

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ORAN
-MOHAMED BOUDIAF-

LA RECONNAISSANCE EN-LIGNE DU MANUSCRIT ARABE

par

Redouane TLEMSANI

FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

Thèse présentée

en vue de l'obtention du grade de **DOCTORAT EN SCIENCES**
en **INFORMATIQUE**

Option

Reconnaissance des Formes - Intelligence Artificielle
(RF-IA)

soutenue publiquement le : 23/02/2012 devant le jury:

Président	Mr Mohamed BENYETTOU	Professeur	USTO-MB
Rapporteur	Mr Abdelkader BENYETTOU	Professeur	USTO-MB
Examineurs	Mr Mohamed BENMOHAMMED	Professeur	Univ. Constantine
	Mr Younès BENNANI	Professeur	Univ. Paris 13
	Mme Naïma HADJ-SAID	Professeur	USTO-MB
	Mr Hafid HAFFAF	Professeur	Univ. Es Sénia

Résumé

Le travail se situe dans le domaine de la reconnaissance de caractères arabes isolés en ligne, et la motivation principale dans ce domaine est d'étudier le manuscrit arabe via à vis la technologie en ligne.

Notre système est un système Markovien que l'on peut voir comme un Réseau Bayésien Dynamique (Dynamic Bayesian Network DBN). Un des intérêts majeurs de ces systèmes réside dans l'apprentissage complet des modèles (topologie et paramètres) à partir de données d'apprentissage.

L'approche est basée sur le formalisme des réseaux bayésiens dynamiques. La théorie des DBNs est une généralisation des réseaux Bayésiens aux processus dynamiques. Parmi nos objectifs revient à trouver les meilleures paramètres qui représentent les liens (dépendances) entre les variables d'un réseau bayésien dynamique. Dans des applications en reconnaissance des formes, on procède généralement à la fixation de la structure qui oblige d'admettre quelques hypothèses fortes (par exemple l'indépendance entre certaines variables). L'application a porté sur la reconnaissance en ligne de caractères arabes isolés en utilisant notre base de données de laboratoire NOUN. Un testeur neuro-génétique a été proposé comme amélioration externe du DBN.

Les scores pour DBN et DBN hybridé sont respectivement **66.78%** et **73,47%** ce qui laisse présager leur exploitation ultérieure ; d'autres approches d'amélioration tenant compte du temps ont été considérées et implémentées jusqu'à obtention d'un taux important **94,79%** à base de reconnaissance.

Mots-clés : Reconnaissance des formes, Reconnaissance de Caractères Isolés en ligne, Manuscrit Arabe, Réseaux Bayésiens Dynamiques

الملخص:

يقع هذا العمل في مجال التعرف الآلي المباشر على الحروف العربية، والدافع الأساسي هو دراسة الكتابة العربية تجاه التكنولوجيا المباشرة. النظام المستعمل هو عبارة عن نظام "ماركوف" الذي نستطيع اعتباره شبكة "بايز" الديناميكية (*Dynamique Bayesian Network*).

واحد من الفوائد الأساسية لهذا النوع من الأنظمة يتلخص في التلقين الكامل لهذه النماذج (هندسة وخصائص) باستعمال معطيات التلقين الآلي.

الطريقة المتبعة تتمركز بصفة أساسية على خاصية شبكة "بايز" الديناميكية. نظرية DBN هي تعبير لشبكة "بايز" نحو الخاصية الوقتية الديناميكية. إن أحد الأهداف يرجع إلى إيجاد أحسن الخصائص التي تمثل الروابط المختلفة بين مكونات الشبكة.

إن بعض التطبيقات في مجال التعرف على الأشكال، تعتمد على تثبيت هندسة النموذج الذي يسمح بإدماج بعض الفرضيات القوية (على سبيل المثال الاستقلالية بين المتغيرات).

تطبيقيا، التجارب لخصت في التعرف الآلي المباشر على حروف اللغة العربية باستعمال قاعدة المعلومات الخاصة بالمخبر "نون". استعملنا أيضا اختبارا عصبيا وراثيا للتطوير الخارجي للشبكة.

نتائج البحث باستعمال DBN و DBN مختلط أعطت النسب على التوالي: 66.78% و 73,47% والتي سمحت بالاستعمال اللاحق لها، طرق أخرى طبقت واستعمل فيها الوقت للتطوير حتى بلوغ نسبة التعرف 94,79%.

الكلمات المفتاحية:

التعرف الآلي على الأشكال، التعرف الآلي المباشر على الحروف، الخط العربي، شبكة "بايز" الديناميكية.

Abstract

Work is in on line Arabic isolated character recognition and the principal motivation is to study the Arab manuscript with on line technology.

This system is a Markovien system which one can see as like a Dynamic Bayesian Network (DBN). One of the major interests of these systems resides in the complete training of the models (topology and parameters) starting from training data.

Our approach is based on the formalism of the dynamic Bayesian Networks. The theory of DBNs is a generalization of the Bayésiens networks to the dynamic processes. Among our objectives amounts finding better parameters which represent the links (dependences) enter the variables of a dynamic network. In applications in pattern recognition, one will carry out the fixing of the structure which obliges us to admit some strong assumptions (for example independence between some variables). Our application will relate to the on line recognition of Arabic isolated characters by using our laboratory data base: NOUN. A neuro-genetic tester proposed for DBN external optimization.

The scores for DBN and DBN mixed are respectively **70,24%** and **62,50%**, which lets predict their further development; other approaches taking account time were considered and implemented until obtaining a significant recognition rate **94,79%**.

Keywords: Pattern recognition, On line isolated character recognition, Arabic Manuscript, Dynamic Bayesians Networks.

CONTRIBUTION DE L'AUTEUR

1. Publication scientifique dans la revue IAJIT : International Arabe Journal of Information Technology, ISSN : 1683-3198, *Octobre 2011 Volume 8 Numéro 4 pp 406-413*
Redouane Tlemsani and Abdelkader Benyettou
« *On Line Isolated Characters Recognition Using Dynamic Bayesian Networks* »
2. 9th Maghrebien Conference on Software Engineering and Artificial Intelligence, Agadir, Morocco. MCSEAI'2006 December 07-09, 2006
Redouane Tlemsani and Abdelkader Benyettou
« *Application des réseaux bayésiens dynamiques à la reconnaissance en-ligne des caractères isolés* »
3. 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications Hammamet, Tunisia. SETIT 2007 March 25-27, 2007.
Redouane Tlemsani and Abdelkader Benyettou
« *Application des réseaux bayésiens dynamiques à la reconnaissance en-ligne des caractères isolés* »
4. Colloque sur l'Optimisation et les Systèmes d'Information. Oran, Algeria. COSI 2007 June 11-13, 2007
Redouane Tlemsani and Abdelkader Benyettou
« *La reconnaissance en-ligne des caractères isolés par les réseaux bayésiens dynamiques: segmentation en strokes élémentaires* »
5. International Conference on Information & Communication Systems Jordanie Amman ICICS 2009 November 1st 2009
Redouane Tlemsani and Abdelkader Benyettou
« *Segmentation and Recognition of On Line Isolated Characters Using Dynamic Bayesian Networks* »
6. Image and Signal Processing and their Applications, Université de Mostaganem Algérie ISPA 2009 October 19-21 2009
Redouane Tlemsani and Abdelkader Benyettou
« *Apprentissage et Reconnaissance d'un Signal d'écriture En ligne par les Réseaux Bayésiens Dynamique* »
7. The International Arab Conference on Information Technology University of Science and Technology, Yemen ACIT'2009 December 15-17, 2009
Redouane Tlemsani and Abdelkader Benyettou
« *On Line Recognition Isolated Characters Using Dynamic Bayesian Networks And Neuronal Verifier Test* »
8. Congres méditerranéen des télécommunications Casablanca Maroc. CMT 2010 March 18-20, 2010
Redouane Tlemsani and Abdelkader Benyettou
« *Modèle Neuro-Bayésien Dynamique pour la Reconnaissance d'un Signal* »

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE	1
Chapitre I : L'ÉCRITURE ARABE	3
I.1. Introduction	3
I.2. Alphabet arabe	4
I.3. Caractéristiques de l'écriture Arabe	6
I.3.1 Signes diacritiques	7
I.3.2 Ascendants et descendants	10
I.3.3 Une ou plusieurs composantes connexes par mot	10
I.3.4 Ligatures verticales	12
I.4 Conclusion	12
Chapitre II : RECONNAISSANCE D'UN SIGNAL D'ÉCRITURE EN LIGNE	13
II.1 Introduction	13
II.2 Reconnaissance Manuscrite	13
II.2.1 Reconnaissance Hors ligne	14
II.2.2 Reconnaissance En Ligne	15
II.3 Problématique générale de la reconnaissance d'écriture	16
II.4 Quelques exemples d'applications	17
II.5 Contexte de l'écriture en-ligne	18
II.5.1 Contexte matériel et les interfaces Stylo	19
II.5.2 Contexte logiciel	21
II.5.3 Axes de recherche autour de la reconnaissance en-ligne	22
II.6 Composants de caractère	23
II.7 Système de reconnaissance d'écriture manuscrite en ligne	24
II.8 Conclusion	25
Chapitre III : ÉTAT DE L'ART DES TRAVAUX RÉALISÉS	26
III.1 Introduction	26
III.2 Synthèse des travaux du manuscrit Arabe En ligne	27
III.2.1 Reconnaissance de caractères Arabes en-ligne par réseau de <i>Kohonen</i>	27

III.2.2	Reconnaissance de caractères Arabes en-ligne par maximum d'entropie	28
III.2.3	Reconnaissance en ligne du manuscrit arabe par les HMMs	30
III.2.4	Système de reconnaissance d'écriture arabe en ligne basé sur le codage visuel et l'algorithme génétique	32
III.3	Bases de données En ligne reconnues	37
III.3.1	Base de donnée latine UNIPEN	37
III.3.2	Base de donnée latine IRONOFF	39
III.3.3	Base de donnée arabe ADAB	39
III.4	Conclusion	40
Chapitre IV :	APPROCHES UTILISÉES ET MODÈLES IMPLÉMENTÉES	41
IV.1.	Introduction	41
IV.2.	Modélisation par des Réseaux Bayésiens Dynamiques	42
IV.2.1.	Introduction	42
IV.2.2.	Modélisation en strokes et leurs relations spatiales	44
IV.2.3	Reconnaissance et Apprentissage	52
IV.2.4.	Amélioration de l'Algorithme de Recherche Dynamique	55
IV.3.	Vérifieur Neuro-génétique	56
IV.3.1	Perceptron Multi Couches	57
IV.3.2	Les Algorithmes Génétiques	63
IV.3.3	Hybridation Neuro-Génétique	71
IV.4.	Conclusion	74
Chapitre V :	RÉSULTATS, ANALYSE ET DISCUSSIONS	75
V.1	Introduction	75
V.2	Dispositifs matériels	75
V.2.2	Ordinateur	75
V.2.3	Tablette PC	75
V.3	Base de données NOUN	79
V.3.1	Fiche d'écriture Scripteur	79
V.2.2	Description de la base	81
V.3.3	Traitement et segmentation des caractères isolés:	81

V.4 Résultats et expériences :	82
V.4.1 Expérience 1 : DBN	82
V.4.4 Expérience 2: DBN profondeur influence	83
V.4.3 Expérience 3 : DBN-NNGA	84
V.4.4 Expérience 4 : DBN Time Segmentation « plage de points »	86
V.4.5 Comparaison et Discussion	87
CONCLUSION GENERALE & PERSPECTIVES	94
BIBLIOGRAPHIE	96
GLOSSAIRE	100

LISTE DES FIGURES

Figure I.1 : <i>Les lettres isolées de l'alphabet arabes</i>	5
Figure I.2 : <i>Sens du d'écriture des lettres arabes</i>	5
Figure I.3 : <i>Quelques styles de Calligraphie arabe</i>	7
Figure I.4: <i>Changement de prononciation de la même forme selon la position des points</i>	8
Figure I.5: <i>Voyelles en rabe : (a) A, (b) OU, (c) I, (d) ~, (e) AN, (f) OUN, (g) IN</i>	8
Figure I.6: <i>Autres signes diacritiques : (a)hamza, (b) chadda, (c) madda</i>	9
Figure I.7 : <i>Les ascendants et descendants sont entourés.</i>	10
Figure I.8 : <i>(a) Le mot « crayon » est composé d'un pseudo-mot, (b) Le mot « école » est composé de 3 pseudo-mots.</i>	11
Figure I.9 : <i>Jointure Conditionnelle</i>	11
Figure I.10 : <i>Ligatures verticales et inversion de l'ordre du tracé</i>	12
Figure II.1: <i>Exemples de systèmes nomades avec leur système de reconnaissance de l'écriture manuscrite.</i>	14
Figure II.2 : <i>Signal en-ligne et signal hors-ligne.</i>	16
Figure II.3 : <i>Illustration du fonctionnement du papier Anoto et du stylo associé.</i>	20
Figure II.4 : <i>systèmes d'acquisition d'écriture en ligne</i>	21
Figure II.5 : <i>Composants de caractère.</i>	23
Figure II.6 : <i>Principe de la reconnaissance d'écriture manuscrite.</i>	24
Figure II.7 : <i>Schéma général d'un système de reconnaissance d'écritures manuscrites</i>	25
Figure III.1 : <i>Réseau de Kohonen</i>	27
Figure III.2 : <i>Schéma synoptique du système proposé de reconnaissance en-ligne de caractères Arabes en utilisant le principe du maximum d'entropie</i>	29
Figure III.3 : <i>La projection du stroke retardé (hamza) dans la lettre Kaf ك</i>	31
Figure III.4 : <i>Le réseau représentant une partie du mot word-part WP</i>	31
Figure IV.1 : <i>Les strokes des caractères.</i>	43
Figure IV.2: <i>Exemple des relations spatiales des strokes.</i>	43
Figure IV.3 : <i>Exemple d'un réseau bayésien</i>	45
Figure IV.4 : <i>Exemples de point et le modèle de point.</i>	47
Figure IV.5 : <i>Relations Spatiales Inter Strokes (RSISs).</i>	48
Figure IV.6 : <i>La construction récursive d'un modèle de stroke.</i>	49
Figure IV.7 : <i>Représentation des relations spatiales inter strokes (ISRs).</i>	50

Figure IV.8 : <i>La représentation par RB d'un modèle de caractère avec N strokes et d=2.</i>	51
Figure IV.9 : <i>Algorithme de recherche dynamique</i>	55
Figure IV.10 : <i>Algorithme de recherche dynamique (ARD) par plage de points</i>	56
Figure IV.11 : <i>Exemple d'un Perceptron Multi Couches</i>	58
Figure IV.12 : <i>Le neurone formel</i>	58
Figure IV.13 : <i>Les fonctions d'activation les plus usitées</i>	60
Figure IV.14 : <i>Convergence vers des erreurs correspondant à des minima locaux</i>	63
Figure IV.15 : <i>Cycle de l'évolution</i>	65
Figure IV.16 : <i>Modèle de roulette pour une population de 6 individus</i>	67
Figure IV.17 : <i>Croisement à un seul locus</i>	68
Figure IV.18 : <i>Croisement à deux locus</i>	68
Figure IV.19 : <i>opérateur de Mutation</i>	68
Figure IV.20 : <i>La représentation génétique d'un PMC</i>	72
Figure IV.21 : <i>Algorithme hybride neuro-génétique détaillé</i>	73
Figure V.1: <i>Fonctionnalités de la tablette WACOM</i>	76
Figure V.2 : <i>Composants de stylets et réglage de paramètres</i>	77
Figure V.3 : <i>Positionnement de stylet</i>	78
Figure V.4 : <i>Zone de travail sur tablette</i>	78
Figure V.5 : <i>Alphabet saisie par le scripteur</i>	79
Figure V.6 : <i>Interface d'acquisition</i>	80
Figure V.7 : <i>Traitement et segmentation des caractères « Ba », « Ain » et « Sad</i>	81
Figure V.8 : <i>Schéma général du modèle implémenté</i>	82
Figure V.9 : <i>histogramme de variation de taux de reconnaissance par rapport la profondeur d</i>	83
Figure V.10 : <i>Vérifieur neuronale</i>	85
Figure V.11 : <i>graphe de l'évolution de l'erreur par itération</i>	85
Figure V.12 : <i>Taux de reconnaissance pour l'ensemble de caractères arabes en utilisant les trois approches d'amélioration implémentées</i>	88
Figure V.13 : <i>Taux de reconnaissance global en utilisant les trois approches d'amélioration implémentées</i>	89

LISTE DES TABLEAUX

Tableau I.1: <i>Les lettres arabes avec leurs positions dans le mot</i>	6
Tableau I.2 : <i>Variabilité des styles d'écriture des points</i>	8
Tableau I.3: <i>Changement du sens du mot « البر » dû à aux trois voyellations</i>	9
Tableau III.1 : <i>L'information du code visuel pour la zone en avant</i>	32
Tableau III.2 : <i>L'information du code visuel pour la zone médiane</i>	33
Tableau III.3 : <i>Étiquettes des Codes Visuels</i>	33
Tableau III.4 : <i>Quelques lettres avec leurs caractéristiques visuelles et le nbr de strokes</i>	34
Tableau III.5 : <i>Exemple de représentation d'un mot arabe</i>	34
Tableau III.6 : <i>Matrice Fitness</i>	35
Tableau III.7 : <i>Opérateur de croisement entre deux mots</i>	36
Tableau III.8 : <i>Résultat de croisement</i>	36
Tableau III.9 : <i>Résultat de mutation pour le mot : علوم</i>	37
Tableau III.10 : <i>Taille de sous-corpus de la base UNIPEN</i>	38
Tableau III.11 : <i>Description de la base de données UNIPEN</i>	38
Tableau III.12 : <i>Taille de sous-corpus de la base IRONOFF</i>	39
Tableau III.13 : <i>sous-corpus de la base de données ADAB 1,2 et 3</i>	40
Tableau V.1 : <i>Paramètres neuronaux</i>	84
Tableau V.2 <i>Paramètre génétique</i>	86
Tableau V.3 : <i>Temps de segmentation des observations pour les trois algorithmes</i>	87
Tableau V.4 : <i>Matrice de confusion pour le modèle DBN</i>	91
Tableau V.5 : <i>Matrice de confusion pour le modèle DBN-NNGA</i>	92
Tableau V.6 : <i>Matrice de confusion pour l'amélioration par ARD Dynamique DBN-SPPD</i>	93

الحمد لله الذي عزّ وجلّ الذي أعطاني القوّة لأخّاز هذا العمل

□ والصلاة والسلام على نبينا محمد ﷺ

□ إلى عائلتي:

□ والديّ العزيزان

□ زوجتي العزيزة و ابنتي الرائعة

□ أختي العزيزة، ابنيها الرائعان وزوجها

□ والديّ زوجتي وأخواتها

□ أصدقائي، جيراني وكلّ من يحبّني

أهدي هذا العمل المتواضع

Je remercie Dieu tout puissant, qui m'a donné la force de concevoir ce travail et que le salut et la bénédiction de dieu soient sur notre prophète Mohamed.

Pour Ma Famille :

Mes chers parents,

Ma chère femme et ma jolie fille,

Ma chère sœur, ses mignons fils et son mari

Mes beaux parents, mes belles sœurs

Mes amis, voisins et tous qui m'aiment.

