin, quelques uns de ces réseaux de neurones qui sont impliqués dans le processus de reconnaissance d'empreinte digitale ont été présentés.

Les résultats obtenus avec les différentes techniques étudiées montrent que l'utilisation des blocs d'images permet de réaliser une reconnaissance avec des taux de fausses acceptations bas dans l'ordre ± 0.1 en moyenne. Par contre, lorsque le système de reconnaissance utilise les classes de moyennes des images entières (algorithme propre à la méthode basée sur les réseaux de neurones localement connectés), ces mêmes taux de fausses acceptations sont très élevés et traduisent la subjectivité de cette méthode. Par ailleurs l'utilisation d'une base de données analytique associée aux calculs de probabilités deux à deux pour la classification d'une donnée nouvelle est appropriée pour l'authentification. De nombreux systèmes de reconnaissance d'empreintes pour l'authentification ont déjà été mis sur pied dans la littérature. La méthode proposée est assez novatrice et prend en compte aussi bien le calcul des probabilités que la comparaison des minuties appartenant à des vecteurs invariants à la translation/rotation pour effectuer la classification, tout en mettant à contribution le principe novateur des réseaux de neurones profonds, ce qui traduit l'originalité de cette approche [95]. Un effort tout particulier a été employé pour prendre en compte d'autres singularités de lignes d'empreintes digitales pour le cas des images provenant du même individu, notamment l'utilisation des

vecteurs caractéristiques avec calcul de l'énergie globale des données. L'utilisation de l'énergie globale des données de chaque base, couplée à l'utilisation des blocs de données plutôt que des valeurs individuelles a pour effet de réduire l'écart entre les blocs de pixels voisins et de diminuer l'effet de rotation lorsqu'il existe.

Les tests effectués avec la technique proposée ont conduit à des résultats encourageants bien que perfectibles. En effet nous nous sommes limités à effectuer des tests avec une erreur quadratique maximale fixée à 10^{-7} entre la sortie attendue et celle obtenue, car au delà de cette valeur, les résultats obtenus sont quelque peu biaisés. Il va sans dire que ce problème est certainement lié aux opérations de traitement effectuées sur les images lors de la comparaison et qu'une application de plusieurs filtres sans perte d'informations permettrait d'accroître les seuils sans grand danger et donc d'améliorer les performances en terme de taux d'erreur généré.

Une des limitations majeures des approches fondées sur l'apprentissage en général, et sur les réseaux de neurones 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 préapprentissage de certaines couches du réseau ou à faire du pooling lors de la phase d'apprentissage. Étant donné que le choix d'effectuer un apprentissage supervisé a été fait dans le cadre de cette thèse, les entrées correspondent aux sorties et l'entraînement concourt à ce qu'il y ait correspondance entre la sortie obtenue et celle attendue. La question du préapprentissage reste donc ouverte en ce qui concerne les poids de connexion entre différents neurones de la couche cachée.

Perspectives

En l'état actuel, les performances des différentes approches étudiées sont satisfaisantes pour des images de bonnes qualités. Cependant une étude plus approfondie sur le choix et le nombre de fois que doivent être appliquées les opérations de traitement sur les images d'empreintes permettrait d'avoir des algorithmes plus performants et mieux adaptés au cas des images latentes ou à moitié visibles. Car rappelons le, lorsque l'image est latente, la tentative de suppression du bruit provoque une élimination de certaines minuties présentes dans les zones fortement bruitées. Dans ces conditions, les opérations de traitement font plus de dégâts même dans le cas d'une convolution et, dégradent les éléments caractéristiques qui servent de base de calcul aux algorithmes de reconnaissance utilisés.

Le système de reconnaissance automatique d'empreintes digitales proposé est spécifique à l'authentification et applicable entre autres domaines au *E-learnig*. Seulement, ce système utilise les traits caractéristiques d'une empreinte digitale (type de minuties,

orientation des stries, etc.) pour générer le gabarit qui servira à effectuer la correspondance entre l'empreinte en entrée et celle déjà enregistrée dans la base. Vraisemblablement, le système tel que conçu ne permet pas d'authentifier un utilisateur déjà connecté pendant que ce dernier est en train d'utiliser les services de la plate-forme. Autrement dit, un utilisateur peut se faire authentifier avec son empreinte digitale et se connecter à la plate-forme ; après quelques instants un tiers parvient à accéder au système avec les coordonnées du précédent utilisateur (de façon frauduleuse ou même de façon concertée). A ce moment, pour le système c'est le même utilisateur qui est connecté et pourtant ce n'est plus le cas dans la réalité.

Des travaux futurs pourraient se tourner vers la mise sur pied d'un système d'authentification qui utilise en plus des empreintes digitales, d'autres techniques biométriques telles que : la forme du visage, la façon dont le doigt est posé sur le capteur et/ou la dynamique des frappes au clavier. La combinaison de ces techniques permettra de mettre en évidence un élément fondamental dans la reconnaissance avant et pendant l'utilisation du système : *la signature de l'utilisateur*. En effet l'utilisation de la *signature* plutôt que le *gabarit* à ceci de particulier qu'il permet d'enregistrer en plus des traits caractéristiques de l'empreinte digitale, un ensemble d'autres éléments spécifiques au comportement individuel de chaque utilisateur et rend ainsi le système capable d'identifier un changement de comportement brutal d'un utilisateur qui sera immédiatement débouté.

D'autres travaux pourraient également s'intéresser aux choix et au nombre de fois que doivent être appliquées les opérations de dropout, dans un contexte d'apprentissage profond avec comparaison de vecteurs caractéristiques invariants à la translation/rotation par calculs de probabilités.

Bibliographie

- [1] H. A. Abdullah. *Fingerprint Identification System Using Neural Networks*. Nahrain University, College of Engineering Journal (NUCEJ) Vol.15 N^o.2, (2012), pp 234 – 244.
- [2] O.E. Agazzi, S.S. Kuo, E. Levin, R. Pieraccini, "Connected and degraded text recognition using planar hidden Markov models", Proc. of ICASSP'93, 1993, pp. V113 – V116.
- [3] 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.
- [4] R. Andrews, J. Diederich and A. Tickle. "Survey and critique of techniques for extracting rules from trained artificial neural networks". Knowledge-Based Systems, 8(6) : 373 – 389, 1995.
- [5] C. Arcelli and G.S.D. Baja, "A Width Independent Fast Thinning Algorithm", IEEE Transaction on pattern Analysis and Machine Intelligence, Vol.4, N^o.7, pp. 463 – 474, 1984.
- [6] Atanu Chatterjee, Shuvankar Mandal, G. M. Atiqur Rahaman, and Abu Shamim Mohammad Arif, "Fingerprint Identification and Verification System by Minutiae Extraction Using Artificial Neural Network". Copyright ©2010 JCIT, ISSN 2078 – 5828 (Print), ISSN 2218 – 5224 (online), Volume 01, Issue 01, manuscript code : 100703.
- [7] Avinash Kumar Ojha. *ATM Security using Fingerprint Recognition*. (IJARCSSE) International Journal of Advanced Research in Computer Science and Software Engineering, Vol.5, (2015), pp. 170 – 175
- [8] A.S. Baek and N.H. Farhat, Biomorphic networks : approach to invariant feature extraction and segmentation for ATR, Proceedings-SPIE The International society for optical engineering, 3462 (3rd conference), 1998, 228 – 241.

- [9] W.A. Barrett, "A survey of face recognition algorithms and testing results", Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers, pp. 301 – 305, 1997.
- [10] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski. "Face recognition by independent component analysis". Transactions on Neural Networks, August 15, 2002.
- [11] P. Baldi, Y. Chauvin, "Neural networks for fingerprint recognition", Neural computation 5, 402 – 418 (1993), Massachusetts Institute of Technology.
- [12] Marco Balsi et Stefano David. "Cellular neural network for image compression by adaptive morphological subband coding". Dans *13 European Conference on Circuit Theory and Design, (ECCTD'97)*, volume 2, pages 634 – 638, 1997.
- [13] S. Bartolic-Zlomislic, A.W.T. Bates, (1999). "Investing in on-line learning : Potential benefits and limitations". Canadian Journal of Communication, 24(3), 349 – 366.
- [14] B. Batchakui, C. Tangha, J. S. Kameni. (2012). "xCCM : an alternative method of appropriate contents creation on an e-Learning platform". *IEEE Global Engineering Conference (EDUCON)*, Maroc, Institute of Electrical and Electronics Engineers (IEEE) (ed), pp 243 – 248.
- [15] A.M Bazen and S.H. Gerez, "An intrinsic Coordinate System for Fingerprint Matching", in Proc. Int. Conf. on Audio- and Video-Based Biometric Person Authentication (3rd), pp. 198 – 204, 2001.
- [16] Y. Belgnaoui, J-C. Guézel et T. Mahé. "La biométrie, sésame absolu. Industries et techniques", Juillet 2000.
- [17] Bengio, Y. and O. Delalleau (2011). *Shallow vs. deep sum-product networks*. In NIPS'11.
- [18] N. Benamrane, Z.B. Daho, J. Shen. "Compression des images médicales par réseau de neurones". USTO, traitement du signal. (1998) Vol. 6, pp. 631 – 638.
- [19] C. M. Bishop. "Pattern Recognition and Machine Learning". Springer, 2006.
- [20] R.E. Blahut, 1987. "Principles and Practice of Information Theory". Addison-Wesley, Reading, MA.
- [21] J.L Blue, G.T. Candela, P.J. Grother, R. Chellappa and C.L. Wilson, "Evaluation of pattern classifiers for fingerprint and OCR applications". Pattern recognition, 27(4), 1994, 485 – 501.
- [22] E. de Bodt, M. Cottrell, et M. Verleysen. "Using the Kohonen Algorithm for Quick Initialisation of Simple Competitive Learning algorithm". Dans : *European symposium on artificial neural networks*. (1999).