Je dédie ce modeste travail

Remerciements

Depuis mon introduction dans le monde de la recherche scientifique, j'ai été très chanceux et très honoré d'être membre de son équipe, motivé, grâce à lui, d'attaquer des thématiques récentes. Il a été la source du soutien scientifique et humain dans le laboratoire SIMPA. Il nous a beaucoup appris comment garder l'esprit courageux et la vision claire. Je ne pourrais jamais remercier mon rapporteur et directeur de thèse : Mr le professeur BENYETTOU Abdelkader pour m'avoir initié dans le domaine de la reconnaissance des formes et qui a apporté sa compétence et ses précieux conseils.

Je tiens à adresser mes remerciements les plus chaleureux aux membres du jury qui ont bien voulu s'intéresser à mon travail.

J'adresse toute ma reconnaissance et mes remerciements les plus profonds à Mr le professeur BENYETTOU Mohamed, qui m'a fait l'honneur de présider ce jury.

Je remercie Mr BENNANI Younès, professeur à l'université de Paris 13, Mr BENMOHAMMED Mohamed professeur à l'université de Constantine, Mr HAFFAF Hafid professeur à l'université d'Oran Es-sénia et Mme HADJ-SAID Naïma professeur à l'USTO d'avoir accepté d'examiner ce travail.

Après une coupure de quelques mois, j'ai manqué de courage pour revenir sur le travail de ma thèse et grâce aux encouragements de Mr le professeur BELBACHIR Ahmed Hafid que j'ai pu redémarrer de nouveau ce travail et une aide précieuse de sa part je n'oublierais jamais.

Un remerciement particulier aux : directeur de l'INTTIC, Mr BOUTALEB Abdelmadjid, pour son soutien moral et ses encouragements continus, directeur des études Mr ABDI Mossadek, pour l'accueil chaleureux depuis mes premiers pas dans l'institut.

Je remercie également mes collègues de travail d'INTTIC, surtout les membres du laboratoire LaRATIC ainsi que son directeur : Pr TEMMAR Abdelkader.

Je n'oublierais jamais les bons conseils et les encouragements continus de Mr KOUNINEF Belkacem, ainsi que son équipe dont on dit toujours qu'elle constitue "une grande famille". Je tiens à les remercier pour le réel plaisir que j'éprouve toujours.

Je remercie toute personne ayant contribué et aidé, de proche ou du loin, pour réaliser ce travail.

INTRODUCTION GÉNÉRALE

L'émergence de nouveaux dispositifs de saisie que sont les stylos numériques couplés à des supports papier, permet de produire de documents en-ligne de façon très efficace. De véritables documents peuvent être produits grâce à ces dispositifs, ils peuvent consister aussi bien en des prises de notes, des cours, des copies d'examens, des rédactions d'articles, etc. Cela élargit les champs d'application de la saisie d'écriture en-ligne cantonnés souvent à des terminaux de petites tailles (PDA, smartphone) où seule la reconnaissance des caractères se justifiait.

Établir une interface où n'importe qui peut écrire des mots et les faire identifier en ligne automatiquement donne un avantage certain et une aisance accrue aujourd'hui. En effet, le développement récent des agendas électroniques, l'aspect de nouveaux types de dispositifs tels que les tablettes électroniques et les stylos et l'apparition du logiciel d'encre électronique, augmentent le besoin de ce type de système.

Le développement des terminaux mobiles telles que les Notepad, les téléphones mobiles et les ordinateurs, les livres et la brochure électronique, etc., crée un besoin de nouveaux types d'interfaces. Les constructeurs de ces interfaces ont proposés un certain nombre d'interfaces stylo basées dans la littérature pour ces dispositifs. Toutes ces interfaces permettent à l'utilisateur d'écrire le texte et/ou de commander le terminal, le stylo est employé comme dispositif de choix et cliquant pour actionner des boutons, menus, et choisit des caractères dans un clavier programmable.

Nous sommes intéressés ici dans la conception des systèmes de reconnaissance capables d'apprendre et interpréter efficacement les caractères isolés et de s'adapter très rapidement au besoin spécifique d'utilisateur.

Notre motivation principale dans ce domaine est d'étudier le manuscrit arabe via à vis de la technologie en ligne, autrement dit, le signal d'écriture en ligne acquis à partir d'une

tablette électronique à digitaliser surtout que ceux d'applications d'écriture arabe en ligne ont vu le jour.

Nous abordons aussi le problème de classification et le choix d'un modèle d'apprentissage pour ce type de système car les échantillons ont une forme spatiale, un sens du tracé et donc le choix de l'approche qui modélise mieux la trajectoire n'est pas toujours facile.

Dans ce domaine, nous proposons une nouvelle approche probabiliste : les réseaux bayésiens pour leur capacité d'effectuer des inférences dans un contexte d'incertitude, en quelque sorte comme alternative aux systèmes experts. On va les utiliser aussi pour leurs algorithmes d'apprentissage, comme alternative aux autres méthodes de modélisation quantitative, en les considérant comme des modèles de régression.

Dans ce travail nous réalisons un système de reconnaissance en ligne du manuscrit arabe exposé dans une thèse organisée en cinq chapitres :

Le chapitre 1 présente un résumé qui montre la spécificité de l'écriture arabe et définit aussi le tracé arabe daté de plusieurs siècles.

Le chapitre 2 est dédié à la reconnaissance et l'acquisition d'un signal d'écriture en ligne et son impact ainsi que leurs contextes matériels et logiciels.

Le chapitre 3 est consacré à quelques travaux réalisés dans ce domaine en cernant quelques problématiques posés dans ce sens.

Le chapitre 4 décrit les méthodes, approches et modèles sélectionnés pour mettre le point sur le fondement théorique avant d'exposer les résultats d'expériences réalisées.

Le chapitre 5 détaille l'ensemble des travaux réalisés étape par étape en exposant tous les résultats obtenus pendant le parcours de nos expériences.

Chapitre I : L'ÉCRITURE ARABE

I.1. Introduction

L'arabe, une des six langues officielles des Nations Unies, est la langue maternelle de plus de 300 millions de personnes.

Une hypothèse possible déclare que le manuscrit arabe est évolué à partir du manuscrit araméen Nabatéen.

Il a été employé depuis la 4^{ème} siècle AD, mais le document le plus ancien, une inscription arabe, Syriaque et grec, datés de 512 AD.

La langue Aramaïque a moins de consonnes que l'arabe, ainsi pendant le 7^{ème} nouvelle siècle les lettres arabes ont été créées en ajoutant des points aux lettres existantes pour éviter des ambiguïtés.

D'autres diacritiques indiquant des voyelles courtes ont été présentés, mais ont été seulement employés pour assurer le Coran ont été lus à haute voix sans erreurs.

Il existe deux types principaux de la langue arabe:

1. **L'arabe classique** - la langue du Coran et de la littérature classique. Elle est la langue pure et diffère de l'arabe standard moderne dans le vocabulaire, dont une partie d'elle reste non définie, inconnue, et implicite.
2. **L'arabe standard moderne** - la langue universelle du monde Arabe compréhensible par tous les parlant. En plus de l'arabe pur, elle inclut nouveaux mots : étranger arabisés, scientifiques et technologiques. C'est la langue scolaire et académique de la grande majorité, écrite et regardée via les médias et les conférences, etc. Son vocabulaire est bien déterminé aussi bien qu'explicite.

Historiquement, le Coran a été utilisé intensivement comme source (ou référence) par des linguistes et des grammairiens pour obtenir le vocabulaire, les règles morphologiques, et la grammaire. Il est nécessaire de mentionner l'avantage du Coran contenant la plupart, sinon tout le vocabulaire arabe pur, les règles morphologiques et grammaticales. Cependant, chaque pays ou région parlant arabe a également sa propre variété de l'arabe familier. Ces

variétés familières de l'arabe apparaissent sous la forme écrite dans quelques poésies, dessins animés, et lettres personnelles.

L'arabe, une langue Sémitique, est parlé par environ 300 millions de personnes dans 22 pays, incluant : L'Afghanistan, l'Iran, l'Arabie Saoudite, l'Égypte, et le Maroc. Il y a puis 30 variétés différentes de l'arabe parlé incluent les langues suivantes, selon les degrés de leurs similitudes :

- **Algérien** parlé par 75 millions de personnes en Algérie, en Tunisie, en Libye et le Sahara occidental
- **Égyptien** - parlé par 50 millions de personnes en Égypte.
- **Marocain/Maghrebi** - parlé au Maroc par 25 millions de personnes.
- **Soudanais** - parlé au Soudan par 25 millions de personnes.
- **Saïdi** - parlé par 20 millions de personnes en Égypte.
- **Levantine du nord** - parlé en Palestine, au Liban et en Syrie par 30 millions de personnes.
- **Mésopotamien** - parlé par 55 millions de personnes en Irak, en Iran et en Syrie.
- **Najdi** - parlé en Arabie Saoudite, en Irak, en Jordanie et en Syrie par 35 millions de personnes.

I.2. Alphabet arabe

L'alphabet arabe est l'alphabet utilisé pour écrire, entre autres, la langue arabe. Bien que très souvent désigné comme un alphabet, à la manière de l'écriture d'autres langues sémitiques, c'est généralement un *abjad*, terme décrivant un système d'écriture ne notant que les consonnes de la langue (ou peu s'en faut). Cet alphabet comporte 28 lettres.(figure I.1)

En tant qu'alphabet de la langue du Coran, sacrée pour les musulmans, son influence s'est étendue avec celle de l'islam et il a été aussi utilisé (ou l'est encore) pour écrire d'autres langues qui n'ont aucune parenté avec l'arabe, comme le persan, le turc (avant 1928, date à partir de laquelle Mustafa Kemal Atatürk a imposé la transcription latine), le kashmiri, le sindhi, ou encore l'ourdou et le kurde (toutes ces langues, d'ailleurs, sauf le turc qui est une

langue altaïque, étant indo-européennes). On a souvent dû ajouter ou modifier certaines lettres pour adapter cet alphabet au système phonologique des langues en question.

Certaines langues d'Afrique, comme l'haoussa et le somali, s'écrivirent par des adaptations de l'alphabet arabe avant d'être écrites avec l'alphabet latin.

ح	ج	ث	ت	ب	ا
ha	jim	tha	ta	ba	alif
س	ز	ر	ذ	د	خ
sin	za	ra	zal	dal	kha
ع	ظ	ط	ض	ص	ش
'ain	za	ta	dhad	sad	shin
م	ل	ك	ق	ف	غ
mim	lam	kaf	qaf	fa	ghain
ي	ء	ه	و	ن	
ya	hamzah	ha	waw	nun	

Figure I.1 : Les lettres isolées de l'alphabet arabes

L'arabe s'écrit de droite à gauche avec un sens du tracé respecté (voir figure I.2) et la construction d'un mot se fait en collant les lettres une à une, de droite à gauche. Mais chaque lettre de l'alphabet s'écrit différemment selon sa position dans le mot. Les notions de lettre majuscule et lettre minuscule n'existent pas (l'écriture est donc monocamérale).

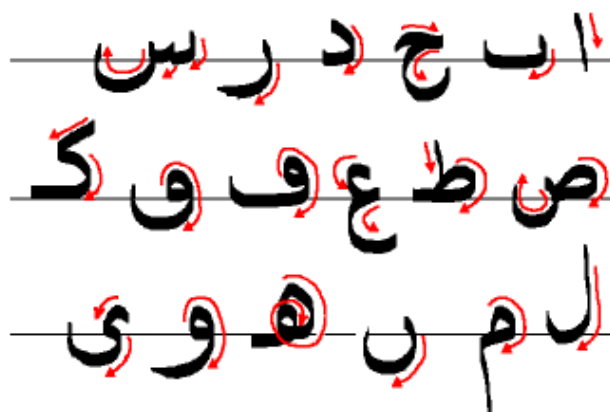


Figure I.2 : Sens du d'écriture des lettres arabes

I.3. Caractéristiques de l'écriture Arabe

L'alphabet Arabe est composé de 28 lettres dont la forme change selon la position dans le mot. Les lettres s'écrivent différemment selon qu'elles sont isolées, au début, au milieu ou à la fin du mot (voir tableau I.1). [1]

Pour la plupart des lettres, les formes début/milieu et fin/isolé sont identiques à la ligature près. La présence d'une ligature avec la lettre précédente ou avec la lettre suivante ne modifie pas la forme de la lettre de manière significative (pas plus que dans l'écriture manuscrite cursive latine). En arabe, les ligatures se situent toujours au niveau de la ligne d'écriture, c'est-à-dire qu'il n'existe pas de lettre à liaison haute comme le 'o' ou le 'v' en alphabet latin.

Le plus grand alphabet aujourd'hui contient maintenant environ 100 formes possibles.

Donc, il existe une influence du scripteur sur les modèles d'écriture arabes et de divers modèles calligraphiques, certains sont montrés sur la figure I.3.

Tableau I.1: Les lettres arabes avec leurs positions dans le mot

lettre	nom	fin	milieu	début	lettre	nom	fin	milieu	début
ا	alif	آ	ـا	ا	ط	Ṭa	ظ	ظ	ظ
ب	ba	ب	ـب	ب	ظ	Za	ظ	ظ	ظ
ت	ta	ت	ـت	ت	ع	ʿayn	ع	ع	ع
ث	tha	ث	ـث	ث	غ	ghayn	غ	غ	غ
ج	jim	ج	ـج	ج	ف	fa	ف	ف	ف
ح	Ḥa	ح	ـح	ح	ق	qaf	ق	ق	ق
خ	kha	خ	ـخ	خ	ك	kaf	ك	ك	ك
د	dal	د	ـد	د	ل	lam	ل	ل	ل
ذ	dhal	ذ	ـذ	ذ	م	mim	م	م	م
ر	ra	ر	ـر	ر	ن	nun	ن	ن	ن
ز	zay	ز	ـز	ز	ه	ha	ه	ه	ه
س	sin	س	ـس	س	و	waw	و	و	و
ش	shin	ش	ـش	ش	ي	ya	ي	ي	ي
ص	Ṣad	ص	ـص	ص	ء	hamza	أ و إ ئ		
ض	Ḍad	ض	ـض	ض					



Figure I.3 : *Quelques styles de Calligraphie arabe*

I.3.1 Signes diacritiques

L'écriture Arabe est riche en diacritique d'une façon générale, et plus particulièrement en points. Le terme 'signe diacritique' peut porter à confusion : dans certains travaux, seules les voyelles arabes sont appelées diacritiques. Dans d'autres travaux, en revanche, tous les signes secondaires sont appelés diacritiques, qu'il s'agisse des voyelles, des points ou des autres signes (chadda, madda, hamza, ...). C'est cette deuxième terminologie que nous employons ici : un signe diacritique est une composante secondaire d'une lettre, qui vient la compléter ou en modifier le sens.

Dans cette thèse, les "signes diacritiques" désigneront à la fois points, voyelles et autres signes secondaires.

I.3.1.1 Les points :

Nous comptons 15 lettres, parmi les 28 de l'alphabet qui comportent des points. Ces points apparaissent au dessus ou en dessous du caractère uniquement. Le nombre maximal de points que peut avoir une lettre est de trois points au-dessus du caractère, ou deux points

en-dessous. Ces points permettent de différencier la prononciation des lettres Arabes comme montrée dans la figure I.4.

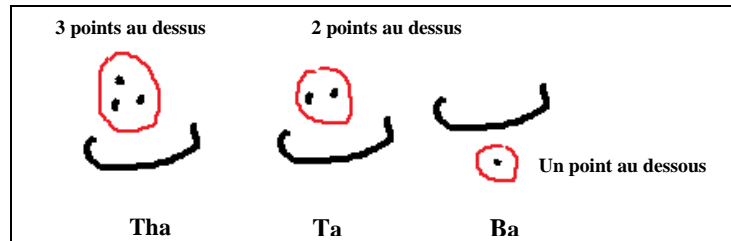


Figure I.4: *Changement de prononciation de la même forme selon la position des points*

Le tableau I.2 illustre la variabilité des styles d'écriture des points ou groupes de points en écriture manuscrite arabe. Un groupe de deux points peut ainsi s'écrire sous forme d'une seule, ou de deux composantes connexes. On remarque la très forte similarité entre deux points reliés par un trait, et une voyelle de type 'A' ou 'I' dont les exemples sont donnés figure I.4. Un groupe de trois points peut donner lieu à une, deux ou trois composantes connexes, en fonction du style d'écriture.[2]

Tableau I.2 : *Variabilité des styles d'écriture des points*

un point	.	˙	˘	
2 points	ˆ	ˆ	ˆ	ˆ
3 points	ˆ	ˆ	ˆ	ˆ

I.3.1.2 Les voyelles :

En arabe, les voyelles ne sont pas seulement des lettres, mais aussi des signes diacritiques associés aux lettres sur lesquelles ils s'appliquent (figure I.5).

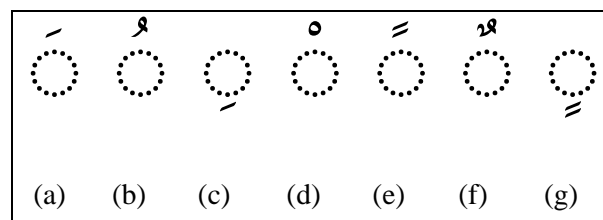


Figure I.5: *Voyelles en arabe : (a) A, (b) OU, (c) I, (d) ~, (e) AN, (f) OUN, (g) IN*

En général on ne représente pas les voyelles signes, sauf dans les manuels scolaires.

L'absence de voyelles peut toutefois être source de confusions. Un mot peut avoir plusieurs voyellations possibles et par conséquent plusieurs catégories grammaticales.

Par exemple comme mentionnés dans la table I.3, le mot « البر » peut prendre trois sens en jouant sur trois voyellations possibles.

Tableau I.3: *Changement du sens du mot « البر » dû à aux trois voyellations*

Le mot	Le sens
الْبَر	La lithosphère
الْبُر	Le blé
الْبِر	La bienfaisance

Les voyelles peuvent parfois être mentionnées sur certaines lettres pour lever l'ambiguïté et faciliter la lecture. Mais en général, les scripteurs les omettent purement et simplement, et c'est au lecteur qu'est réservé le soin d'interpréter correctement le sens de la phrase en fonction du contexte.

I.3.1.3 Les autres signes diacritiques :

Les autres signes diacritiques sont la *hamza*, la *chaddet* la *madda*, (voir figure I.6). La chadda est une accentuation de la lettre (c'est l'équivalent d'une consonne doublée). Hamza et madda suivent des contraintes morphosyntaxiques plus complexes.[2]

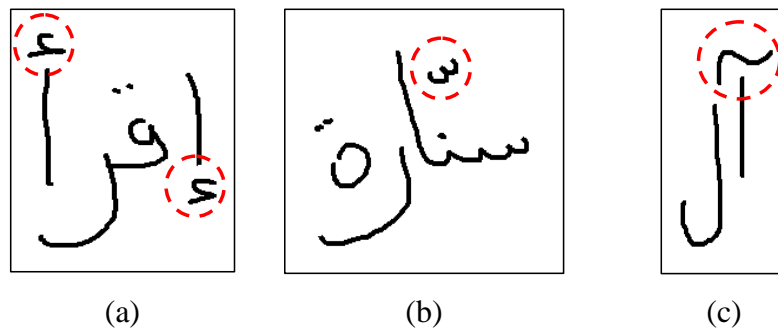


Figure I.6: *Autres signes diacritiques : (a) hamza, (b) chadda, (c) madda*

I.3.2 Ascendants et descendants

Comme dans l'écriture latine, l'écriture arabe contient des ascendants et des descendants (voir figure I.7). En arabe, les descendants peuvent se prolonger horizontalement sous la bande de base, ce qui introduit une superposition verticale entre la lettre qui comprend le descendant et la lettre suivante.[2]

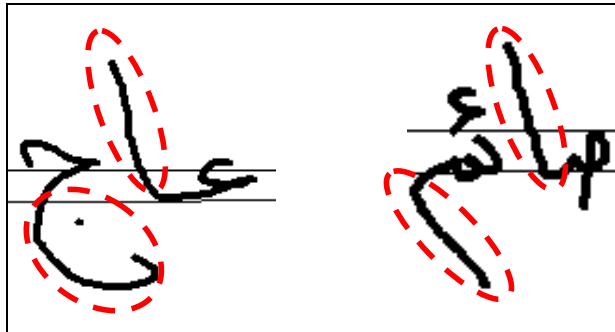


Figure I.7 : *Les ascendants et descendants sont entourés.*

I.3.3 Une ou plusieurs composantes connexes par mot

L'écriture arabe est semi-cursive dans ses deux formes, imprimée et manuscrite. En effet, un mot arabe est une séquence d'entités connexes entièrement séparées appelées *pseudo-mot*. Un mot peut être composé d'un ou de plusieurs pseudo-mots ; ceci est dû à la présence de caractères qui ne peuvent pas être attachés à leur successeur. Chaque pseudo-mot est une séquence de lettres liées, ce qui donne l'aspect de cursivité à cette écriture. Notons qu'un caractère isolé peut constituer un pseudo-mot à lui seul, comme le montre la figure I.8.