- [23] D. Boukerroui, J.A. Alison and M. Brady, "*On the choice of Bande-Pass Quadrature Filters*", Journal of Mathematical Imaging and Vision, Vol.21, Issue 1, pp. 53 – 80, July 2004.
- [24] 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) : 3481, 2009.
- [25] I. Buciu. "*Non-negative matrix factorization, A new tool for feature extraction : Theory and applications*". International Journal of Computers, Communications & Control, III(S.) : 6774, May 2008.
- [26] P. Buyskens and M. Revenu. "*Learning sparse face features : Application to face verification*". In International Conference on Pattern Recognition (ICPR), Istanbul, August 2010.
- [27] E. J. Candès and T. Tao. "*Decoding by linear programming*". IEEE Transactions on Information Theory, 51, 2005.
- [28] R. Cappelli, A. Lumini, D. Maio and D. Maltoni, "*Synthetic fingerprint-Database Generation*", proceeding of the 16th International Conference on Pattern Recognition (ICPR 2002), Québec City, Vol. 3, pp 744 – 747, August 2002.
- [29] R. Cappelli, D. Maio, and D. Maltoni. "*Combining fingerprint classifiers*". In Multiple Classifier Systems, pages 351 – 361, 2000.
- [30] A. Cichocki et R. Unbehauen. "*Neural Networks for Optimization and Signal Processing*". Wiley, (1993).
- [31] H. Chang, D. Y. Yeung, and Y. Xiong. "*Super-resolution through neighbor embedding*". In IEEE Conference on Computer Vision and Pattern Recognition, pages 275282, 2004.
- [32] H. B. Cheikh El Hocine and M. Marir-Benabbas. Etude des performances électriques et thermiques d'un capteur hybride PVT. Revue des Energies Renouvelables Vol. 16 N°2 (2013) 257 – 268.
- [33] 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.
- [34] Y. Chen, S. Dass , and A. Jain, "*A Localized Iris Image Quality Measure Using 2D Wavelets*," in Proc. of International Conf. on Biometrics, HongKong, China, Jan 6 – 8, 2006.
- [35] Vincent Cheung, Kevin Cannons. "*An Introduction to Probabilistic Neural Networks*". Signal & Data Compression Laboratory Electrical & Computer Engineering University of Manitoba Winnipeg, Manitoba, Canada, June 2002.

- [36] Cho S., Kim J. H., "*Bayesian Network Modeling of Hangul Characters for on-Line Handwriting Recognition*", Proceeding of the Seventh International Conference on Document Analysis and Recognition (ICDAR03), 2003.
- [37] L.O. Chua and L. Yang. "*Cellular Neural Networks : Theory*". IEEE Transactions on Circuits and Systems, 35(10) : 1257 – 1272, 1988.
- [38] C. Coudert. *Sagem security : new terminal for fingerprint recognition and biometrics techniques*. Sagem security, stand 409. Hitachi Europe, London Biometrics, 2009.
- [39] Cox, R. T. 1946. "*Probability, frequency and reasonable expectation*". Am. J. Phys. 14(1), 1 – 13.
- [40] D. Dardari, C.C. Chong, and M.Z. Win, "*Analysis of threshold-based toa estimators in uwb channels*". Proceedings of 14th European Signal Processing Conference (EUSIPCO 06), Sept. 2006.
- [41] G. Davis, S. Mallat, and Z. Zhang. "*Adaptive time-frequency decompositions*". Optical Engineering, 33(7) : 2183 – 2191, July 1994.
- [42] D. DeMers and G. Cottrell. "*Non-linear dimensionality reduction*". In Neural Information Processing Systems, pages 580 – 587, 1993.
- [43] John S. Denker, "*Les Réseaux de Neurones*". InterEditions, pp.118, 1990.
- [44] A. Devijver and J. Kittler. Pattern recognition : A statistical approach (London ; Prentice-Hall, 1982).
- [45] 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.
- [46] D. L. Donoho and C. Grimes. Hessian eigenmaps : "*New locally linear embedding techniques for high-dimensional data*". Technical report, July 02 2003.
- [47] R.D. Dony and S. Haykin. "*Neural Network approaches to image compression*". Proceedings of the IEEE, 83(2) : 288 – 303, 1995.
- [48] J. Doublet. "*Étude de nouveaux caractères biométriques de la main dans un contexte télécom*". Thèse de doctorat, Université de Caen, December 2009.
- [49] R. O. Duda, P. E. Hart and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.
- [50] G. Dreyfus, J.M. Martinez, M. Samuelides M. Gordon, F. Badran et S. Thiria. *Apprentissage statistique*. Eyrolles, 2008.
- [51] R. Eckhorn, H.J. Reitboeck, M. Arndt and P. Dicke. "*Feature linking via synchronization among distributed assemblies : simulations of results from cat visual cortex*". Neural Computation. 2, 1990, 293 – 307.

- [52] Y. Emyrodlu. "*Fingerprint image : enhancement and recognition*". Rapport de thèse, Université d'Herdforshire, Octobre 1997.
- [53] Erhan, D., Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio (2010). *Why does unsupervised pre-training help deep learning?* Journal of Machine Learning Research 11, 625?660.
- [54] Ken Ferens, W. Lehn, et Witold Kinsner. "*Image Compression Using Learned Vector Quantization*". Dans *Wescanex'93*, 1993.
- [55] A.P.Fitz and R.J.Green, "*Automatic Fingerprint identification using cluster algorithm*", IEEE Trans. Pattern Anal. and Machine Intel l., Vol. 17,N°4, 2002.
- [56] D. Gabor. "*Theory of Communication*". J. Inst. Electr. Eng., 93 : 429 – 457, 1946.
- [57] S. Gallant. "*Perceptron based learning algorithms*". IEEE Transactions on Neural networks, 1 : 179 – 191, 1990.
- [58] F. Galton, "*Fingerprint*", London, McMillan and Co., 1892.
- [59] Geoffrois E., Chevalier S., Prêteux F., "*Programmation dynamique 2D pour la reconnaissance de caractères manuscrits par champs de Markov*", Reconnaissance des Formes et Intelligence Artificielle, janvier 2004.
- [60] M. Gilloux, "*Reconnaissance de chiffres manuscrits par modèle de Markov pseudo-2D*". Traitement du Signal 1995 - Volume 12 - n°6.
- [61] R. Giot. "*Thèse : Contributions à la dynamique de frappe au clavier : multibiométrie, biométrie douce et mise à jour de la référence*". Machine Learning. Université de Caen, France, 2012.
- [62] Xavier Glorot. "*Apprentissage des réseaux de neurones profonds et application en traitement automatique de la langue naturelle*". Thèse de Doctorat, Département d'Informatique et de recherche opérationnelle, Facultés des arts et des Sciences. Université de Montréal. Novembre 2014.
- [63] Glorot, X., A. Bordes, and Y. Bengio (2011). "*Deep sparse rectifier neural networks*". In Proc. of AISTATS ?11.
- [64] L.O. Gorman. "*Comparing passwords, tokens and biometrics for user authentication*", *Proceedings of the IEEE*, Vol. 91, N12, pp. 2019 – 40, Dec.2003.
- [65] Govard Bidloo, "*Human body anatomy*", London, 1684.
- [66] Gram-Schmidt, "*Orthogonalization*", in Earliest Known Uses of Some of the Words of Mathematics (G), 1989.
- [67] Nehemiah Grew, "*Philosophical Transactions of the Royal Society of London*", London, 1684.

- [68] S. Grossberg, "*Adaptive Pattern Classification and Universal Recoding : I. Parallel Development and Coding of Neural Feature Detectors*", *Biological Cybernetics*, 23 : 121 – 134, 1976.
- [69] Hallouli K., Likforman-Sulem L., Sigelle M., "*Réseaux Bayésiens Dynamiques pour la reconnaissance des caractères imprimés dégradés*", GRETSI, CD-ROM, 2003.
- [70] Håstad, J. and M. Goldmann (1991). *On the power of small-depth threshold circuits*. *Computational Complexity* 1, 113 ?129.
- [71] A. Hayar, H. Anouar, and B. Zayen. "*Performance analysis of some fingerprinting based localization techniques using uwb signaling*". IEEE Information Theory Winter School, Mar. 2009.
- [72] D.O. Hebb. "*The Organization of Behavior*". Lawrance Erlbaum Associates, (1949).
- [73] M. Heidari, F.O. Akgul, N.A. Alsindi, and K. Pahlavan. "*Neural network assisted identification of the absence of direct path in indoor localization*". *Global Telecommunications Conference*, pp. 387 – 392, Nov. 2007.
- [74] International Biometric Group, "*The Henry Classification*", www.biometricgroup.com, consulted on 28 Feb. 2011.
- [75] B.K. Heung, H.J. Sung, G.K. Tag, and P. Kyu, "*Fast learning method for back-propagation neural network by evolutionary adaptation of learning rates*". *Neurocomputing*, pp. 101 – 106, 1996.
- [76] J.J. Hopfield. "*Neural networks and physical systems with emergent collective computational abilities*". *Proceedings of the National Academy of Sciences*. (1982). pp. 2554 – 2558.
- [77] Thomas Hofmann et Joachim Buhmann. "*Competitive Learning Algorithms for Robust Vector Quantization*". *IEEE Transactions on Signal Processing*, 46(6) : 1665 – 1675, 1998.
- [78] A.K. Hrechak and J.A Mc Hugh, "*Automated Fingerprint Identification Using Structured Matching*", *Pattern Recognition*, 23, 1990, 893 – 904.
- [79] Igaki, S., Eguchi, S., and Shinzaki, T. 1990. "*Holographic fingerprint sensor*". *Fujitsu Tech. J.* 25(4), 287 – 296.
- [80] S. Impedevo, L. Ottaviano, and S. Occhingro. "*Optical Character Recognition - A Survey*". *International Journal of Pattern Recognition and Artificial Intelligence*, 1991.
- [81] P.K. Isenor and S.A Zapy. "*Fingerprint Identification using graph matching, Pattern recognition*". 19, 1986, 113 – 122.

- [82] A.K. Jain, S. Shaoyun, K. Karu, N.K. Ratha. "A real time matching system for large fingerprint databases". IEEE Trans. on pattern analysis and machine intelligence, Vol. 18, n°8, August 1996.
- [83] J. Jiang. "Image compression with neural networks, A survey". Signal Processing : Image Communication, (1998).
- [84] J. Jiang, "A neural network based lossless image compression", in : Proc. Visual'96 : Internat. Conf. on Visual Information Systems, 5 – 6 February 1996, Melbourne, Australia, pp. 192 – 201, ISBN : 1 – 875 – 33852 – 7.
- [85] M.L. Johnson, "Biometrics and the threat to civil liberties", Computer, Volume 37, Issue 4, pp. 90 – 92, April 2004.
- [86] Kawagoe M. and Toko A., "Fingerprint Pattern Classification", Pattern Recognition, vol. 17, pp. 295 – 303, 1984.
- [87] Kaymaz E. and Mitra S., "Analysis and Matching of Degraded and Noisy Fingerprints," Proc. of SPIE (*Application of Digital Image Processing XV*, vol. 1771), pp. 498 – 508, 1992.
- [88] T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps", Biological Cybernetics, 43 : 59 – 69, 1982.
- [89] T. Kohonen, "Self-Organisation and Associative Memory", Springer, Berlin, 1984.
- [90] T. Kohonen, "Learning Vector Quantization for Pattern Recognition", Technical Report *TKK – F – A601*, Helsinki University of Technology, Department of Technical Physics, Laboratory of Computer and Information Science, November 1986.
- [91] T. Kohonen, "Self-Organizing Maps : Optimization Approaches", In T. Kohonen, K. Makisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, pages 981 – 990. Elsevier Science Publishers, 1991.
- [92] S. Kouamo, P. Melatagia, R. Ndoundam, "La compression d'images avec les réseaux de neurones artificiels". Dans actes du Colloque Africain sur la Recherche en Informatique et en Mathématiques appliquées (*CARI'08*). Vol., 151. pp 535 – 543, 2008.
- [93] S. Kouamo, C. Tangha, "Handwritten character recognition with artificial neural network". Distributed Computing and Artificial Intelligence. Advances in Intelligent and Soft Computing. Vol., 151. pp 535 – 543. Springer Verlag 2012.
- [94] S. Kouamo, C. Tangha, "Images compression with artificial neural network". Advances in Intelligent and Systems and Computing. Vol., 189. pp 515 – 524. Springer Verlag 2013.