Six lettres ne sont pas liées à leur successeur : و, ز, ر, ذ, د et ا. Ces lettres introduisent donc une coupure dans le mot.

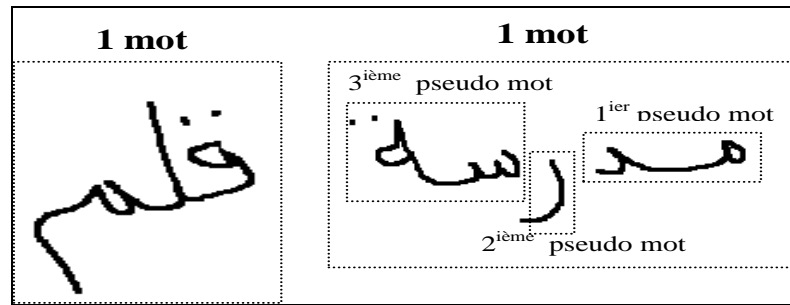


Figure I.8 : (a) Le mot « crayon » est composé d'un pseudo-mot, (b) Le mot « école » est composé de 3 pseudo-mots.

En manuscrit, l'espacement entre les différents pseudo-mots d'un même mot n'est pas forcément systématiquement supérieur à l'espacement entre deux mots différents, ce qui pose parfois des problèmes de segmentation. Lorsqu'une des 22 lettres (28 moins les 6 qui ne se lient pas avec la suivante) apparaît dans sa forme "fin de mot" ou "isolée", cela signifie obligatoirement que l'on arrive à la fin d'un mot.

Certaines lettres des 22, présentent un phénomène appelé : *jointure conditionnelle* où la lettre changera sa forme à la fin du mot. (voir figure I.9)

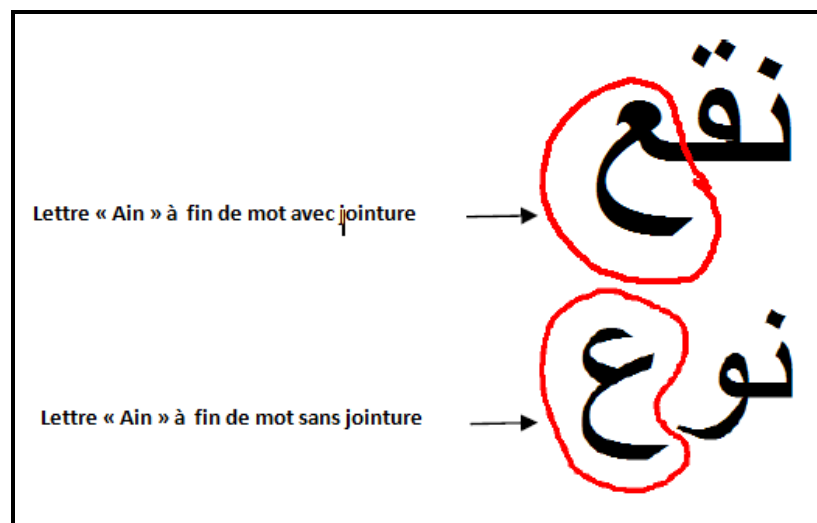


Figure I.9 : Jointure Conditionnelle

I.3.4 Ligatures verticales

En écriture arabe, il n'y a pas de liaisons hautes comme le 'v' ou le 'o' en latin : les ligatures se situent au niveau de la ligne support de l'écriture (ligne de base). En revanche, les scribes sont libres de constituer certains groupes de deux ou trois lettres liées verticalement en début de pseudo-mot. Ce sont les ligatures verticales (figures I.10). En général très complexes à segmenter, nous choisissons de les reconnaître telles quelles.[1]

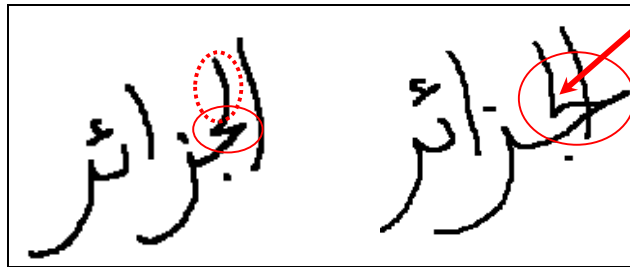


Figure I.10 : *Ligatures verticales et inversion de l'ordre du tracé*

I.4 Conclusion

L'écriture arabe est une écriture cursive qui présente l'avantage d'une liaison favorisant la fluidité et la rapidité du geste.

Mais une des premières difficultés est la variabilité d'écriture entre scribes. La difficulté surmontée pour le traitement du caractère est sa forme et sa position dans le mot. La reconnaissance de l'écriture arabe manuscrite est en soi une tâche qui est fondamentalement plus complexe que la reconnaissance de l'écriture latine manuscrite cursive à cause des spécificités de l'écriture arabe qui présente un certain nombre de difficultés supplémentaires.

Chapitre II : RECONNAISSANCE D'UN SIGNAL D'ÉCRITURE EN LIGNE

II.1 Introduction

Les moyens essentiels pour se communiquer entre les personnes sont la parole et l'écriture manuscrite. Pour faciliter l'interaction homme-machine, on a créé des systèmes pour les reconnaître.

Donc, un système de reconnaissance d'écriture manuscrite va transformer la représentation spatiale en celle symbolique.

II.2 Reconnaissance Manuscrite

5000 ans après son invention, 550 ans après son automatisation, l'écriture est toujours au cœur des communications entre les hommes. A l'heure des interactions entre l'humain et la machine toujours plus sophistiquées et performantes à l'aide de " boutons", de microphones ou de cameras, il est naturel de chercher à comprendre automatiquement l'écriture. Depuis les premières tentatives, les systèmes de lecture des adresses pour le traitement automatique du courrier ou de lecture des chèques ont connu d'importants développements et sont maintenant largement utilisés. Toutefois, la compréhension de l'écriture par un ordinateur est encore loin d'être pleinement satisfaisante. La raison est liée au fait que l'étude de la reconnaissance de l'écriture est un domaine très vaste tant par ses applications que par ses techniques.

Donc, la reconnaissance de l'écriture manuscrite n'est pas un sujet nouveau, il remonte à plus d'une trentaine d'années. La reconnaissance de l'écriture manuscrite s'avère un problème extrêmement complexe qui n'a pas de solution satisfaisante à ce jour. Ainsi rapidement, les chercheurs ont restreint leurs études à des problèmes particuliers en liaison avec des applications bien définies. Parmi celles-ci on cite les applications de la reconnaissance de l'écriture manuscrite : le tri automatique du courrier postal, le traitement automatique de dossiers administratifs, des formulaires d'enquêtes, ou encore l'enregistrement des chèques bancaires.

L'émergence de l'informatique nomade a rendu indispensable l'utilisation des interfaces orientées stylo : l'utilisateur écrit et dessine directement des textes, des croquis, des commandes, etc sur l'écran tactile du système, l'interface classique clavier/souris étant à la fois plus encombrante et moins adaptée à l'utilisation de ces périphériques. Ce type d'interface a de nombreux avantages dont celui d'être accessible dans de nombreuses situations (nécessité de déplacement, bruit, station debout...) dans lesquelles les utilisateurs peuvent se trouver (contrôle de chantier, médecin à l'hôpital...). De plus les interfaces stylo permettent de réduire la taille du périphérique puisque le clavier n'est plus nécessaire. La figure II.1 présente deux types de périphériques mobiles avec chacun une interface différente pour la reconnaissance de l'écriture. [3]



Figure II.1: Exemples de systèmes nomades avec leur système de reconnaissance de l'écriture manuscrite.

La reconnaissance de l'écriture manuscrite est un traitement informatique qui a pour but de traduire un texte écrit en un texte codé numériquement.

Il faut distinguer deux reconnaissances distinctes, avec des problématiques et des solutions différentes :

- Reconnaissance d'écriture manuscrite hors-ligne
- Reconnaissance d'écriture manuscrite en-ligne.

II.2.1 Reconnaissance Hors ligne

Dans les applications de reconnaissance d'écriture manuscrite hors-ligne, l'écriture de l'utilisateur est acquise par un scanner. L'entrée de ces systèmes est une image. Le

problème majeur est comment on peut enlever le bruit comme des éléments du fond d'image, comment on peut traiter le manque des traits etc.

D'ailleurs, pour être utilisés largement, ces systèmes doivent traiter rapidement des images avec le taux de reconnaissance discutés et estimé élevés. Applications typiques sont classification automatique des lettres, la lecture des chèques de banque etc.

II.2.2 Reconnaissance En Ligne

Dans le cadre de la reconnaissance en-ligne, l'échantillon d'encre est constitué d'un ensemble de coordonnées ordonnées dans le temps. Il est ainsi possible de suivre le tracé, de connaître les posés et levés de stylo et éventuellement l'inclinaison et la vitesse. Il faut évidemment un matériel spécifique pour saisir un tel échantillon, c'est le cas notamment des stylos numériques ou des stylets sur agendas électroniques ou sur les *Tablets PC*.

La reconnaissance en-ligne est généralement beaucoup plus efficace que la reconnaissance hors-ligne car les échantillons sont beaucoup plus informatifs. En revanche, elle nécessite un matériel beaucoup plus coûteux et impose de fortes contraintes au scripteur puisque la capture de l'encre doit se faire au moment de la saisie (capture synchrone) et non *a posteriori* (capture asynchrone). (Voir figure II.2).

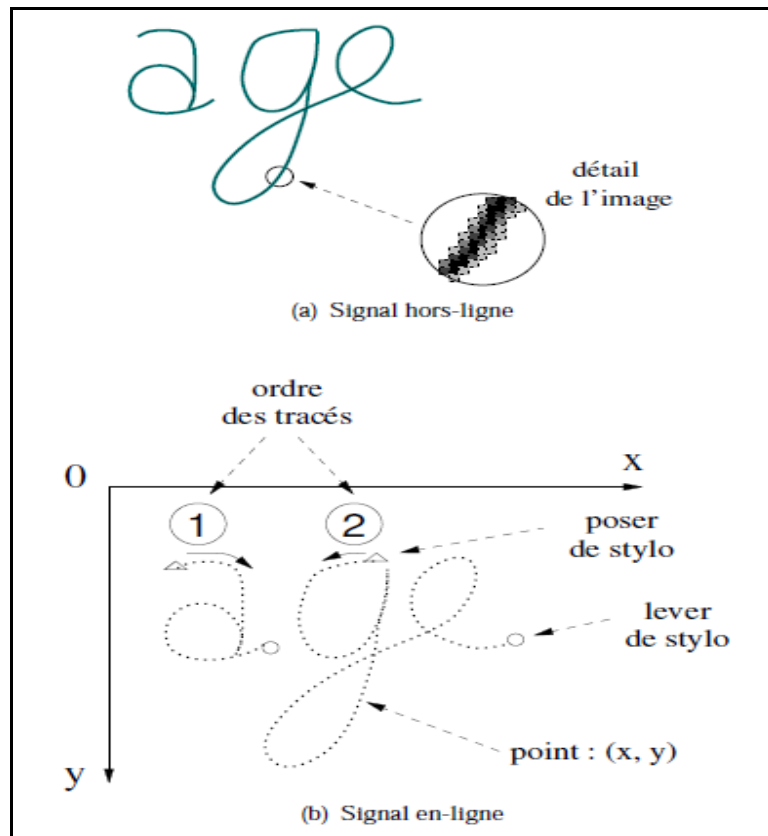


Figure II.2 : Signal en-ligne et signal hors-ligne.

II.3 Problématique générale de la reconnaissance d'écriture

Contrairement aux documents papier qui sont numérisés sous forme d'images, les documents saisis en-ligne (et plus spécifiquement les caractères) sont stockés sous forme d'encre électronique.

Les documents en-ligne peuvent être saisis en utilisant plusieurs types de périphériques :

- la souris d'un ordinateur de bureau ;
- une tablette graphique sans retour visuel ;
- un écran tactile sur lequel on dessine avec le doigt ou un crayon passif (guichet automatique, PDA, Smartphone...) ;
- un écran sensitif utilisé avec un stylo spécifique (Tablet PC...) ;
- un stylo enregistrant sa position absolue (comme le stylo Anoto utilisant une caméra sur un papier spécifique);

Ces différents périphériques font que l'encre électronique enregistrée pourra être de nature et de qualité différente. Les domaines applicatifs sont tout aussi vastes, de l'écriture de textes à la saisie de schémas en passant par le remplissage de formulaires ou les gestes d'édition de documents.

Mais la principale difficulté rencontrée lors de la reconnaissance de l'écriture manuscrite est la variabilité des styles d'écriture. En effet, la forme des caractères manuscrits varie énormément d'un scripteur à l'autre et même pour un scripteur donné suivant le contexte du caractère (sa position dans le mot, les lettres voisines...). Cette variabilité est même source d'ambiguïté entre les caractères puisque un même tracé peut avoir différentes significations suivant le contexte ou selon le scripteur. Ces propriétés font que la reconnaissance de l'écriture manuscrite est un domaine applicatif pour la reconnaissance de formes très riche en difficultés et en défis.

Reconnaître de l'écriture manuscrite consiste à associer une représentation symbolique à une séquence de symboles graphiques : on parle aussi de lecture automatique. Le but est de pouvoir utiliser cette représentation dans une application informatique. On distingue deux grands types d'utilisation :

- traiter automatiquement des documents contenant de l'écriture manuscrite dont l'analyse par des individus prend trop de temps
- faciliter l'utilisation des ordinateurs pour des applications où un stylo est plus pratique qu'un clavier et une souris.

Du point de vue informatique, les difficultés de la reconnaissance d'écriture sont liées bien évidemment à la variabilité des écritures, mais aussi au fait qu'on ne sait pas encore très bien modéliser l'énorme masse de connaissances et d'informations utiles à la reconnaissance.

II.4 Quelques exemples d'applications

Voici quelques exemples d'applications informatiques utilisant des systèmes de reconnaissance d'écriture :[4]

1. Prendre des notes avec un stylo électronique directement sur l'écran de son PDA ou Tablet PC et pouvoir les éditer ensuite avec un logiciel de traitement de texte.
2. Rédiger des messages ou des mails à l'aide d'un stylo sur l'écran de son Smartphone.
3. Saisir des formules mathématiques sur l'écran de son Tablet PC.
4. Rétro-convertir des documents papiers : par exemple pour numériser à l'aide d'un scanner un document manuscrit et le transformer en document éditable avec un logiciel de traitement de texte.
5. Indexer des documents d'archives : ce sont souvent des documents abîmés et peu accessibles au public, les numériser et reconnaître certaines parties du document (par exemple le nom sur une fiche d'état civil) permet de faciliter les recherches.
6. Trier automatiquement le courrier en analysant automatiquement des adresses postales sur des enveloppes.
7. Traiter des formulaires remplis à la main : comme par exemple un sondage, un bon de commande ou un cheque.

Ces différents exemples correspondent à l'analyse de documents plus ou moins complexes contenant de l'écriture manuscrite. Suivant les cas, la reconnaissance consiste à reconnaître des documents complets, des mots ou seulement des caractères dans une partie des documents.

II.5 Contexte de l'écriture en-ligne

La reconnaissance d'écriture manuscrite en-ligne implique la conversion automatique du texte pendant qu'on lui écrit sur un spécial tablette électronique ou PDA (Personal Digital Assistant), où on prend les mouvements de stylo $X(t)$, $Y(t)$ aussi bien que la commutation de lever/poser de stylo (pen-up/pen-down).

Que le genre de données est connu en tant qu'encre numérique et peut être considéré comme représentation dynamique d'écriture. Le signal obtenu est converti en codes de lettre qui sont utilisables dans l'ordinateur et les applications de traitement de texte. Les éléments d'une interface en ligne de reconnaissance d'écriture incluent typiquement :

- Un stylo ou une aiguille pour que l'utilisateur écrive avec.
- Une surface sensible de contact, avec laquelle peut être intégré, ou un affichage de rendement.
- Une application de logiciel qui interprète les mouvements de l'aiguille à travers la surface d'écriture, traduisant la courbe de résultat dans le texte numérique.

Trois propriétés caractérisent la reconnaissance en ligne: la notion d'ordre d'écriture (enchaînement temporel de traits), la dynamique du tracé (vitesse, accélération, lever du stylo) et squelette du tracé (aucune épaisseur de trait).

II.5.1 Contexte matériel et les interfaces Stylo

Différents matériels de saisie ont été proposés, des stylos et des tablettes. Les premiers stylos électroniques permettaient de capturer leur position par rapport à une tablette, les positions successives étant envoyées au fur et à mesure à un terminal. Les stylos électroniques plus récents, e.g. SmartQuill, C-Pen ou Smartpen, sont équipés de capteurs leur permettant de mesurer des caractéristiques comme l'inclinaison, la vitesse de déplacement ou la pression du stylo, ce qui permet notamment de reconstituer un tracé indépendamment d'un support quelconque. Ils disposent de plus d'une capacité de stockage qui leur confère une certaine autonomie. Les tablettes sont utilisées depuis assez longtemps en CAO, elles sont sensibles au contact d'un stylo spécial et capturent les positions successives de ce stylo. Assez récemment d'autres types de support sont apparus. D'une part Anoto4 a proposé des feuilles de papier tout à fait semblables aux feuilles habituelles mais sur lesquelles sont dessinées très finement (invisibles à l'oeil nu) de très nombreux points formant des motifs (voir figure.II.3).

Ce papier doit être utilisé avec un stylo spécial, équipé d'une micro-caméra qui lorsque l'on écrit, interprète le motif visible et en déduit la position du stylo sur la feuille. Ainsi, pendant que l'on écrit. Normalement avec ce stylo le signal est mémorisé dans le stylo et peut être "déchargé" sur un ordinateur personnel sous forme de documents de notes manuscrites. D'autre part, même s'il ne s'agit que d'un dispositif d'affichage et non de saisie, on peut citer ici un papier électronique mis au point par plusieurs équipes dont Xerox et le MediaLab de MIT, sensiblement plus épais qu'une feuille, et sur lequel on peut afficher ce

que l'on veut, comme sur un écran par polarisation de billes assimilables à des pixels d'un écran.[5]

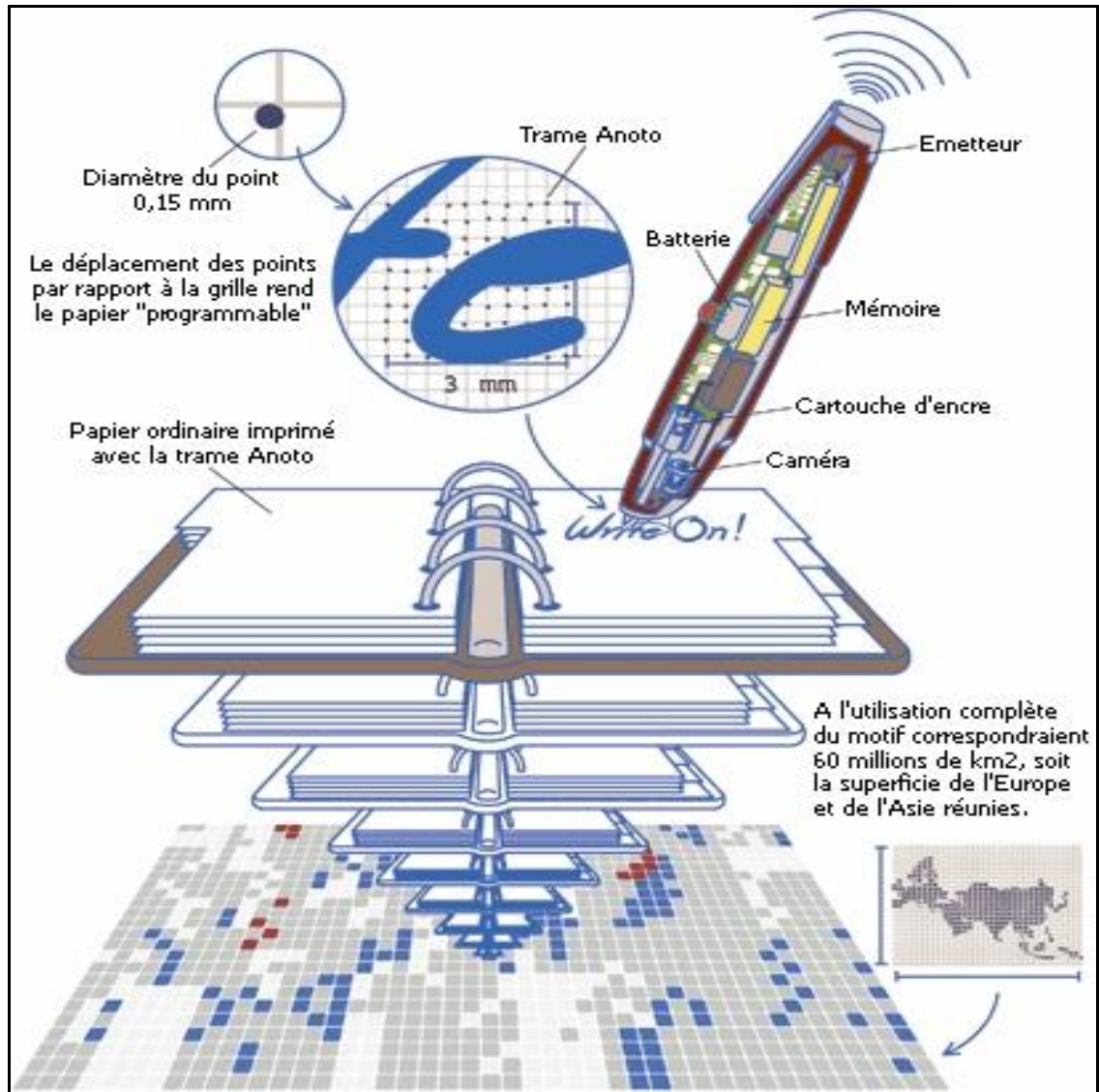


Figure II.3 : Illustration du fonctionnement du papier *Anoto* et du stylo associé.

La figure suivante présente différents outils et supports d'acquisition : [6]



Figure II.4 : systèmes d'acquisition d'écriture en ligne

II.5.2 Contexte logiciel

Les premiers logiciels de reconnaissance manuscrite commerciales ont été intégrés a des organisateurs électroniques muni d'un stylet permettant la saisie de caractères voire de mots et phrases aujourd'hui et parfois muni de clavier. En dehors de ce marché important des petits assistants personnels, d'autres applications se sont développées a partir de tablette graphique et récemment des stylos digitaux. Dans le milieu médical, pour la saisie et stockage d'information auprès du lit des malades, pour la saisie d'ordonnance médicale.

Dans le monde de l'éducation, pour seconder l'enseignement dans sa tache d'apprentissage de l'écriture qui ne peut surveiller qu'un enfant a la fois pendant son geste de production d'écriture. Autant dans le milieu scolaire que médical, pour détecter rapidement les différentes causes liées aux troubles psychiques et moteurs (parkinson, sclérose...) et aux échecs scolaires. Et dans l'univers des réunions, avec les possibilités de prise de notes, annotations, conservations de toutes traces écrites et orales des différents intervenants de la réunion.

Avec le développement de nouveaux terminaux mobiles de types variés, la naissance de l'encre électronique, des nouveaux besoins sont apparus. D'une part, pour gérer des documents Encre, il faut être capable de les éditer, de les segmenter, de les indexer pour pouvoir faire de la recherche d'informations. D'autre part, le développement de systèmes de reconnaissance doit prendre en compte les caractéristiques de ces terminaux. Il s'agit pour la plupart de terminaux personnels (agendas, portables, cahier), pour lesquels le stylo est utilisé à la fois pour entrer du texte, mais également des dessins, des gestes de commande etc. Les systèmes de reconnaissance adaptés aux différents objets mobiles et à leurs applications sont très variés.

Le but de ce contexte est de montrer que, si de nombreuses recherches ont été menées depuis une dizaine d'années sur la reconnaissance de l'écriture en-ligne, aboutissant à des systèmes assez performants pour la reconnaissance de caractères ou de mots, la pertinence de ces systèmes pour le développement d'interfaces homme-machine n'est pas démontrée dans le sens où, par exemple, un utilisateur courant d'ordinateurs personnels écrit plus vite au clavier qu'avec un stylo. De la même façon que les systèmes de reconnaissance de la parole, malgré leurs performances, n'ont pas remplacé l'interface traditionnelle clavier/souris, le stylo électronique n'est pas non plus nécessairement voué à devenir notre interface privilégiée pour entrer du texte dans une machine. Malgré cela, les systèmes de reconnaissance peuvent trouver leur place dans le monde des interfaces stylo.

La reconnaissance de l'écriture étant un problème complexe, les moteurs actuels, même performants, commettent des erreurs en reconnaissance et sont du coup plus ou moins acceptés par un utilisateur. D'autre part ces systèmes sont en règle générale assez consommateurs de ressources machine. Ceci a conduit au développement de techniques peu coûteuses algorithmiquement et permettant d'entrer du texte dans une machine, à l'aide d'un stylo, mais avec des taux d'erreurs suffisamment faibles pour être acceptables par l'utilisateur. Ces systèmes sont en général livrés avec des objets mobiles de petite ou très petite taille du type agendas électroniques, disposant de peu de ressources machine.

II.5.3 Axes de recherche autour de la reconnaissance en-ligne

Les principaux axes de recherche actuels autour de la reconnaissance en-ligne concernent :

- la reconnaissance de caractère, de mots, de phrases, de textes en utilisant des connaissances contextuelles (spécifiques au document, linguistiques, etc.)
- l'adaptation automatique à l'écriture d'un scripteur à partir d'un système de reconnaissance générique.
- la présentation des résultats de reconnaissance, l'ergonomie de l'interface stylo, l'édition d'un document.
- les outils d'éducation d'aide à l'apprentissage de l'écriture, la détection de troubles liés à l'écriture
- l'authentification du scripteur, la reconnaissance de signatures.

L'axe dans lequel se situent nos travaux est le premier : la reconnaissance de caractères isolés d'écriture manuscrite en-ligne.

II.6 Composants de caractère

Un caractère se compose hiérarchiquement de composants qui ont des relations spatiales propres. Par exemple, le manuscrit anglais a des mots, des alphabets, des *strokes* et des *points*. La figure II.5(a) montre la hiérarchie du manuscrit anglais (b) montre des exemples des composants. Un composant se compose de sous composants à un niveau plus bas et à leurs relations. Le composant le plus primitif, un point, est représenté par sa position de X et de Y dans un plan de deux dimensions. [7][8]

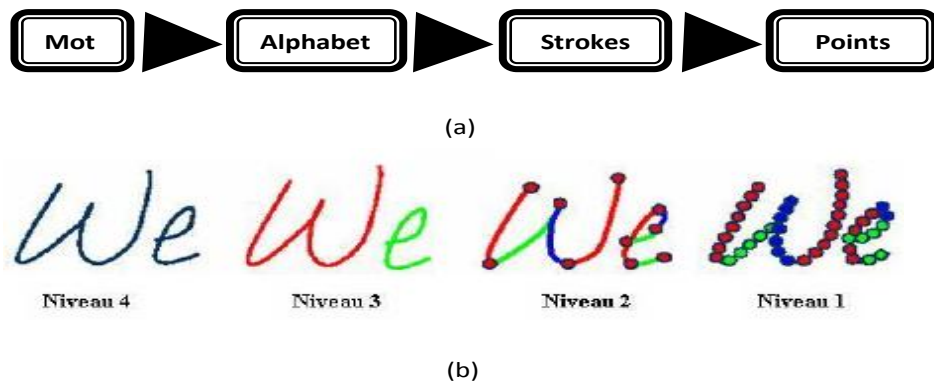


Figure II.5 : Composants de caractère.
 (a) Les composants hiérarchiques
 (b) Exemples des composants des manuscrits anglais.

II.7 Système de reconnaissance d'écriture manuscrite en ligne

Comme déjà expliqué, la reconnaissance d'écriture manuscrite consiste à transformer un signal numérique en sa représentation symbolique (voir figure II.6).[9]

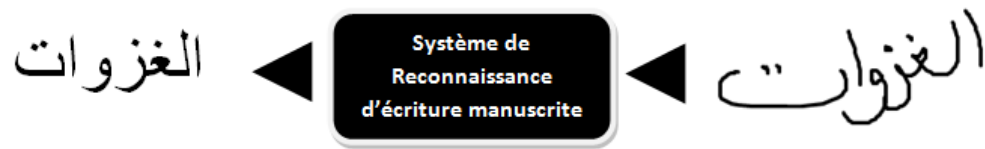


Figure II.6 : Principe de la reconnaissance d'écriture manuscrite.

La reconnaissance d'écriture manuscrite est une tâche complexe, à-cause de la variabilité des styles d'écriture. [10]

L'utilisation d'un système de reconnaissance d'écriture en ligne ne doit pas être pénible ou encombrante puisqu'ils sont souvent employés en tant qu'élément du calcul dominant mobile.

Les schémas de reconnaissance sont pour autant globalement communs et se déclinent en quatre notions clés: (voir figure II.7) :

1. Acquisition et stockage de données
2. Prétraitement
3. Extraction des caractéristiques
4. La reconnaissance

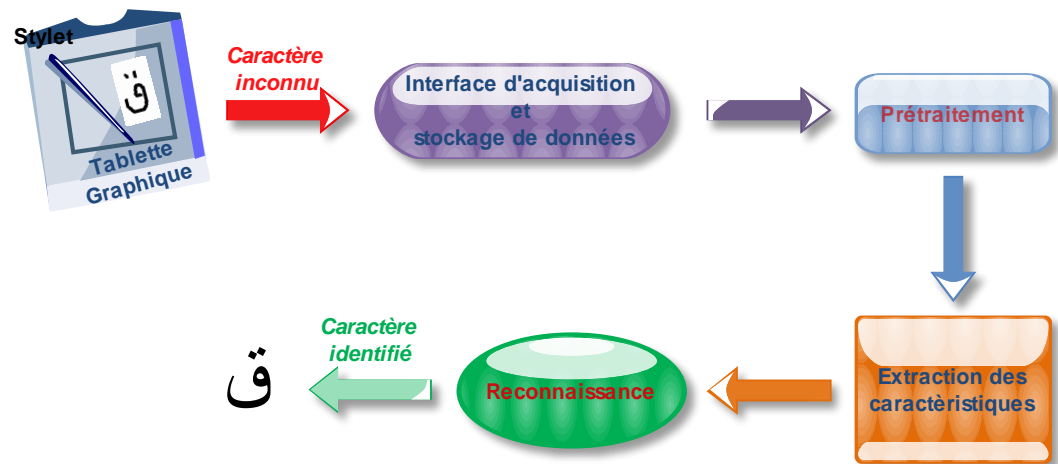


Figure II.7 : Schéma général d'un système de reconnaissance d'écritures manuscrites

II.8 Conclusion

Cette thèse se place dans le cadre de la reconnaissance d'écriture manuscrite et plus particulièrement dans celui de la reconnaissance de caractères isolés, saisis en-ligne sur des périphériques tels que des Tablet PC.

La principale différence entre les deux domaines de l'écriture manuscrite en ligne et hors-ligne, souvent traités séparément, réside dans la nature des données à traiter (temporelles ou spatiales) et de l'information pertinente que l'on peut extraire dans un but de reconnaissance.

Chapitre III : ÉTAT DE L'ART DES TRAVAUX RÉALISÉS

III.1 Introduction

Ces dix dernières années, beaucoup de travaux ont été réalisés basés sur l'écriture en ligne sur différentes type du manuscrits (latins, chinois, arabes et d'autres) et en utilisant plusieurs approches de reconnaissances. On distingue, parmi ces travaux :

- Le travail d'E. Poisson [11] basé sur un système de reconnaissance en ligne hybride Neuro-Markovien traitant le manuscrit latin.
- Un système d'inférence floue appliqué à la reconnaissance des caractères chinois en ligne a été proposé par A. Delaye et al. [12]
- R. Saabni et al ont traité la reconnaissance du manuscrit arabe en ligne par une approche hiérarchique détaillé dans [13].
- Il y a d'autres travaux traitant le manuscrit arabe en ligne ancien comme le travail de A. Elbaati [14] qui a réalisé un système permettant le passage de l'hors ligne vers l'en ligne.
- N. Mezghani [15] [16] et al, F. Biadsy et al [17], ont réalisé un système de reconnaissance de caractères arabe isolés en ligne. Par contre M. Kherallah et al [18] ont réalisé un système de reconnaissance de mots basé sur les codes visuels et une approche génétique.

Notre axe de recherche est bien l'écriture arabe en ligne. Pour cela, dans ce qui suit, on va éclaircir quelques travaux réalisés dans ce sens.

III.2 Synthèse des travaux du manuscrit Arabe En ligne

III.2.1 Reconnaissance de caractères Arabes en-ligne par réseau de *Kohonen*

Le travail de [16] a été réalisé pour but de développer une mémoire de *Kohonen* pour la reconnaissance en-ligne de caractères arabes manuscrits. La mémoire de *Kohonen* s'obtient par un apprentissage non supervisé, elle présente des propriétés attractives telles que l'ordonnement et le pouvoir de généralisation. Les caractéristiques sont extraites à partir de la représentation dynamique des caractères à travers les signaux $x(t)$ et $y(t)$, qui correspondent à la position du stylo électronique sur la tablette au temps t .

Le réseau de *Kohonen* est composé de deux couches; la première constitue l'entrée du réseau et la seconde est la sortie.

Le réseau réalise une liaison entre une entrée à I composantes et un ensemble de J sorties par l'intermédiaire de $I \times J$ coefficients aussi appelés poids (voir figure III.1).

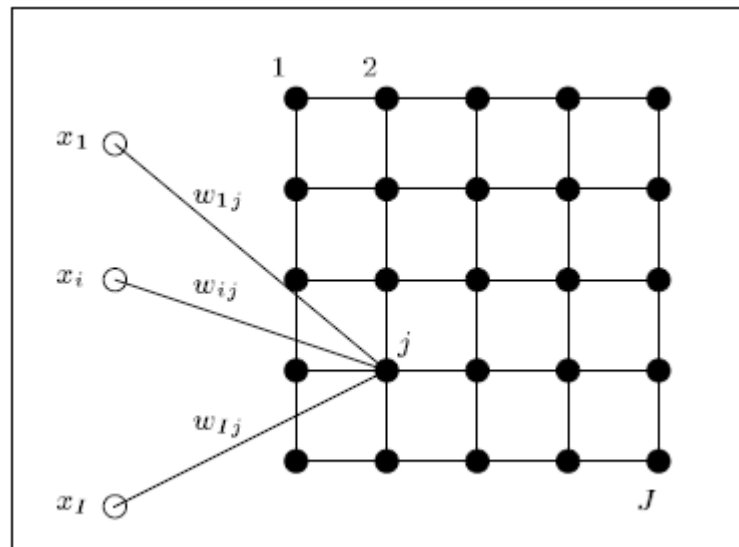


Figure III.1 : Réseau de *Kohonen*

La carte topologique *SOM* de *Kohonen* (*Self-Organizing Map*) impose une structuration spéciale à ses neurones. Cette structuration lie les neurones et les contraint à respecter une

certaine topologie lors de l'apprentissage. Ainsi, des données proches dans l'espace d'entrée ont des représentations proches dans la carte topologique de *Kohonen*.

La représentation des caractères sous forme de caractéristiques joue un rôle important dans le processus de reconnaissance.

Ils ont adopté les descripteurs elliptiques de Fourier sur des contours fermés et ils les ont utilisées sur les signaux $x(t)$ et $y(t)$ de l'écriture en-ligne. C'est-à-dire, une approximation des signaux a été utilisée sur les signaux de la trajectoire du stylet.

Dans ce travail une variation de certains paramètres a été étudiée sur la mémoire de *Kohonen*, ils ont exécuté trois expériences en faisant varier un paramètre à la fois.

1. la dimension du vecteur de caractéristiques;
2. le nombre d'itérations dans l'entraînement de la mémoire;
3. le nombre de nœuds de la mémoire.

III.2.2 Reconnaissance de caractères Arabes en-ligne par maximum d'entropie

Dans la thèse [15], une méthode bayésienne de reconnaissance en-ligne de caractères Arabes a été décrite où les densités de probabilité de classes sont estimées par le principe du maximum d'entropie.

Une représentation de caractères Arabes en-ligne a été utilisée, basée sur les distributions empiriques (histogrammes) de tangentes et de différences de tangentes à des points régulièrement espacés sur le tracé des caractères. Les paramètres des densités *gibbsiennes* des vecteurs de représentation sont estimés via le principe du maximum d'entropie.

La méthode fait intervenir essentiellement trois aspects (Figure III.2). Premièrement, il s'agit de déterminer les statistiques des caractéristiques retenues pour la représentation. Deuxièmement, et parce que la méthode d'estimation par maximum d'entropie est une méthode paramétrique, il s'agit d'estimer les paramètres des densités de probabilité des classes. Finalement, il faut estimer, à une constante multiplicative commune près, les fonctions de partition des différentes classes, ce qui peut se faire à partir des échantillons d'entraînement (méthode directe) ou en utilisant une distribution de référence (méthode indirecte).