- [95] S. Kouamo, C. Tangha, "*Fingerprint recognition with artificial neural networks : Application to E-learning*". Journal of Intelligent Learning Systems and Applications Vol., 8. N2. Pub. Date : May 30, 2016, DOI : 10.4236/jilsa.2016.82004.
- [96] A. Kundu, Y. He, P. Bahl, "*Recognition of handwritten word : First and second order Hidden Markov Model based approach*", in the proceedings of CVPR 88, pp 457 – 462, 1988.
- [97] Ashok K. Krishnamurthy, Stanley C. Ahalt, Douglas E. Melton and Prakoon Chen. "*Neural Networks for Vector Quantization of Speech and Images*". IEEE Journal on Selected Areas in Communications, 8(8) : 1449 – 1457, octobre 1990.
- [98] R. Lancini et S. Tubaro. "*Adaptive Vector Quantization for Picture Coding using Neural Networks*". IEEE Transactions on Communications, 43(2/3/4) : 534 – 544, 1995.
- [99] Lee H.C. and Gaensslen R.E., "*Advances in Fingerprint Technology*", 2nd edition, Elsevier, New York, 2001.
- [100] Y. Le Cun. "*Une procédure d'apprentissage compétitif pour réseau à seuil asymétrique*". Dans Cognitiva 85 : A la Frontière de l'Intelligence Artificielle des Sciences de la Connaissance des Neurosciences, pp. 599 – 604, 1985.
- [101] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel. 1990. Handwritten digit recognition with a backpropagation network. "*In Neural Information Processing Systems*", Vol.2, pp. 396 – 404. Morgan Kaufmann, San Mateo, CA.
- [102] Y. Le Cun, S. Chopra, R. Hadsell, R. marc'aurelio, and f. Huang (2006). "*A tutorial on Energy-Based learning*". In G. Bakir, T. Hofman, B. schölkopf, A. Smola, and B. Taskar (Eds.), Predicting Structured Data. MIT Press.
- [103] Y. Le Cun, Y. Bengio and G. Hinton. "*Deep learning*". Macmillan Publishers Limited. Nature, Vol.521, pp. 436 – 444. May 2015. DOI :10.1038/nature14539.
- [104] Y. Le Cun, Y. and Y. Bengio (1994). "*Word-level training of a handritten word recognizer based on convolutional neural networks*". In IEEE (Ed.), International Conference on Pattern Recognition (ICPR'94), Jerusalem 1994.
- [105] Lettvin, J. Y., McCulloch, Maturana, H.R., W. Sturgis. et W. H. Pitts. "*What the frog's eye tells the frog's brain*". Proceedings of the I.R.E. (1959). Vol. 47, N^o.11, pp. 1940 – 1951.
- [106] Lee, H., P. Pham, Y. Largman, and A. Ng (2009). "*Unsupervised feature learning for audio classification using convolutional deep belief networks*". In Y. Bengio, D. Schuurmans, C. Williams, J. Lafferty, and A. Culotta (Eds.), Advances in Neural Information Processing Systems 22 (NIPS '09), pp. 1096 – 1104

- [107] J. Li, M. Testorf and M.A. Fiddy, "*Fourier Properties of Fingerprints*", Proc. SPIE Enforcement and Security Technologies, Vol. 3575, pp. 201 – 210, 1998.
- [108] H. Liu, H. Darabi, P. Banerjee, and J. Liu. "*Survey of wireless indoor positioning techniques and systems*". IEEE Transactions on Systems, Man, and Cybernetics, vol.37, n°.6, pp. 10671080, Nov. 2007.
- [109] D. Maio et D. Maltoni. "*Neural Network based minutiae filtering in fingerprints*". IEEE, 1998.
- [110] S.G. Mallat, "*A wavelet tour of signal processing*", Academic Press 2nd edition, 1994.
- [111] S.G. Mallat, "*A theory for multiresolution signal decomposition : The wavelet representation*", IEEE Trans. Pattern Anal. Mach. Intell. 11 (7) (July 1989) 674 – 693.
- [112] D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar, "*Handbook of fingerprint recognition*", Springer, New York, 2003.
- [113] C.N. Manikopoulos. "*Neural network approach to DPCM system design for image coding*". IEEE Proc. – I 139 (5) (October 1992) 501 – 507.
- [114] T. Matsumoto "*Impact of Artificial "Gummy" Fingers on Fingerprint Systems*". Proceedings of SPIE, Vol. 4677, pp. 275 – 289, 24 – 25 January 2002. pp. 46 – 57.
- [115] Y.Minami, S.Tamura, H.Sawai, K.Shikano, "*Output Smoothing for TDNN Using Information on Input Vectors Neighborhood in Input Layer, Hidden Layer*", Proc. ASJ, 1 – 3 – 18, pp.35 – 36 (Mar. 1990) (in Japanese).
- [116] Moenssens A., "*Fingerprint Techniques*", Chilton, London, 1971.
- [117] V. Monthe, B. Batchakui, C. Tangha, F. Tietche. (2012). "*MV-SYDIME : A virtual patient for medical diagnosis apprenticeship*". International Journal of Online Engineering (iJOE), Octobre 2012.
- [118] O. Moreira-Tamayo et J. Pineda de Gyvez. "*Preprocessing Operators for Image Compression Using Cellular Neural Networks*". Dans IEEE International Conference on Neural Networks, juin 1996. pp. 1500 – 1505.
- [119] M. Mostafa Abd Allah. "*Artificial Neural Networks Based Fingerprint Authentication With Clusters Algorithm*". Informatica 29 (2005) pp. 303 – 307.
- [120] Milan S., Vaclav H., and Roger B. "*Image processing analysis and machine vision*". London : Chapman and Hall, 255 – 280, 2002.
- [121] M. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge MA, 1969.
- [122] Mark Nelson and Jean-Loup Gailly. "*The Data Compression Book*". M&T Books, New York, second edition, 1995.