Pour accroître les taux de reconnaissance, ils ont considéré une combinaison de classificateurs bayésiens. La représentation et les méthodes de reconnaissance développées ont été validées sur une base de données de caractères Arabes isolés enligne contenant 9504 échantillons écrits par 22 scripteurs.[15]

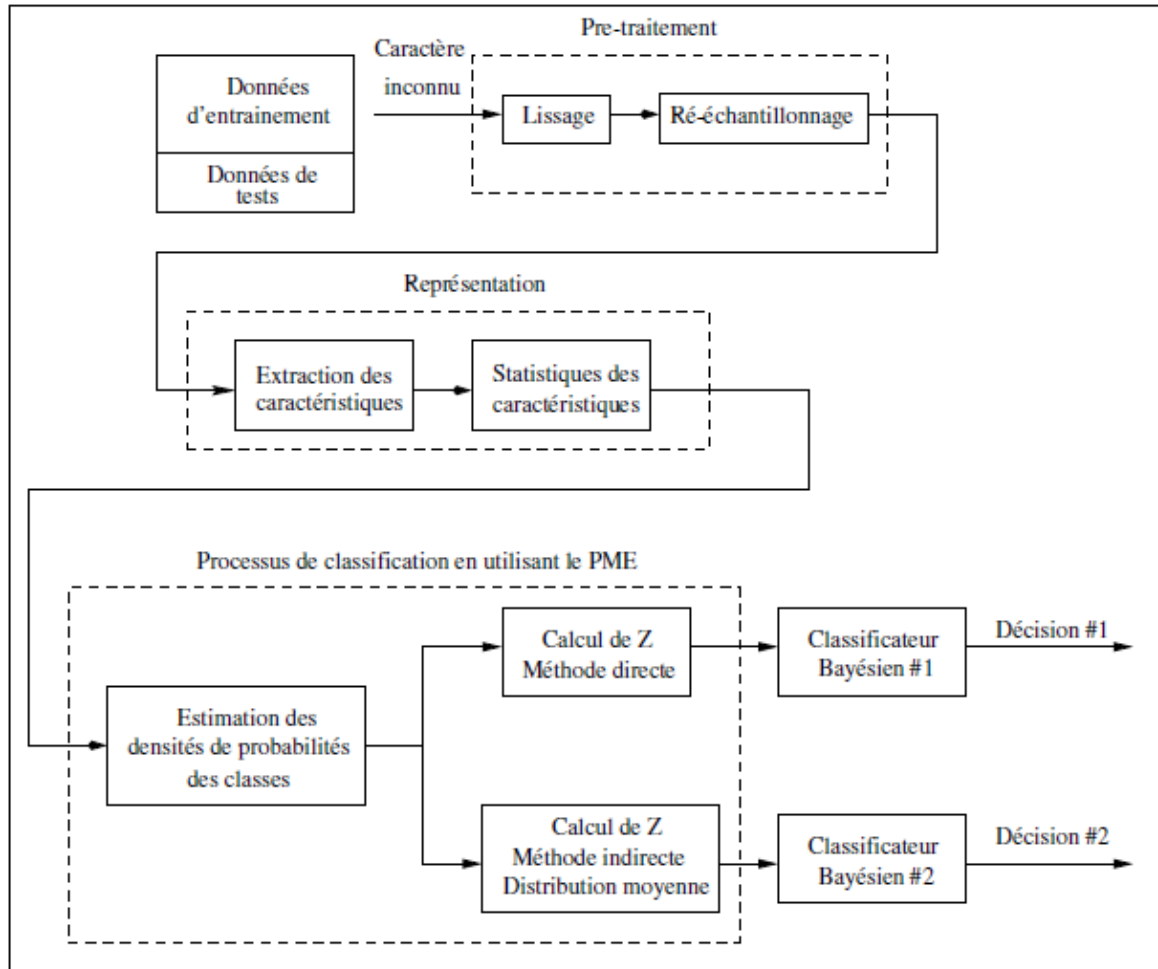


Figure III.2 : Schéma synoptique du système proposé de reconnaissance en-ligne de caractères Arabes en utilisant le principe du maximum d'entropie

Pour estimer les paramètres des densités de probabilités, ils ont conduit plusieurs tests en faisant varier le nombre de caractéristiques, le nombre d'intervalles des histogrammes et le nombre d'itérations. Les matrices de confusion des classificateurs bayésiens retenus ont été présentées et une combinaison simple de classificateurs bayésiens a été proposée pour

améliorer sensiblement les taux de reconnaissance. les prétraitements effectués sur les tracés des caractères d'entrée ont été décrites brièvement.

III.2.3 Reconnaissance en ligne du manuscrit arabe par les HMMs

Fadi Biadisy et al présentent dans un système basé sur les Chaines de Markov Cachées : *HMM (Hidden Markov Model)* pour fournir des solutions pour la plupart des difficultés inhérentes au manuscrit arabe : connectivité des lettres, formes des caractères et la dépendance de leurs positions et les délais de segmentation des strokes.

Ce travail est effectué en quatre étapes:[17]

III.2.3.1 Prétraitement Géométrique:

À ce stade, les séquences de points *PS (point sequence)* acquis passent une phase de traitement géométrique pour réduire au minimum des variations d'écriture. Ils ont employé un algorithme de filtre passe-bas pour réduire le bruit et enlever des défauts provoquées par des dispositifs d'acquisition. Ensuite, l'algorithme de *Douglas et de Peucker* a été adopté pour simplifier les séquences de point en employant une tolérance t_1 (déterminé empiriquement) afin d'éliminer les points superflus non pertinents pour la classification des formes. Dans l'étape finale, ils ont effectué la normalisation par ré-échantillonnage des séquences de points conséquents.

III.2.3.2 Extraction de paramètres :

Trois paramètres extraits pour la séquence de points: *Local-angle, super-segment et loop-presence*.

III.2.3.3 Manipulation des strokes avec délais :

Ces manipulations sont essentielles à la distinction de diverses lettres arabes. Ainsi, la manipulation des strokes avec délais (*ou bien retardés*) est correctement essentielle pour l'identification appropriée du manuscrit arabe. Ils ont développé l'algorithme de projection des strokes retardés comme méthode de manipulation de *strokes*.(voir figure III.3)

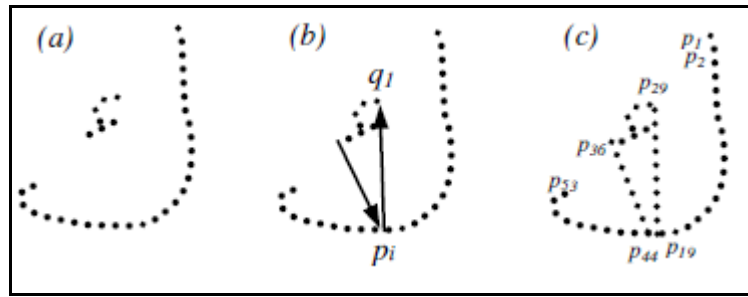


Figure III.3 : La projection du stroke retardé (hamza) dans la lettre Kaf ء
 (b) le stroke retardé est projeté dans le corps
 (c) la nouvelle PS généralisée (p1 à p53).

III.2.3.4 Construction de vecteurs de paramètres :

Un HMM est employé pour la tâche de reconnaissance, l'entrée (séquence d'observations) à ce type de modèle est une séquence des valeurs discrètes. Ainsi, un processus de quantification est exigé pour convertir la séquence du vecteur de paramètres 3D, extrait à partir d'une partie manuscrite de mot, en séquence discrète d'observation. Pour contraindre l'espace de recherche, dictionnaire des mots valides a été utilisé.[17]

La forme de la letter est incluse dans un réseau qui représente le dictionnaire du partie de mot (*word-part dictionary*) $WPD_{k,i}$. Ce réseau est optimisé en groupant tous les suffixes partagés, comme représenté sur la figure III.4.

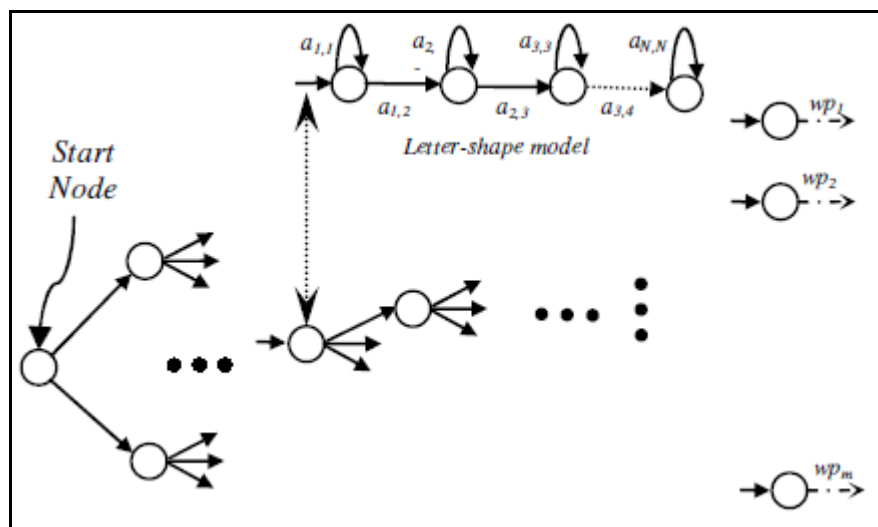


Figure III.4 : Le réseau représentant une partie du mot word-part WP

III.2.4 Système de reconnaissance d'écriture arabe en ligne basé sur le codage visuel et l'algorithme génétique

Dans le travail de M. Kherallah et al [18], l'encodage des mots arabes consiste à représenter le mot à partir de la séquence originale des coordonnées 2D par une séquence des codes visuels. Cette approche est basée sur la perception visuelle pour extraire la description primitive du mot, qui sera utilisable dans la phase de reconnaissance manuscrite en ligne. Pour cette raison, quelques algorithmes est utilisés pour l'extraction automatique des codes visuels à partir du script cursive. Le taux d'extraction est atteint 72%.

Les limites de cette stratégie heuristique ont été corrigées par un concept de base de l'algorithme génétique afin de choisir la meilleure combinaison des caractères qui représentent le mot arabe correct. Dans ce travail, un système a été développé pour reconnaître l'écriture cursive arabe en ligne

III.2.4.1 Extraction des primitives

Le mot a généralement deux zones principales : la zone *médiane* et la zone *saisissante*. Chaque zone se compose de ses propres indications visuelles présentée dans les deux tableaux suivants : [19]

Tableau III.1 : L'information du code visuel pour la zone en avant




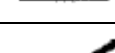
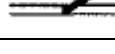
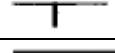






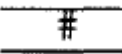
Notation	Code visuel	La forme
Alif	<i>Al</i>	
Shaft	<i>Ha</i>	
Left oblique shaft	<i>Hog</i>	
Left oblique shaft	<i>Hod</i>	
Leg	<i>Ja</i>	
Pocket	<i>Po</i>	

Tableau III.2 : L'information du code visuel pour la zone médiane

Notation	Code visuel	La forme
Valley	<i>Va</i>	
Occlusion	<i>Oc</i>	
Curve open right	<i>Cod</i>	
Left open curve	<i>Cog</i>	
Ain	<i>Ain</i>	
Sad	<i>Sad</i>	
Space inter tracing	#	

A chaque code visuel est attribué une étiquette comme montré dans le tableau III.3.

Tableau III.3 : Étiquettes des Codes Visuels

Code Visuel	<i>Va</i>	<i>Oc</i>	<i>po</i>	<i>Al</i>	<i>Ja</i>	<i>Ha</i>	<i>CoD</i>	<i>CoG</i>	<i>HoG</i>	<i>HoD</i>	<i>Ain</i>	<i>Sad</i>	#
Etiquette	1	2	3	4	5	6	7	8	9	10	11	12	13

III.2.4.2 Extraction des caractéristiques du script arabe

Le mot arabe est présenté par deux chaînes: la première représente les indices visuels et la deuxième représente les lettres différentes appelées : « chromosomes ». Le nombre de strokes est déterminé par le calcul du nombre de taux maximal de la trajectoire.

Le tableau suivant montre les indices visuels et les codes visuels de quelques lettres arabes. Ces caractéristiques sont employées dans les opérations préliminaires de l'algorithme génétique.

Tableau III.4 : Quelques lettres avec leurs caractéristiques visuelles et le nbr de strokes

Lettres	0	0	0	0	ا	0	0	ل	0	0	ع
Les indices visuels	5	0	0	2	1	1	4	1	1	0	7
Les chromosomes	0	0	0	0	42	0	0	40	0	0	27
Ordre des strokes	11	10	9	8	7	6	5	4	3	2	1

III.2.4.3 Approche génétique

Les algorithmes génétiques est une classe des méthodes d'optimisation et de recherche globale. Il existe trois étapes dans le processus de recherche génétique : sélection, croisement et mutation. Le mot est représenté par une continuation des codes visuels des lettres. Dans ce cas l'ordre des lettres est pris en compte. N est le nombre des lettres basique attribuées et extraites à partir du mot cursif. Par conséquent, chaque gene de la population a N chromosomes et chaque chromosome a un parmi les 58 valeurs possibles (de 1 à 57 pour les caractères arabes et la valeur 0 pour les caractères qui ont plus d'indication visuelle) énumérées du droite à gauche. Par exemple le mot: علم. Le code visuel est: *Cod/Va/Va/Al/Va/Va/Occ/Ja*. Les chromosomes: 27/40/42. Le tableau suivant montre les différents codes:

Tableau III.5 : Exemple de représentation d'un mot arabe

Lettres Arabes	Indices	Nombre de strokes	Codes visuels
ا	1	1-2	Al
ل	2	2-3	Al/Va
ع	3	3-4	Va
م	4	3-4-5	Va/Va/Va

La valeur retournée par la fonction d'adaptation *Fitness* pour un chromosome représente le degré de *matching* entre le mot représenté par le chromosome et le mot réel écrit.

Le calcul de la valeur de cette fonction est basé sur la somme des mesures de ressemblance entre deux chaîne d'indices visuels. Le tableau suivant présente la valeur de fonction fitness. Le moteur génétique va chercher la meilleure combinaison des caractères pour construire le mot analysé.

Tableau III.6 : *Matrice Fitness*

VI	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	1	1	1	1	1	1	1	1	1	1	1
2	1	0	1	1	1	1	0.5	0.5	1	1	0.5	0.5	1
3	1	1	0	1	0.5	1	1	1	1	1	1	1	1
4	1	1	1	0	1	0.5	1	1	0.5	0.5	1	1	1
5	1	1	0.5	1	0	1	1	1	1	1	1	1	1
6	1	1	1	0.5	1	0	1	1	0.5	0.5	1	1	1
7	1	0.5	1	1	1	1	0	1	1	1	0.5	0.5	1
8	1	0.5	1	1	1	1	1	0	1	1	0.5	0.5	1
9	1	1	1	0.5	1	0.5	1	1	0	0.5	1	1	1
10	1	1	1	0.5	1	0.5	1	1	0.5	0	1	1	1
11	1	0.5	1	1	1	1	0.5	0.5	1	1	0	0.5	1
12	1	0.5	1	1	1	1	0.5	0.5	1	1	0.5	0	1
13	1	1	1	1	1	1	1	1	1	1	1	1	0

"VI": *Visual indices*

L'étape de la reproduction et sélection, les fonctions avec les valeurs fitness les plus élevées ont plus de chance pour reproduire. Dans cette étape les meilleurs chromosomes vont dominer tant que les plus mauvais seront jetés.

L'opérateur de croisement consiste à sélectionner deux parents, avec la meilleure probabilité pour les chromosomes les plus adaptés. Pendant cette opération des parties de chaque paire de parents ont sélectionnées et les chromosomes sont générés à partir une concaténation des extrémités échangées. (Voir le tableau III.7)

Tableau III.7 : Opérateur de croisement entre deux mots

Le 1 ^{ier} mot	0	0	ح	ر	،	0	ر	0	0	0	ح	ر
IV	0	3	1	1	13	5	1	1	0	2	1	1
CH	0	0	46	4	10	0	13	0	0	0	44	4
Ordre du stroke	12	11	10	9	8	7	6	5	4	3	2	1
Le 2 ^{ième} mot	0	ر	،	0	0	0	و	0	0	0	ح	
IV	0	5	13	5	0	2	1	1	0	0	8	
CH	0	14	10	0	0	0	51	0	0	0	9	
Ordre du stroke	11	10	9	8	7	6	5	4	3	2	1	

Le croisement sera fait sur quatre strokes. Deux générations ont été reproduites.(voir le tableau II.8)

1^{ier} mot : حزين

2^{ième} mot : تمور

Tableau III.8 : Résultat de croisement

Le 1 ^{ier} mot	0	0	ح	0	ر	،	0	ر	0	0	0	ح	
IV	0	3	1	0	1	13	5	1	1	0	0	8	
CH	0	0	46	0	4	10	0	13	0	0	0	9	
Ordre du stroke	12	11	10	9	8	7	6	5	4	3	2	1	
Le 2 ^{ième} mot	0	ر	،	0	0	0	و	0	0	0	ح	0	ر
IV	0	5	13	5	0	2	1	1	0	2	1	0	1
CH	0	14	10	0	0	0	51	0	0	0	44	0	4
Ordre du stroke	13	12	11	10	9	8	7	6	5	4	3	2	1

Comme en nature, quelques individus feront se produire des mutations aléatoires dans leurs gènes. Le taux de mutation indique la chance qu'un gène donné dans un individu sera muté. Si un gène est choisi pour la mutation puis sa valeur sera changée. Dans le cas de la représentation par bit, le gène sera simplement renversé. (voir tableau III.9).

Tableau III.9 : Résultat de mutation pour le mot : علوم

Le mot	0	0	ﻡ	ﺀ	ﻭ	0	0	0	ﻝ	0	0	ﻉ	0	0
IV	5	0	2	13	5	0	2	1	1	4	1	1	0	7
CH	0	0	41	10	0	0	0	51	0	0	40	0	0	27
Ordre du stroke	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Le mot muté	0	0	ﻡ	ﺀ	ﻭ	0	0	0	ﻝ	0	0	ﻉ	0	0
IV	3	0	7	13	5	0	2	1	1	4	1	1	0	7
CH	0	0	53	10	0	0	0	51	0	0	40	0	0	27
Ordre du stroke	14	13	12	11	10	9	8	7	6	5	4	3	2	1

III.3 Bases de données En ligne reconnues

III.3.1 Base de donnée latine UNIPEN

UNIPEN est un projet international pour la collection de données en ligne et des exemples pour l'écriture en-ligne, dans lesquelles un grand nombre d'universités et de compagnies. Le projet a eu une étape fermée, dans laquelle les environ 40 donateurs ont voulu réaliser des essais de test et d'apprentissage. Plusieurs compagnies et universités ont fournis des échantillons d'écriture en ligne d'UNIPEN. La première série de la base de données manuscrite en ligne d'UNIPEN est un "ensemble d'apprentissage", contenant assez de données pour extraire les configurations de test et d'apprentissage.

UNIPEN est une base de grande taille, très employée dans l'évaluation des performances des différents systèmes de reconnaissance d'écriture latine en ligne. [20]

La base UNIPEN est la base de référence pour l'élaboration et la comparaison de systèmes de reconnaissance d'écriture. Cette base contient des tracés de plus de 200 scripteurs. La difficulté de cette base est due principalement au nombre de scripteurs et donc aux nombreux allographes qu'ils emploient. La version R01/V06 est souvent usitée dans le domaine. [21]. Cette base de données présente l'avantage de contenir un grand nombre de données dépaillées dans le tableau suivant :

Tableau III.10 : *Taille de sous-corpus de la base UNIPEN*

Base	Nombre de Classes	Nombre total d'exemples	Nombre d'exemples en Apprentissage	Nombre d'exemples en Test
Chiffres	10	15635	10423	5212
Minuscules	26	52267	34844	17423
Majuscules	26	26605	17736	8869

Le tableau ci-dessous présente l'ensemble des répertoires fournis dans le CD-ROM de la base de données.[8]

Tableau III.11 : *Description de la base de données UNIPEN*

Répertoires	Description	Unité
1a	Chiffres isolés	char
1b	Lettres majuscules isolées	char
1c	Lettres minuscules isolées	char
1d	Symbols isolés (ponctuations etc.)	char
2	Caractères isolés, cas mixte	char
3	Caractères isolés dans le contexte des mots et du texte	char
4	Mots isolés imprimés, non mixités avec les chiffres et les symboles	word
5	Mots isolés imprimés, l'ensemble des caractères complet.	word
6	Le style mixte des mots (sans les chiffres et les symboles)	word
7	Mots isolés, aucun style, l'ensemble des caractères complet	word
8	Texte (minimum deux mots)	text

III.3.2 Base de donnée latine IRONOFF

La base de données IRONOFF (IReste ON/OFF database) est une base de données duales en-ligne et hors-ligne collectées et distribuées par LORIA. Elle contient un nombre important de caractères isolés, de chiffres et de mots en français et anglais au format UNIPEN. Cette base de données a été créée de telle sorte qu'un point en-ligne puisse être projeté sur sa position correspondante dans l'image scannée, et inversement chaque élément du tracé hors-ligne peut être temporellement indexé.[11]

Cette base a été acquise avec une tablette Wacom UltraPad, la résolution spatiale typique est de l'ordre de 300 points par pouce tandis que la fréquence d'échantillonnage des données est de l'ordre de 100 points à la seconde. Elle a été collectée auprès d'environ 700 scripteurs différents. Le tableau suivant montre un sous-corpus de la base de données contenant : les chiffres, lettres minuscules, lettres majuscules isolés.

Tableau III.12 : *Taille de sous-corpus de la base IRONOFF*

Base	Nombre de Classes	Nombre total d'exemples	Nombre d'exemples en Apprentissage	Nombre d'exemples en Test
Chiffres	10	4108	3059	1510
Minuscules	26	11868	7952	3916
Majuscules	26	11879	7953	3926

III.3.3 Base de donnée arabe ADAB

La base de données ADAB (Arabic DAtaBase) est développée pour améliorer la recherche et le développement des systèmes de reconnaissance du texte manuscrit arabe en ligne. Cette base de données est développée en coopération entre l'Institut des Technologies de communication (*Institute for Communications Technology IfN*) et l'Ecole Nationale

d'Ingénieurs de Sfax (ENIS), le groupe de recherche (*Research Group on Intelligent Machines REGIM*), Sfax, Tunis.[22] La base de données dans sa version 1.0 contient 15158 mots arabes écrits par 130 scripteurs différents, la pluparts d'entre eux ont été sélectionnés de l'Ecole Nationale d'Ingénieurs de Sfax (ENIS). Le text écrit contient 937 noms de villes et villages tunisiens. Des outils spéciaux ont été développés pour l'acquisition et la vérification de données. Ces outils donnent la possibilité d'enregistrer les données écrites en ligne et d'autres informations concernant les scripteurs. Cette base de données est décomposée en 3 sous-corpus. Les détails à propos le nombre de fichiers, les mots, les caractères et les scripteurs pour chaque ensemble sont résumés dans la table suivante :

Tableau III.13 : sous-corpus de la base de données ADAB 1,2 et 3

<i>Sous-corpus</i>	<i>Fichiers</i>	<i>Mots</i>	<i>Caractères</i>	<i>Scripteurs</i>
1	5037	7670	40500	56
2	5090	7851	41515	37
3	5031	7730	40544	39
Somme	15158	23251	122559	132

III.4 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art relatif à notre tâche de reconnaissance de manuscrit en-ligne. Après avoir introduit par la citation de quelques travaux du manuscrit en ligne à base de référence, nous avons préféré de focaliser sur le manuscrit arabe. Comme nous avons détaillé trois travaux récents et leurs problématiques différentes. Nous avons vu que les travaux et les problématiques posé par N. Mezghani [1] [15][16] sont les plus proches du point de vue objectifs à notre réalisation. La différence est mentionnée au niveau de l'approche utilisée et les améliorations de modèle.

Enfin, nous avons présenté trois bases de données à titre d'exemple. Parmi ces dernières la base de données ADAB qui est la seule base de données arabe validée et utilisée pour l'implémentation. Et vu l'accès payant et la limite à cette base d'une part et d'autre part pour mieux guider la recherche dans notre labo, il est préférable toujours de construire nos propres échantillons dans une nouvelle base de données arabe en ligne.

Chapitre IV : APPROCHES UTILISÉES ET MODÈLES IMPLÉMENTÉES

IV.1. Introduction

Les réseaux Bayésiens sont des modèles représentant des connaissances incertaines sur des phénomènes complexes. Ils sont une union entre la théorie des probabilités et la théorie des graphes afin de donner des outils efficaces pour représenter une distribution de probabilités jointes sur un ensemble de variables aléatoires. La représentation de la connaissance se base sur la description, par des graphes, des relations de causalité existant entre les variables définissant le domaine d'étude. A chaque variable, on associe une distribution de probabilités locales spécifiant une relation causale.

On utilise les réseaux bayésiens pour leur capacité d'effectuer des inférences dans un contexte d'incertitudes, en quelque sorte comme alternative aux systèmes experts. On les utilise aussi pour leurs algorithmes d'apprentissage, comme alternative aux autres méthodes de modélisation quantitative, en les considérant comme des modèles de régression.

La structure graphique d'un réseau bayésien traduit les relations de dépendance conditionnelle entre variables aléatoires à l'aide de liens munissant l'ensemble des ces variables de la structure de graphe orienté acyclique.

Les réseaux bayésiens dynamiques (DBN) prolongent la représentation de BN aux processus dynamiques. Dans la littérature la représentation de DBN est employée le plus souvent pour les premiers processus stationnaires de Markov. Pour ce cas, Friedman et autres décrivent une représentation simplifiée en termes de deux charges statiques BNs définie au-dessus des variables d'une tranche simple de temps.[23]

La reconnaissance en-ligne d'écriture manuscrite est employée pour reconnaître automatiquement des caractères manuscrits en utilisant des tablettes électroniques et des stylos. Elle est différente de la reconnaissance hors-ligne parce qu'elle peut utiliser l'information du mouvement de main saisie par les dispositifs électroniques. Selon la

popularité croissante des PDAs et d'ordinateurs de stylo, de nos jours, il devient plus important de les utiliser comme méthode d'entrée commode et portable.

Pour que la reconnaissance de caractères isolés soit fortement précise, il est important aussi de modéliser la structure des caractères ordinairement que possible.

A base de test, un modèle hybride neuro-génétique est proposé pour améliorer et adapter le système dans sa phase finale. Cette hybridation se focalise sur une optimisation paramétrique des poids synaptiques dans topologie neuronale.

IV.2. Modélisation par des Réseaux Bayésiens Dynamiques

IV.2.1. Introduction

Dans cette partie, nous considérons qu'un caractère se compose de strokes et de leurs relations spatiales. Les strokes sont des tracés élémentaires directs ou presque droits qui ont des directions distinctes des tracés reliées dans l'ordre d'écriture. Figure III.1 montre quelques exemples des strokes des chiffres numériques. Les caractères 1 et 4 ont quatre strokes respectivement comprenant les premiers levés du stylo. Le tracé curviligne comme la partie la plus inférieure du caractère 5 peut être rapprochée par plusieurs strokes.

Les relations spatiales des strokes indiquent les dépendances des positions entre les strokes ; la position d'un stroke obtient une influence sur les autres strokes. Par exemple, quand les trois strokes du caractère 4 sont écrites comme la figure IV.2 (a), la position de X du point final du dernier stroke (E_x) est susceptible d'être centré sur l'axe X. Cependant, quand elles sont légèrement dans le sens des aiguilles d'une montre comme la figure IV.2 (b), E_x est susceptible d'être situé sur le côté gauche de l'axe X.

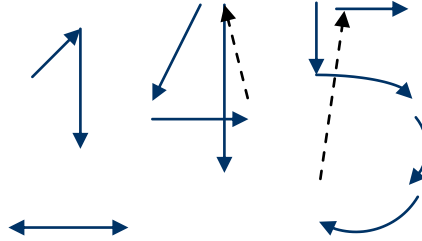


Figure IV.1 : Les strokes des caractères.

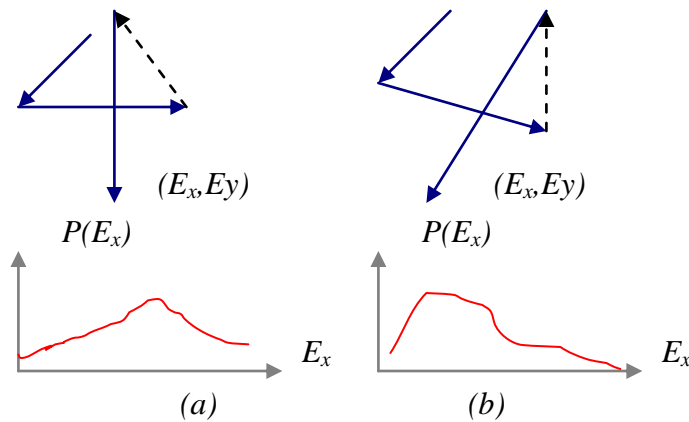


Figure IV.2: Exemple des relations spatiales des strokes.

Selon les positions d'autres strokes, E_x (position X du point final de dernier stroke du caractère « 4 » est susceptible d'être: (a) centré dans l'axe X et (b) dans le côté gauche de l'axe X.

Modéliser clairement des strokes et leurs relations spatiales a beaucoup d'avantages. Premièrement, les strokes sont les éléments conceptuels et leurs relations spatiales sont conceptuellement significatives, particulièrement dans les caractères des langues orientales telles que coréen et chinois. Deuxièmement, leurs formes sont habituellement simples et trompeuses et écrites de sorte qu'elles puissent être sûrement modelées. En outre, leur complexité modelant est réduite et comparé à celle du caractère entier. En conclusion, les relations spatiales de stroke sont habituellement robustes contre des variations géométriques et importantes pour les caractères distinctifs des formes semblables.

Malgré les avantages, les strokes et leurs relations spatiales n'ont pas été activement adoptés dans les systèmes de reconnaissance on-line de l'écriture manuscrite pour des approches conventionnelles de reconnaissance qui sont basées sur les modèles de Markov cachés (HMMs) [3][6] et les réseaux de neurones à délais temporels [24]. Cependant, ils ne modélisent pas des strokes explicitement. Dans ces approches, une entrée d'écriture est habituellement convertie en vecteurs locaux de dispositif et leurs indépendances sont supposées. Ainsi, une entrée de stroke devient divisée en plusieurs vecteurs locaux indépendants du dispositif, qui le rend difficile pour modéliser explicitement des modèles statistiques et des dépendances des strokes.

Notre but de recherche est de concevoir un travail dans lequel des strokes et leurs relations spatiales sont explicitement modélisés et de développer un système robuste de reconnaissance basé sur ce concept.

Pour réaliser ce but, nous proposons une approche graphique probabiliste : un réseau bayésien avec les propriétés suivantes. Premièrement, un caractère est représenté dans la hiérarchie d'un modèle de caractère, des modèles de stroke et des modèles de point. Deuxièmement, des relations spatiales à l'intérieur et entre des strokes sont explicitement représentées comme des dépendances. En conclusion, les réseaux bayésiens sont adoptés pour représenter tous les modèles parce qu'ils sont utiles pour citer les dépendances graphiquement et d'une façon probabiliste [3][25][6].

IV.2.2. Modélisation en strokes et leurs relations spatiales

IV.2.2.1 La modélisation par réseau bayésien :

Un réseau bayésien [24] est un modèle graphique pour représenter des relations spatiales entre les variables aléatoires. Il peut efficacement coder la probabilité jointe des variables aléatoires. En outre, il peut modéliser des dépendances entre elles d'une façon probabiliste et graphique.

Le réseau bayésien est un graphe acyclique direct des variables aléatoires qui sont représentées par ses noeuds. Leurs dépendances sont représentées par ses arcs et codées en tant que probabilité de distribution conditionnelle aux niveaux des noeuds. Si une variable

aléatoire X dépend de Y , alors le noeud X a un arc directionnel du noeud Y . Par exemple, le réseau bayésien montré dans la figure IV.3(a) indique que les variables aléatoires X_1 et X_2 n'ont aucune dépendance, X_3 dépend de X_1 et X_4 dépend de X_1 , de X_2 et de X_3 . Les noeuds X_1 , X_2 , X_3 et X_4 codent les probabilités conditionnelles $P(X_1)$, $P(X_2)$, $P(X_3/X_1)$ et $P(X_4/X_1;X_2;X_3)$, respectivement. La distribution jointe de probabilité, $P(X_1;X_2;X_3;X_4)$, est obtenu à partir du produit des probabilités de distributions conditionnelles pour tous les noeuds (figure IV.3 (b)).[26]

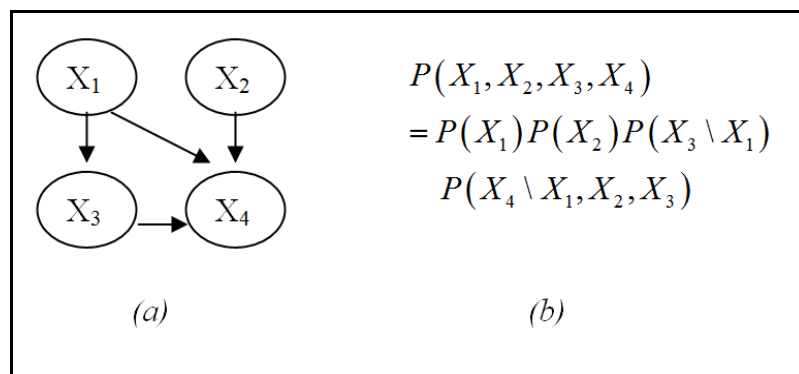


Figure IV.3 : Exemple d'un réseau bayésien

(a) représentation des dépendances entre les variables aléatoires

(b) calcul de leur probabilité jointe.

Ce n'est pas une tâche simple d'obtenir la probabilité de distribution conditionnelle exacte quand les variables aléatoires ont des valeurs continues et des dépendances d'ordre élevé. L'approche conventionnelle est de faire une table conditionnelle de probabilité pour toutes les configurations possibles des valeurs de variable aléatoire. Cependant, elle exige la quantification des valeurs continues. En outre, la taille de la table se développe exponentiellement à mesure que l'ordre de la dépendance augmente. Par conséquent, on a besoin d'un grand nombre de données d'apprentissage formées exponentiellement proportionnelles à l'ordre de dépendance pour l'évaluation fiable de probabilité.

Pour ces raisons, nous adoptons les distributions gaussiennes conditionnelles [27]. On assume qu'une variable aléatoire a une distribution gaussienne et son moyen est déterminé par la somme linéaire de poids de valeurs de variable dépendante. Cependant, sa covariance

ne dépend pas d'elles. Quand une variable aléatoire multi variable X dépend de X_1, \dots, X_n , la probabilité de distribution conditionnelle est donnée comme suit :

$$P(X = x \mid X_1 = x_1, \dots, X_n = x_n) = (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left[-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right]. \quad \text{IV-1}$$

Le vecteur moyen μ est déterminé à partir des valeurs de variable dépendante :

$$\mu = W \left[x_1^T, \dots, x_n^T, 1 \right]^T \quad \text{IV-2}$$

Où W est une matrice de régression linéaire $d \times k$, d est la dimension de X , et k est la dimension de $[x_1^T, \dots, x_n^T, 1]^T$.

Quand il n'y a aucune variable dépendante, le vecteur moyen conditionnel d'équation (III.2) est réduit au moyen de la distribution gaussienne conventionnelle.

$$\mu = \text{mean}(X) \cdot 1. \quad \text{IV-3}$$

IV.2.2.2 Modèle de point (distribution gaussienne)

Un exemple de point a l'attribut des positions (x,y) dans le plan 2-D. La figure IV.4(a) montre des exemples des premiers points du caractère 'Mim'. Figure IV.4 (b) montre la distribution de leurs positions dans des boîtes de bondissement de même caractère. Les petits points correspondent aux exemples des données d'apprentissage.

Un modèle de point dans 2-D qui suit une distribution gaussienne pour modéliser les positions de point (figure IV.4 (c)). Il est représenté par un noeud dans un réseau bayésien comme montré dans la figure IV.4 (d).

La probabilité jointe d'un exemple de point et d'un modèle de point est obtenue à partir de la distribution gaussienne. On pose $O=(x,y)$ dénote l'exemple de point et $P=(X,Y)$ le modèle de point. Puis, la probabilité jointe est donnée comme suit :

$$\begin{aligned}
 P(P = O) &= P(X = x, Y = y) \\
 &= (2\pi)^{-1} |\Sigma|^{-1/2} \exp\left[-\frac{1}{2}(O - \mu)^T \Sigma^{-1}(O - \mu)\right].
 \end{aligned}
 \tag{IV-4}$$

$$\mu = [\text{mean}(X) \quad \text{mean}(Y)]^T
 \tag{IV-5}$$

Quand le point P dépend d'autres points P_1, \dots, P_n , sa probabilité jointe est donnée par la distribution gaussienne conditionnelle. En substituant la variable aléatoire P par X des équations (V.1) et (IV.2), il est donné comme suit :

$$\begin{aligned}
 P(P = O \mid P_1 = O_1, \dots, P_n = O_n) \\
 &= P(x, y \mid x_1, y_1, \dots, x_n, y_n) \\
 &= (2\pi)^{-1} |\Sigma|^{-1/2} \exp\left[-\frac{1}{2}(O - \mu)^T \Sigma^{-1}(O - \mu)\right].
 \end{aligned}
 \tag{IV-6}$$

Où:

$$\mu = \begin{pmatrix} w_{1,1} & w_{1,2} \dots w_{1,2n+1} \\ w_{2,1} & w_{2,2} \dots w_{2,2n+1} \end{pmatrix} [x_1, y_1, \dots, x_n, y_n, 1]^T.
 \tag{IV-7}$$

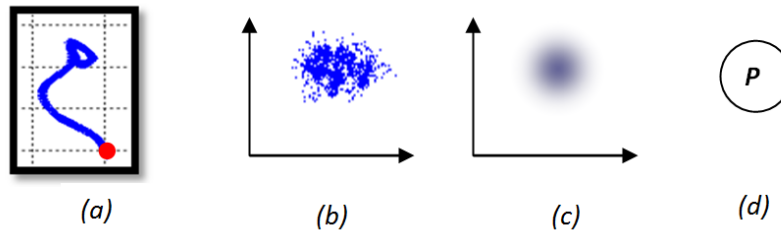


Figure IV.4 : Exemples de point et le modèle de point.

(a) Les exemples des premiers points de caractère 'Mim'.

(b) Distribution d'échantillon de leurs positions $X Y$.

(c) Modèle gaussien pour la représentation en 2-D.

(d) représentation par un réseau bayésien d'un modèle P de point : un seul nœud.

IV.2.2.3 Modèle de stroke (réseau bayésien statique)

Un exemple de stroke se compose de points. Par conséquent, un modèle de stroke se compose de modèles de point avec leurs relations spatiales, appelé relations spatiales intra strokes « Within Stroke Relationships » (WSRs).