- [123] E. Oja. "A simplified neuron model as a principal component analyzer". Journal of Mathematical Biology, Vol.15, pp. 267 – 273, 1982.
- [124] Osterburg, Parthasarathy, Raghanvan and Slove. "Development of a mathematical formula for the calculation of fingerprint probabilities based on individual characteristics". Journal of the American statistical association, Vol. 72, n° 360, 1977.
- [125] S. R. Patil (Waghjale), Prof. S. R. Suralkar, "Fingerprint classification using artificial neural network", IJETAE ISSN 2250 – 2459, Volume 2, Issue 10, September 2012.
- [126] D. Petrovska Delacretaz, G. Chollet, and B. Dorizzi. "Guide to Biometric Reference Systems and Performance Evaluation". Springer, 2009.
- [127] Harold Cummins, Rebecca Wright Kennedy, Purkinje's Observations (1823) on Finger Prints and Other Skin Features, 31 Am. Inst. Crim. L. & Criminology 343 (1940 – 1941).
- [128] Khajeh Rashiduddin Fazlollah Hamadani, "Jaamehol-Tawarikh (histoire universelle)" , 1247 – 1318.
- [129] Y. Qian, J. Dong, W. Wang, and T. Tan. "Deep learning for steganalysis via convolutional neural networks". Media Watermarking, Security, and Forensics 2015, vol.9409 of Proceedings of SPIE, pp 9409J – 9409J – 10, San Francisco, CA, March 2015.
- [130] N. Ratha, R. Bolle. "Automatic Fingerprint Recognition Systems", Springer, New York, 2004.
- [131] S.A. Rizvi, N.M. Nasrabadi, "Finite-state residual vector quantization using a tree structured competitive neural network", IEEE Trans. Circuits Systems Video Technol. 7 (2) (1997) 377 – 390.
- [132] S.A. Rizvi, Ling-Cheng Wang, et Nasser M. Nasrabadi. "Neural Network Architectures for Vector Prediction". Proceedings of the IEEE, 84(10) : 1513 – 1528, 1996.
- [133] S.A. Rizvi et N.M. Nasrabadi. "Neural Networks for Image Coding : A Survey". Dans *IS&T/SPIE Conference on Applications of Artificial Neural Networks in Image Processing*, (1999). pp. 46 – 57.
- [134] Christine Redecker, "Review of Learning 2.0 Practices : Study on the Impact of Web 2.0 Innovations on Education and Training in Europe", JRC Scientific and technical report, n° EUR 23664 EN2009, (2009).
- [135] Herbert Robbins and Sutton Monro, "A Stochastic Approximation Method," Annals of Math. Stat., 22 : 400 – 407, 1951.

- [136] F. Rosenblatt. "*The Perceptron : A Probabilistic Model for Information Storage and Organization in the Brain*". Psychological Review. Vol. 65, N^o. 6, pp. 386 – 408. 1958.
- [137] Ross, J.L. et Schulz, R.A. (1999). "*Using the world wide web to accommodate diverse learning styles*". College Teaching, 47(4), 123 – 131.
- [138] D.E. Rumelhart and J.L. Mc Clelland. "*Explorations in the Microstructure of Cognition*". ISBN : 0 – 262 – 68053 – X, (1984).
- [139] D.E. Rumelhart, G.E. Hinton and R.J. Williams. "*Learning Internal representations by error propagation*". In Parallel Distributed Processing, Cambridge, MA : MIT Press, 1 : 318 – 362, 1986.
- [140] D.E. Rumelhart, G. E. Hinton and R.J. Williams. "*Learning representations by back-propagating errors*". Nature, 323 : 533 – 536, 1986.
- [141] Sakshica, Dr. Kusum Gupta, Banasthali Vidyapith, Jaipur Campus, Jaipur . (2015). *Handwritten Digit Recognition using various Neural Network Approaches*. International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 2, (February 2015) Copyright ©IJARCCE DOI 10.17148/IJARCCE.2015.4218 78.
- [142] T.D. Sanger. "*Optimal unsupervised learning in a single layer linear feedforward neural networks*". Neural networks 2. (1989). 459 – 473.
- [143] Schiaffino, S., Garcia, P., & Amandi, A. "*eTeacher : Providing personalised assistance to e-learning students*". Computers & Education 51 (2008) 1744 – 1754.
- [144] Jurgen Schmidhuber. "*Deep learning in neural network : an overview*". The Swiss AI Lab IDSIA. Istituto Dalle Molle di Studi sull'Intelligenza Artificiale University of Lugano & SUPSI. Galleria 2, 6928 Manno-Lugano Switzerland. 28 May 2014.
- [145] F.W. Schmidt. "*Neural pattern classifying systems, theory and experiments with trainable classifiers*". Thesis, Technische Universiteit Deif, 1994.
- [146] M. Sezgin and B. Sankur, "*Survey over image thresholding techniques and quantitative performance evaluation*". Journal of Electronic Imaging, Vol. 13, pp. 146 – 165, January 2004.
- [147] Donald F. Specht. "*Probabilistic neural networks*". Neural networks, Vol.3. pp. 109 – 118, 1990, printed in the USA.
- [148] J. Takami, S. Sagayama, "*A Pairwise Discriminant Approach to Robust Phoneme Recognition by Time-Delay Neural Networks*", Proc. of ICASSP91, 8.S2.13 (May 1991).

- [149] T.J Thomas, "*Locally-connected Neural Network for fingerprint recognition*". Proceedings of the IASTED International Conference, Intelligent Systems and Control 2000, Honolulu, Hawaii, USA.
- [150] J. Thornton, M. Savvides and B.V.K Vijaya Kumar "*A Bayesian Approach to Deformed Pattern Matching of Iris Images*", IEEE Transaction on Pattern Analysis and Machine Learning, Vol.29, N^o.4, April 2007.
- [151] J.M. Torres-Moreno. "*Apprentissage et Généralisation par des réseaux de neurones : étude de nouveaux algorithmes constructifs*". PhD thesis, Institut National Polytechnique de Grenoble, (1997).
- [152] 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.
- [153] N. Tsopzé, E. Mephu Nguifo, et G. Tindo. "*Une approche basée sur les treillis de Galois pour la construction des réseaux de neurones*". Dans actes du Colloque Africain sur la Recherche en Informatique et en Mathématiques Appliquées (CARI'08), pages 399 – 407, 2008.
- [154] K. Touafek, A. Malek et M. Haddadi, "*Etude Expérimentale du Capteur Hybride Photovoltaïque Thermique*", Revue des Energies Renouvelables, Vol. 9, N3, pp. 143 – 154, 2006.
- [155] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586 – 590, Hawaii, June 1992.
- [156] O.R Vincent, O. Folorunso. "*A Descriptive Algorithm for Sobel Image Edge Detection*". Proceedings of Informing Science & IT Education Conference (InSITE) 2009.
- [157] M.K. Sparrow and P.J. Sparrow. "*A topological approach to the matching of single fingerprints : development of algorithms for use on rolled impressions*". Technical report special publication sw – 124 (Washington, D.C., National Bureau of Standards, 1985).
- [158] Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). "*Dropout : A simple way to prevent neural networks from overfitting*". Journal of Machine Learning Research 15, 1929?1958.
- [159] Vijaya Sathiaraj, "*Study on the Neural Network Model for Finger Print Recognition*", IJCER ISSN 2250 – 3005 (Online), Volume 2, Issue 5, October 2012.
- [160] Viragkumar N. Jagtap and Shailendra K. Mishra. "*Fast Efficient Artificial Neural Network for Handwritten Digit Recognition*". International Journal of Computer Science and Information Technologies, Vol. 5 , (2014).

- [161] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon. "*Accelerating the convergence of the back propagation method*". *Biological Cybernetics*, 59 : 257 – 263, 1988.
- [162] Gustav N. Wassel and Jack Sklansky, "*Training a OneDimensional Classifier to Minimize the Probability of Error*," *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC* – 2(4) : 533 – 541, 1972.
- [163] C.I. Watson and C. L. Wilson, "*NIST Special Database 4 Fingerprint Database*" (National Institute of Standards and Technology, Advanced Systems Division, Image Recognition Group; March 1992).
- [164] G.O. Williams, "*Iris Recognition Technology*", *IEEE Aerospace and Electronics Systems Magazine*, Vol. 12, Issue 4, pp. 23 – 29, April 1997.
- [165] A.J. Willis and L. Myers, "*A cost-effective fingerprint recognition system for use with low-quality prints and damaged fingertips*", *Pattern Recognition*, Vol. 34, pp. 255 – 270, 2001.
- [166] C.L. Wilson, G.T. Candela, and C.I. Watson, "*Neural network fingerprint classification*", *Artificial Neural Networks*, Vol. 1, N^o.2, pp. 203 – 228, 1993.
- [167] Woolf, B.P., "*Building intelligent interactive tutors : student-centered strategies for revolutionising e-learning*". (USA, Morgan Kaufmann, 2009). pp 343 – 344.
- [168] J. Wood, *Invariant Pattern Recognition* : "*A review, Pattern Recognition*", 29 (1), 1996, 1 – 17.
- [169] X. Xia and L. O’Gorman, "*Innovations in fingerprint capture devices*", *Pattern Recognition*, Vol. 36, pp. 361 – 369, 2003.
- [170] J. Yang, L. Liu, T. Jiang and Y. Fan, "*A modified Gabor filter design method for fingerprint image enhancement*", *Pattern Recognition Letters*, Vol. 24, N^o.11, pp. 1835 – 1849, 2000.
- [171] P. Yonta Melatagia, R. Ndoundam, M. Tchuenté., "*Dynamical behavior of neural networks iterated with memory*". UNESCO and IAEA, The Abdus Salam International Centre for theoretical physics, *IC/2005/111*. (2005). Available at : [http : //www.ictp.it/pub – off](http://www.ictp.it/pub-off).
- [172] Lei Yu, M. Laaraiedh, S. Avrillon , B. Uguen, "*Fingerprint localisation based on neural networks and ultra-wideband signals*", *Signal Processing and Information Technology (ISSPIT)*, *IEEE International Symposium on*, Bilbao : Spain (2011) DOI : 10.1109/ISSPIT. 2011. 615155.
- [173] Zeiler, M. D. and R. Fergus (2014). "*Visualizing and understanding convolutional networks*". In *ECCV’14*.