Une WSR est définie comme dépendance d'un point médian entre deux points finaux d'un stroke. Le point médian est le point d'où la longueur de stroke partielle gauche et celle de la droite sont égales. Dans la figure IV.5(a), IP sont des points médians, et ep0's et ep1's sont les points finaux des exemples strokes. Figure IV.5(b) est la représentation par un réseau bayésien d'une WSR, composé d'un point médian (IP), deux modèles de point final (EP₀, EP₁) et des arcs de dépendance d'EP₀, EP₁ à l'IP.

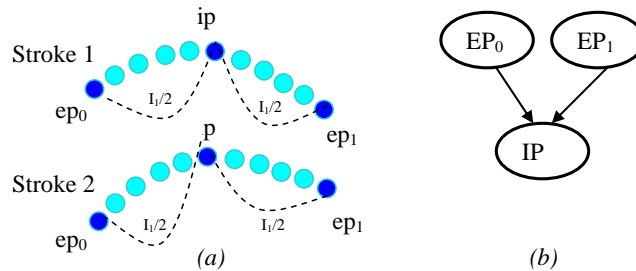


Figure IV.5 : Relations Spatiales Inter Strokes (RSISs).

(a) IP et deux points finaux ep0's et ep1's de points médians des exemples de course.

(b) Représentation par un réseau bayésien d'une WSR.

La figure IV.6 montre l'exemple récursif de construction de stroke. À la première itération récursive ($d = 1$), IP₁ est ajouté pour modéliser des points médians de tous les exemples de stroke. Elle a la WSR des points finaux (les arcs d'EP₀ et d'EP₁ à IP₁). À la deuxième itération récursive ($d = 2$), IP₂ et IP₃ sont ajoutés pour des points médians des strokes partiels gauches et droites, respectivement. En outre, elles ont les WSR des points finaux des strokes partiels. Figure IV.6 (c) est le modèle prolongé de stroke.

Avec ce procédé récursif, un modèle de stroke peut avoir autant de modèles de point selon les besoins. Dans ce chapitre, la profondeur de récursivité $d=3$ est choisie pour tous les modèles de stroke.

Il vaut la peine de noter que les modèles de point à de grandes profondeurs de récursivité n'encourent pas le problème de modèle non adéquat. Car lorsque la profondeur est grande, les strokes partiels deviennent beaucoup plus courts et plus linéaires. Par conséquent, les WSRs deviennent beaucoup plus fortes et les probabilités jointes des modèles supplémentaires de point obtiennent plus près la probabilité d'un seul.

La probabilité jointe est obtenue à partir de celles des modèles de point. Supposons qu'un modèle S a la profondeur d et un exemple de stroke a des points de longueur $t : O(1), \dots, O(t)$. Pour assortir, l'exemple de stroke est périodiquement prélevé dans les 2^{d-1} points médians. Ils sont dénotés comme $IP_1, IP_2, \dots, IP_{2^{d-1}}$ selon l'ordre du procédé de prélèvement récursif.

Puis, les IP_i d'exemples de point sont assortis aux modèles l' IP_i de point. La probabilité jointe est calculée comme suit par la propriété locale de Markov des probabilités conditionnelles dans les réseaux bayésiens [28]:

$$\begin{aligned}
 P(S = O(1), \dots, O(t)) &= P(EP_0 = O(1), EP_1 = O(t), IP_1 = ip_1, \dots, IP_{2^{d-1}} = ip_{2^{d-1}}) \\
 &= P(EP_0 = O(1)) P(EP_1 = O(t)) \times \prod_{i=1}^{2^{d-1}} P(IP_i = ip_i \setminus pa(IP_i))
 \end{aligned}
 \tag{IV-8}$$

Où le $pa(IP_i)$ est la configuration des nœuds parents lesquels les arcs de dépendance viennent à IP_i .

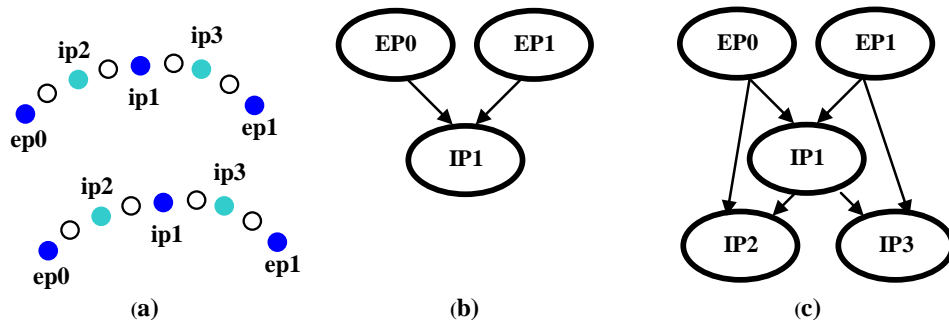


Figure IV.6 : La construction récursive d'un modèle de stroke.
 (a) Exemple des ip_1 's : point médian de stroke ip_2 s et ip_3 's
 (b) modèle de stroke de la profondeur $d=1$.
 (c) Modèle de stroke pour $d=2$.

IV.2.2.4 Modèle de caractère (réseau bayésien dynamique)

Un exemple de caractère se compose des strokes. En outre, les relations étroites existent entre eux. Par conséquent, un modèle de caractère se compose des modèles de stroke avec leurs relations spatiales, appelées relations spatial inter strokes ou en anglais « Inter Stroke Relationships (ISRs) ». Les ISRs sont représentées avec des dépendances des points finaux

des strokes. Dans le meilleur des cas, un stroke obtient l'influence de tous les points d'autres strokes. Figure IV.7(a) illustre la situation que le point final de dernier stroke du caractère 4 a des relations spatiales avec tous les points d'autres strokes (marquées par des flèches). Cependant, la représentation de toutes les relations spatiales possibles est trop complexe et superflue. Ainsi, nous les encapsulons comme des relations des points finaux de strokes (figure IV.7(b)). Par la suite, les ISRs sont représentées comme des arcs d'autres points finaux de strokes dans les réseaux bayésiens comme montré dans la figure IV.7(c).

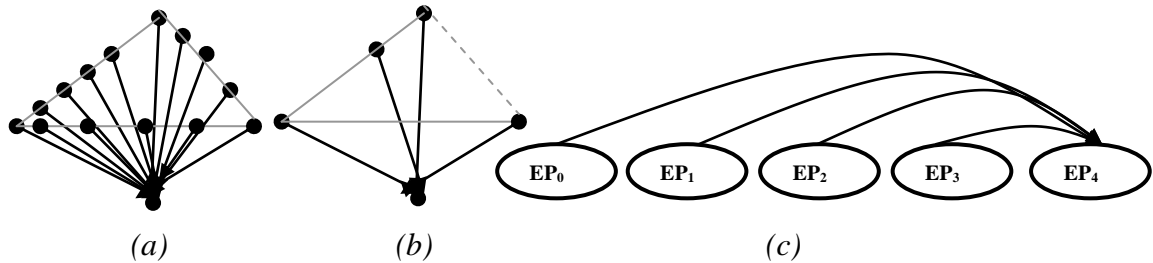


Figure IV.7 : Représentation des relations spatiales inter strokes (ISRs).

(a) le dernier stroke du caractère 4 peut avoir des relations avec tous les points des autres strokes.

(b) Encapsulation d'ISRs entre les points finaux de stroke.

(c) Représentation de réseau bayésien d'ISRs du dernier stroke.

Un modèle de caractère est construit en enchaînant des modèles de stroke selon leur ordre d'écriture et en indiquant les ISRs [25]. Figure IV.8 montre un réseau bayésien de caractère avec N strokes. Les EP_i sont les modèles des points finaux et les $IP_{i,j}$ sont les modèles de point dans le $i^{\text{ème}}$ stroke. Le point final droit du stroke précédent est partagé avec le point final gauche du stroke suivant. ISRs sont représentés par les arcs entre EP_i , et WSRs sont représentés par les arcs entrants aux $IP_{i,j}$. Cette figure prouve que des modèles de point, les modèles de stroke et leurs relations spatiales sont représentés hiérarchiquement dans un modèle de caractère.

L'ordre d'écriture des modèles de point est représenté par leurs positions gauches droites dans le modèle de caractère, les modèles de point dans le côté gauche sont écrits avant ceux dans le côté droit. Dans la figure IV.8, EP_0 est le premier modèle de point écrit dans un caractère.

Les modèles de point de premier stroke sont écrits dans l'ordre d' $IP_{1,2}$, $IP_{1,1}$, $IP_{1,3}$. Puis, les modèles de point des deuxièmes strokes sont écrits dans l'ordre d' EP_1 , $IP_{2,2}$, $IP_{2,1}$, $IP_{2,3}$.

Alternativement, les strokes suivants sont écrits de la même façon. En conclusion, EP_N est le dernier modèle de point écrit dans un caractère.[3][25][6]

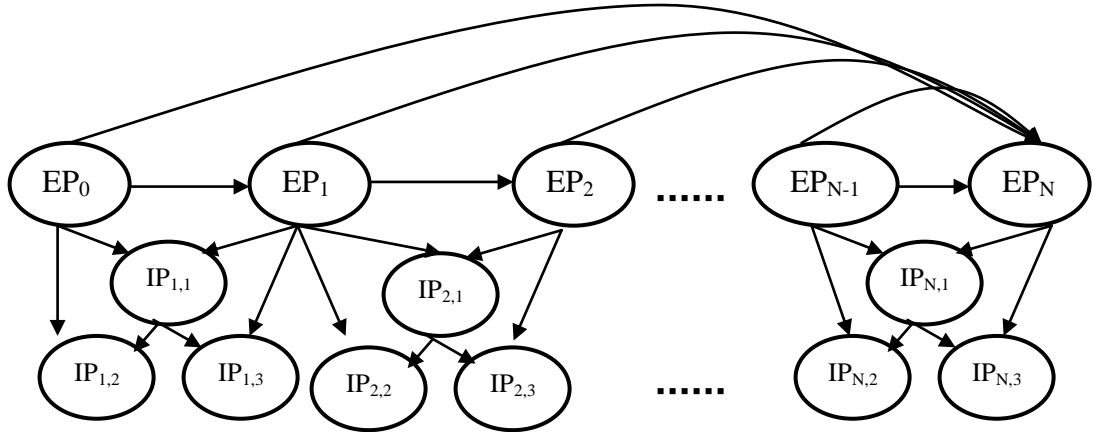


Figure IV.8 : La représentation par RB d'un modèle de caractère avec N strokes et $d=2$.

Le modèle de vraisemblance d'un caractère est calculé par l'énumération de toutes les segmentations possibles de stroke. Supposons qu'un modèle BN de caractère a N modèles de stroke et une entrée de caractère avec T points : $O(1), \dots, O(T)$. Puisque l'entrée n'a pas l'information de frontière, différentes segmentations sont possibles. On pose un exemple de segmentation de stroke par $\gamma = (t_0, t_1, \dots, t_N)$, $t_0 = 1 < t_1 < \dots < t_N = T$, et l'ensemble de totalité par Γ . Puis le modèle de vraisemblance d'un caractère est donné comme suit :

$$\begin{aligned}
 & P(O(1), \dots, O(t) \setminus BN) \\
 &= \sum_{\gamma=(t_0, \dots, t_N) \in \Gamma} P(S_1 = O(t_0, t_1), \dots, S_N = O(t_{N-1}, t_N)) \\
 &= \sum_{\gamma \in \Gamma} \prod_{i=1}^N P(S_i = O(t_{i-1}, t_i) \setminus S_1 = O(t_0, t_1), \dots, S_{i-1} = O(t_{i-2}, t_{i-1})) \quad \text{IV-9} \\
 &= \sum_{\gamma \in \Gamma} \prod_{i=1}^N P(S_i = O(t_{i-1}, t_i) \setminus EP_0 = O(t_0), \dots, EP_{i-1} = O(t_{i-1}))
 \end{aligned}$$

Où $O(t_i, t_j) = O(t_i), O(t_{i+1}), \dots, O(t_j)$. La probabilité jointe donnée par des strokes précédents est calculée comme suit :

$$\begin{aligned}
 P(S_i = O(t_{i-1}, t_i) \setminus EP_0 = O(t_0), \dots, EP_{i-1} = O(t_{i-1})) = & \\
 \left\{ \begin{array}{l} P(EP_i = O(t_i) \setminus O(t_0), \dots, O(t_{i-1})) \\ \prod_{j=1}^{2^d-1} P(IP_{i,j} = ip_{i,j}(O(t_{i-1}, t_i)) \setminus pa(IP_{i,j})) \text{ if } i > 1, \\ P(EP_0 = O(t_0)) P(EP_1 = O(t_1) \setminus O(t_0)) \\ \prod_{j=1}^{2^d-1} P(IP_{i,j} = ip_{i,j}(O(t_{i-1}, t_i)) \setminus pa(IP_{i,j})) \text{ if } i = 1, \end{array} \right. & \text{IV-10}
 \end{aligned}$$

Où $ip_{i,j}(O(t_{i-1}, t_i))$ est le $j^{\text{ième}}$ point échantillon de $O(t_{i-1}, t_i)$. En substituant Eq. (IV.10) par Eq. (IV.9), la probabilité du modèle n'est qu'un produit des probabilités jointes d'EPs et d'IPS :

$$\begin{aligned}
 P(O(1), \dots, O(t) \setminus BN) = & \\
 \sum_{\gamma \in \Gamma} \prod_{i=0}^N P(EP_i = O(t_i) \setminus O(t_0), \dots, O(t_{i-1})) & \text{IV-11} \\
 \times \prod_{i=1}^N \prod_{j=1}^{2^d-1} P(IP_{i,j} = ip_{i,j}(O(t_{i-1}, t_i)) \setminus pa(IP_{i,j})) &
 \end{aligned}$$

Les probabilités jointes des EPs peuvent être interprétées par des probabilités des positions globales de stroke et celles des IPs avec des probabilités des formes de stroke locales.

IV.2.3 Reconnaissance et Apprentissage

IV.2.3.1 Algorithme de reconnaissance

Un caractère manuscrit est identifié en trouvant le modèle de caractère qui produit la probabilité a posteriori la plus élevée donnée à l'entrée. Quand le $i^{\text{ième}}$ modèle de caractère est dénoté BN_i et les points d'entrée en tant que $O(1) \dots, O(T)$, alors le problème de reconnaissance peut être formulé comme suit :

$$\begin{aligned}
 & \arg \max_i P(BN_i \setminus O(1), \dots, O(T)) \\
 &= \arg \max_i \frac{P(BN_i)P(O(1), \dots, O(T) \setminus BN_i)}{P(O(1), \dots, O(T))} \\
 &= \arg \max_i P(BN_i)P(O(1), \dots, O(T) \setminus BN_i)
 \end{aligned}
 \tag{IV-12}$$

La vraisemblance du modèle de caractère est décrite précédemment dans (équation (IV.11)). Pour la calculer, toutes les segmentations possibles de stroke Γ sont considérées. Pour empêcher la complexité exponentielle de temps, nous supposons qu'elle peut être rapprochée par la probabilité jointe de caractère de la segmentation la plus probable γ^* dans Γ comme suit :

$$P(O(1), \dots, O(T) \setminus BN_i) \approx \max_{\gamma \in \Gamma} P(S_1 = O(t_0, t_1), \dots, S_N = O(t_{N-1}, t_N))
 \tag{IV-13}$$

Pour effectuer le calcul de la vraisemblance du modèle maniable dans le temps, nous avons besoin en plus une prétention pour la recherche de γ^* . En assortissant un stroke, toutes les segmentations possibles de ses strokes précédents devraient être considérées en raison des dépendances d'inter-stroke. Pour la simplicité de la recherche, nous supposons que la probabilité jointe d'un stroke est la plus haute avec la configuration la plus probable des strokes précédents. Ensuite, l'algorithme de recherche dynamique ARD « *dynamic programming search algorithm* » [29] peut être adopté comme dans la figure IV.9.

IV.2.3.2 Algorithme d'apprentissage

Pour construire un modèle de caractère, sa structure de dépendance et les paramètres conditionnels de probabilité devraient être déterminés. Dans ce chapitre, la structure de dépendance est déterminée par un concepteur modèle à partir de la connaissance a priori et les expériences.

La profondeur de récursivité des modèles de stroke est choisie égale à trois ($d=3$). Le nombre de modèles est déterminé à partir du nombre typique de stroke dans le caractère.

Les paramètres conditionnels de probabilité sont formés par des données d'apprentissage. Ils sont les matrices de régression linéaire W 's ($W=[w_{i,j}]$) et les covariances Σ 's pour les

modèles de points (équations IV.6) et (IV.7)). Si tous les modèles de point sont assortis aux exemples de point, alors ils peuvent être estimés à partir des algorithmes statistiques conventionnels de régression avec l'objet de maximum de vraisemblance ML « maximum likelihood » [16]. Supposons qu'un point P dépend de P_1, \dots, P_k et il y a N échantillons d'apprentissage. On note le $i^{\text{ème}}$ échantillon de P comme $p(i)$ et les valeurs de variable dépendante par $z^{(i)} = [x^{(i)}_1, y^{(i)}_1, \dots, x^{(i)}_k, y^{(i)}_k, 1]$. Puis, les paramètres sont estimés comme suit:[30]

$$W = \left(\sum_{i=1}^N p^{(i)} (z^{(i)})^T \right) \left(\sum_{i=1}^N z^{(i)} (z^{(i)})^T \right)^{-1} \quad \text{IV-14}$$

$$\Sigma = \frac{1}{N} \sum_{i=1}^N p^{(i)} (p^{(i)})^T - \frac{1}{N} W \sum_{i=1}^N z^{(i)} (p^{(i)})^T \quad \text{IV-15}$$

Pendant l'apprentissage du modèle de caractère, la ré-estimation des paramètres et la recherche de la segmentation en strokes la plus probable γ^* est répétée alternativement [29][30][28]. Cette approche est semblable à l'algorithme d'apprentissage EM (Expectation-Maximization). Etant donnés les paramètres (W et Σ), γ^* est mis à jour. Ensuite, avec la nouvelle γ^* , les paramètres sont ré-estimés. L'algorithme d'apprentissage détaillé est comme suit:

- Etape 1: Initialiser le modèle de caractère avec les données initiales (une partie des exemples des strokes manuellement segmentés).
- Etape 2: Chercher la segmentation la plus probable γ^* de la totalité des caractères d'apprentissage non segmentés en utilisant l'algorithme de recherché précédent.
- Etape 3: Estimer les paramètres (W et Σ) sur les exemples partitionnés par les γ^* .
- Etape 4: Répéter les étapes 2 et 3 jusqu'à ce que la somme des vraisemblances du modèle ne changera plus (stabilité).

```

Algorithme ARD dynamic programming search algorithm
 $S_i$  :  $i^{\text{ème}}$  modèle de stroke
 $\gamma_i(t)$  : la segmentation la plus probable quand  $S_1, \dots, S_i$  et  $O(1,t)$  sont matchés.
 $\delta_i(t)$  : la probabilité jointe donnée selon  $\gamma_i(t)$  .

Initialisation
 $\delta_0(1)=1, \gamma_0(1)= \{ \}$ 

Stroke matching
for t=2 to T
    for i=1 to N
         $\delta_i(t) = \max_{1 \leq b < t} P(S_i = O(b,t) \setminus \gamma_{i-1}(b) ) \cdot \delta_{i-1}(b)$ 
         $b^* = \operatorname{argmax}_{1 \leq b < t} P(S_i = O(b,t) \setminus \gamma_{i-1}(b) ) \cdot \delta_{i-1}(b)$ 
         $\gamma_i(t) = \gamma_{i-1}(b^*) \cup \{t\}$ 
    end
end

Vraisemblance du modèle de caractère
 $P(O(1), \dots, O(T) \setminus BN_i) \approx \delta_N(T)$ 

```

Figure IV.9 : Algorithme de recherche dynamique

IV.2.4. Amélioration de l'Algorithme de Recherche Dynamique

IV.2.4.1 Segmentation par Plage de Points

L'idée de ce nouveau concept est de partager le signal d'écriture en des tranches de points, alors que cet algorithme donne une segmentation par un ensemble de points par contre l'algorithme de recherche dynamique ARD donne la segmentation à chaque instant 't' (c'est-à-dire qu'au niveau de chaque point du signal, on affecte une segmentation de ce dernier, ce qui est très lent pour un système de reconnaissance en ligne).