- [174] W. Zhao, R. Chellappa, P.J. Phillips and A. Rosenfeld, "*Face recognition : A literature survey*", ACM Computing Surveys (CSUR), Vol. 35, Issue 4, December 2003.
- [175] [http : //www.cite – sciences.fr/](http://www.cite-sciences.fr/). Consulté le 30 Juin 2015.
- [176] [http : //policescientifique – role21.e – monsite.com/pages/content/empreintes – digitales/](http://policescientifique-role21.e-monsite.com/pages/content/empreintes-digitales/). Consulté le 07 Juillet 2015.
- [177] [http : //media – 2.web.britannica.com](http://media-2.web.britannica.com). Consulté le 30 Juin 2015.
- [178] [http : //onin.com/fp/fphistory.html](http://onin.com/fp/fphistory.html). Consulté le 02 Juillet 2015.
- [179] [http : //www.criminaljustice.ny.gov/ojis/history/fp _ sys.htm](http://www.criminaljustice.ny.gov/ojis/history/fp_sys.htm). Consulté le 05 Juillet 2015.

Annexes

A.1 Exemples d'images d'empreintes digitales légèrement bruitées



FIGURE 4.15 – Quelques images de la base de données utilisée par Baldi lors de ses tests : 512×464 avec 8 bits par pixel

A.2 Exemples d'images d'empreintes digitales de BDAL

Les images de la base de données BDAL ont été obtenus à partir des doigts des personnes coopératives et sensibilisée sur l'importance de l'opération envisagée. Cependant, comme le montre les figures 4.16 et 4.17 ci-dessous, quelques images de mauvaise

qualité ont quand même été enregistrées. Nous en avons certaines qui sont pratiquement illisibles, d'autres qui sont trop foncées, d'autres dont la tailles est inégale avec la norme de départ pour cette base de données, etc. Ceci traduit la délicatesse de l'opération de scannérisation et montre que même lorsque les utilisateurs sont assez sensibilisés, il faut prendre beaucoup de précaution afin d'éviter les cas d'images trop bruitées qui seraient difficiles pour notre système à reconnaître. Toutefois de nombreuses recherches ont déjà produit des résultats intéressants sur le cas d'images fortement bruitées [6, 71, 82].



FIGURE 4.16 – Quelques images de la base de données BDAL



FIGURE 4.17 – Quelques images de la base de données BDAL (2)

Index

- adaline, 73, 88
- algorithme de l'index de poincaré, 44, 45
- algorithme de Sobel, 45
- apprentissage compétitif, 93, 117
- apprentissage supervisé, 87
- apprentissage non supervisé, 89
- approche bayésienne, 99
- backpercolation, 88
- base de données, 111, 116
- base de données BDAL, 116, 125, 130
- base de données NIST, 111, 125, 131
- binarisation, 37, 49, 61
- camera CCD, 12, 28, 29, 100
- classification des empreintes, 69
- comparaison des empreintes 63, 64, 65
- convolution, 42, 53, 101, 103
- crossing number, 62, 63
- détection des minuties 53, 54, 56, 60
- direction différence, 65
- élagage, 85
- filtrage de Gabor, 55, 56
- filtrage fréquentielle auto-adaptatif, 59
- fonction d'activation, 80
- interpolation bilinéaire, 41
- learning vector quantization, 89
- machine de Cauchy, 88
- matching de minuties, 67
- minuties, 22, 27, 38, 60, 67
- minuties crossing number, 62,63
- modèle coordonnées, 67
- modèle récurrent, 83
- ordinateur de Von Neumann, 79
- perceptron multicouche, 87, 91, 98, 115, 132
- plus proches voisins, 111, 114
- k -plus proches voisins 111, 112
- prédiction, 91
- probabilistic neural network, 88
- probabilité de collision de 2 empreintes, 25
- quantification scalaire et vectorielle, 92
- règle de Hebb, 94
- règle d'apprentissage d'Oja, 91
- régions singulières 70, 71
- représentation spatiale de l'image, 104
- réseau de Kohonen, 89, 92
- réseau de neurones, 72, 77, 82, 92, 107, 113
- réseau de neurones artificielles, 3, 74

réseau de neurones probabilistes, 92, 99
réseau de neurones localement connectés, 107
réseau de Hopfield, 82
rétro propagation, 91, 98, 104, 114, 117
segmentation, 91
self organization map, 89
spatial difference, 67
squelettisation, 50
sur apprentissage, 85
time delay neural network, 88
type de minuties, 60, 61
type de réseau de neurones 86
taux d'acceptation, 125, 131
taux d'erreur, 125
propagation en avant, 83
xor, 77