Deux approches proposées pour effectuer la segmentation par plage de points :

a) Plage de points statiques : On fixe un nombre de points statiques pour toutes les observations, (exemple : le nombre total des points du signal $T=25$, la plage $P=5$, l'ancien

algorithme ARD se boucle 47 fois pour donner la segmentation finale alors que la nouvelle boucle exige que le nombre $((T/P)+1)$ est arrondi aux entiers proches supérieurs)

b) Plage de points dynamiques : On fixe ici un pourcentage de points au lieu d'un nombre fixe pour toutes les observations, ainsi le nombre de points des plages se change selon le nombre T total des points d'une observation. (Exemple : le nombre total des points du signal $T=25$, le pourcentage des points $Pr=20\%$ alors la plage $P=((T \cdot Pr)/100)$, ce nombre est arrondi aux entiers proches supérieurs)

Remarque : On garde les points intermédiaires dans une plage de points pour ne pas perdre une grande quantité d'informations.

L'idée de ce nouveau concept se résume dans la figure suivante :

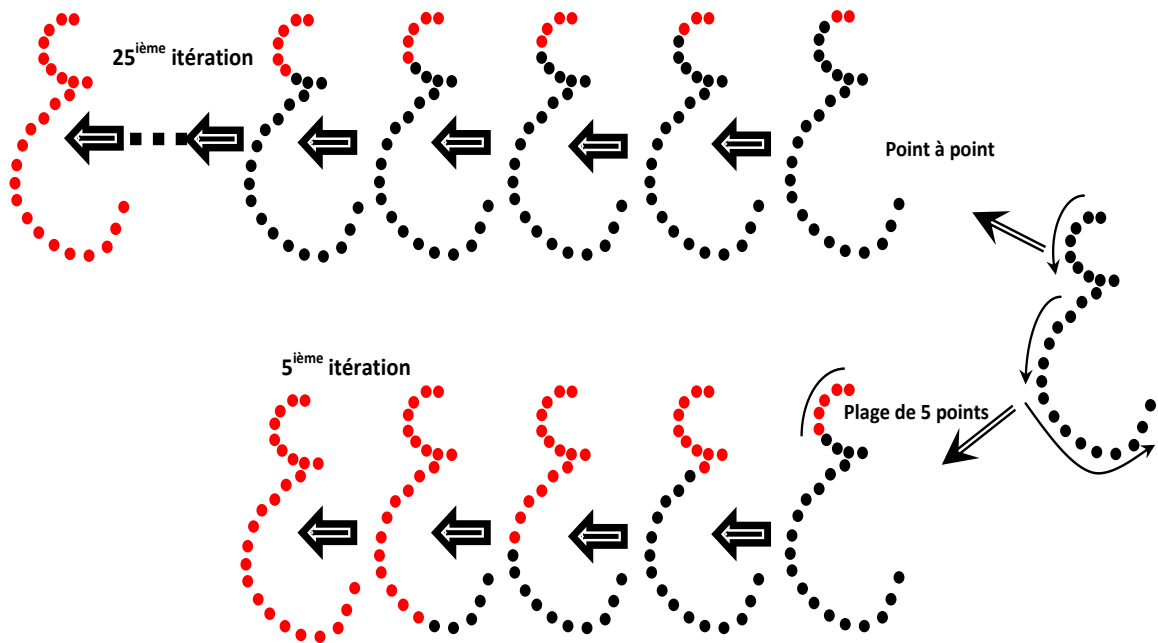


Figure IV.10 : *Algorithme de recherche dynamique (ARD) par plage de points*

IV.3. Vérifieur Neuro-génétique

La méthode implémentée dans cette thèse est un modèle probabiliste qui se résume au concept des réseaux bayésiens dynamiques et après avoir obtenu la segmentation la plus probable γ^* , une adaptation des valeurs dans le vecteur des vraisemblances est nécessaire,

les chercheurs en reconnaissance des formes utilisent souvent un vérifieur à ce niveau pour améliorer le système.

Les réseaux de neurones (perceptrons multicouches) sont les classificateurs les plus utilisés dans le domaine de l'intelligence artificielle. Ce mécanisme discriminant a été intégré comme outil de vérification des formes.

IV.3.1 Perceptron Multi Couches

Le Perceptron Multi-Couches (PMC) est le type de réseaux de neurones le plus répandu et le plus utilisé, vu la simplicité de sa structure et la rapidité de son apprentissage. Dans le passé, les PMC étaient peu utilisés, à cause du manque de règles rigoureuses et d'algorithmes efficaces pour gérer la phase d'apprentissage. un perceptron à deux couches seulement a été utilisé, ces derniers peuvent être entraînés avec des règles d'apprentissage relativement simples.

IV.3.1.1 Architecture

Le perceptron multicouche 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
- une ou plusieurs couches cachées (dans notre cas une seule)

Chaque neurone n'est relié qu'aux neurones des couches précédentes (voir figure IV.11)

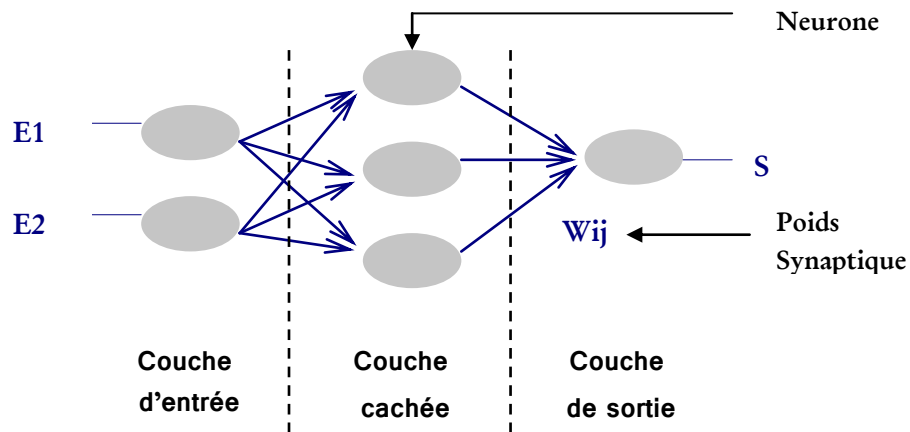


Figure IV.11 : Exemple d'un Perceptron Multi Couches

IV.3.1.2 Le neurone Formel

Le neurone formel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones en amont. A chacune de ces entrées, est associé un poids w (abréviation de weight (poids en anglais) représentatif de la force de la connexion (voir figure IV.12). Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones en aval. A chaque connexion est associé un poids.[31]

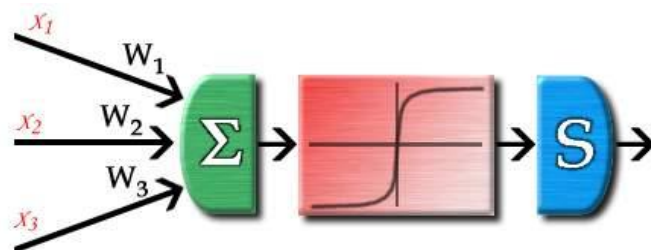


Figure IV.12 : Le neurone formel

- x_1, x_2, x_3 sont les entrées, ou encore les excitations.
- W_1, W_2, W_3 sont appelés les coefficients synaptiques ou poids synaptiques.

Nous utilisons des poids différents pour $W_1, W_2, W_3 \dots$ qui peuvent même être "corrigés" dans le cas de l'apprentissage. Le potentiel P est donné par :

$$P = x_1 * W_1 + x_2 * W_2 + x_3 * W_3 + \dots + x_n * W_n \quad \text{IV-16}$$

La fonction de sortie est $S = f(P)$ où f est la fonction d'activation.

IV.3.1.3 Fonctions d'activation

Un PMC est constitué de cellules, ou neurones. Les neurones sont reliés entre elles par des connexions, ce qui leur permet de disposer d'un canal pour envoyer et recevoir des signaux en provenance d'autres cellules du réseau. Chacune de ces connexions reçoit une valeur, un poids (une pondération en quelque sorte). Les neurones disposent d'une entrée, qui permet de recevoir de l'information d'autres cellules. Elles disposent aussi d'une fonction d'activation, qui est dans le cas le plus simple, une simple identité du résultat obtenu par l'entrée.

Pour un PMC avec neurones dans la première couche, l'entrée d'un neurone de la seconde couche sera généralement une somme pondérée des valeurs de sortie des neurones précédents :

$$X = a_1 * C(1) + a_2 * C(2) + a_3 * C(3) + \dots + a_n * C(n) \quad \text{IV-17}$$

Peut utiliser une fonction d'activation identité du type : $Y = d * X$

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 et souvent des fonctions non linéaires et plus évoluées seront nécessaires. A titre illustratif voici quelques fonctions couramment utilisées comme fonctions d'activation :

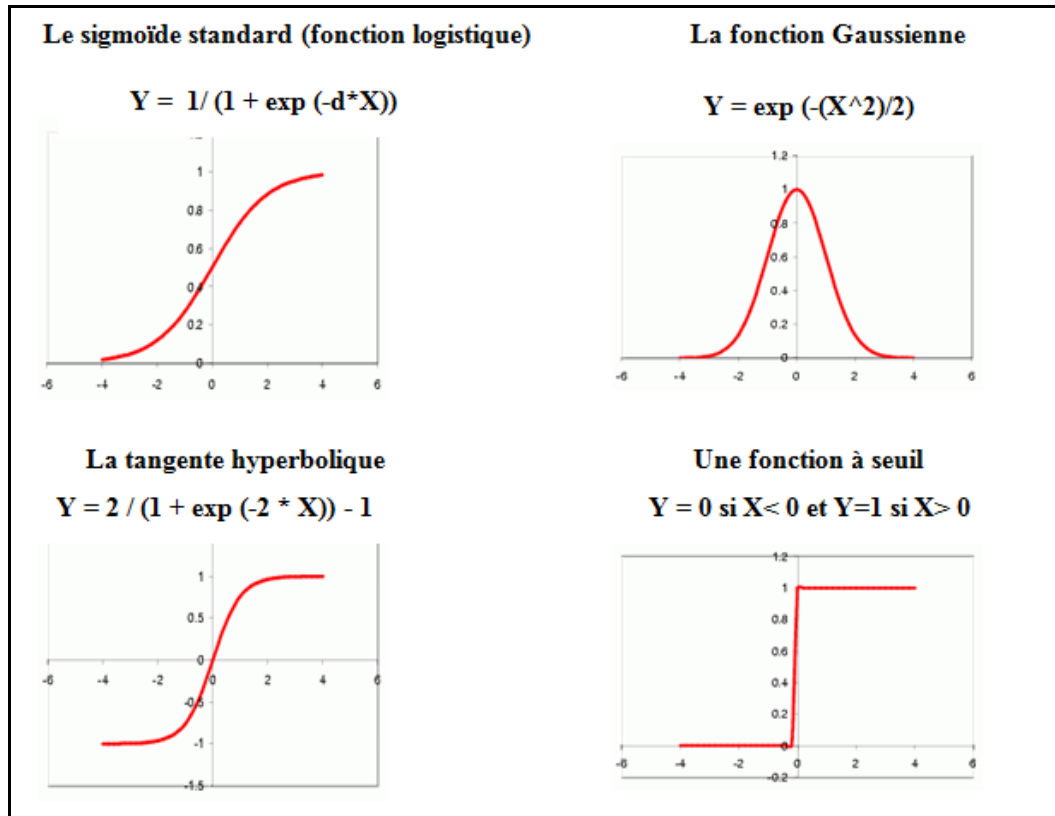


Figure IV.13 : Les fonctions d'activation les plus utilisées

IV.3.1.3 La Rétro Propagation du Gradient

C'est un algorithme d'apprentissage pour un réseau de type PMC

Pour un ensemble d'éléments (d'apprentissage), trouver les poids du réseau PMC qui fournissent la réponse attendue pour chacun des éléments.

Le traitement d'un élément se fait en 2 passes :

- **Passé avant** : on présente l'élément à l'entrée du réseau de neurones et on obtient le résultat en sortie
- **Passé arrière** : on calcule l'erreur de ce résultat par rapport au résultat attendu pour chaque neurone de sortie et on répercute cette erreur aux neurones des couches internes avec un algorithme de type « gradient-descent ».

Les poids ne sont pas adaptés après chaque élément, on se contente d'additionner les gradients pour chaque neurone, afin d'avoir une vue globale de l'erreur pour l'ensemble.

A la fin d'une époque, les poids sont adaptés.

Le gradient pour un neurone est l'erreur relative à ce neurone. Il peut en quelque sorte être vu comme la contribution du neurone à l'erreur globale

A chaque passe arrière, on calcule le gradient de chaque neurone, en commençant par ceux de la couche de sortie (car les gradients des couches inférieures se calculent à partir des gradients des couches supérieures).

La rétro-propagation du gradient utilise l'erreur quadratique, a_i est l'activation et d_i est la valeur désirée du neurone de sortie i , qui est donnée par :[31]

$$E = \frac{1}{2} \sum_i (d_i - a_i)^2 \quad \text{IV-18}$$

Les poids sont modifiés en y ajoutant la valeur ΔW_{ij} où h est le pas d'apprentissage.

$$\Delta W_{ij} = -h \frac{\partial E}{\partial w_{ij}} \quad \text{IV-19}$$

Le calcul du gradient d'erreur par rapport aux poids :

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial net_i} \cdot \frac{\partial net_i}{\partial w_{ij}} \quad \text{IV-20}$$

Où net_i est la somme de l'activité pondérée qui converge sur le neurone i .

Pour la couche externe (la couche de sortie), cette valeur se calcule de la manière suivante :

$$\frac{\partial E}{\partial net_i} = \frac{\partial}{\partial net_i} \left[\frac{1}{2} \sum_i (d_i - f(net_i))^2 \right] = -(d_i - a_i) \cdot f'(net_i) \quad \text{IV-21}$$

Où $f(net_i)$ est la fonction d'activation du neurone i .

$$\frac{\partial net_i}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \left[\sum_k w_{ik} \cdot a_k \right] = a_j \quad \text{IV-22}$$

Pour les neurones de la couche interne (la couche cachée) :

$$\frac{\partial E}{\partial net_i} = \left(\sum_k \frac{\partial E}{\partial net_k} \cdot \frac{\partial net_k}{\partial a_i} \right) \cdot \frac{\partial a_i}{\partial net_i} = \left(\sum_k \frac{\partial E}{\partial net_k} \cdot w_{kj} \right) \cdot f'(net_i) \quad \text{IV-23}$$

L'indice k parcourt tous les neurones des couches supérieures à celle du neurone i . A partir de ce calcul, on détermine la modification à apporter aux poids du réseau :

$$\Delta w_{ij} = -h \frac{\partial E}{\partial net_i} \frac{\partial net_i}{\partial w_{ij}} \quad \text{IV-24}$$

Et les liens du réseau doivent être corrigés comme suit :

$$\Delta w_{ij} = -h \cdot e_i \cdot a_j \quad \text{IV-25}$$

Où e_i représente la valeur accréditée du neurone i de l'erreur de j .

Pour la couche externe, l'erreur se calcule :

$$e_i = -(d_i - a_i) \cdot f'(net_i) \quad \text{IV-26}$$

Pour l'autre couche, elle se calcule :

$$e_i = \left(\sum_k e_k \cdot w_{ki} \right) \cdot f'(net_i) \quad \text{IV-27}$$

Sachant que e_k est l'erreur précédemment calculée pour le neurone k .

IV.3.1.4 Problème des Minima locaux

La plupart des problèmes peuvent être résolus par une méthode de type recherche locale. C'est le cas de l'apprentissage par rétro-propagation du gradient. Partant d'une configuration de poids initiale, la méthode va chercher la meilleure solution dans le voisinage de cette configuration.

Cette solution est optimale localement mais peut ne pas correspondre à un optimal global car il est possible qu'il existe une meilleure solution qui n'est pas dans le voisinage de la configuration initiale. La figure suivante illustre ce type de problème.

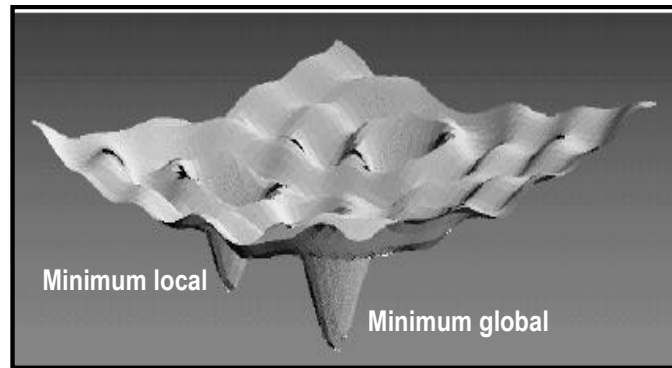


Figure IV.14 : *Convergence vers des erreurs correspondant à des minima locaux*

L'algorithme de rétro propagation du gradient est un outil performant et simple mais les études pratiques ont montré que son application appelle les remarques suivantes :

- Il n'existe pas de théorème garantissant la convergence du réseau, et l'on se contentera le plus souvent d'une erreur faible. Donc la valeur initiale des poids influe sur la solution finale.
- Le temps d'apprentissage est relativement long.

Les méthodes de descente de gradient sont sujettes à des variations de performances dues à la position initiale des poids du PMC menant parfois à une convergence vers des minima locaux. Les Algorithmes Génétiques (AG), à l'opposé, assurent une recherche dans le domaine complet. Au fur et à mesure des générations, cet espace de recherche est affiné vers des sous-espaces potentiellement performants. Cependant, il est courant pour les AG de trouver une solution proche de la meilleure sans jamais l'atteindre. On peut supposer que ces deux méthodes sont complémentaires. La section suivante est consacré à définir le concept des AG comme approche d'optimisation naturel.

IV.3.2 Les Algorithmes Génétiques

Les algorithmes de calcul évolutifs tels que les AG, constituent une famille de méthodes stochastiques d'optimisation, c'est à dire reposant sur des tirages aléatoires. Ce type d'algorithmes est de plus en plus utilisé pour trois principales raisons :

Ces algorithmes sont applicables dans un contexte beaucoup plus vaste que les méthodes d'optimisation classiques. Les fonctions à optimiser peuvent être non dérivables, non continues ou même non calculables (ce sont alors des programmes).

Les calculateurs récents sont de plus en plus puissants et rapides. Il devient donc possible de manipuler ces algorithmes dans des temps de calculs raisonnables d'autre part, ces méthodes sont en générale aisément parallélisables.

Les analogies entre ces méthodes et la biologie sont intéressantes à plusieurs titres. Elles sont une bonne source d'inspiration pour concevoir ou améliorer les algorithmes actuels. En retour, l'observation et l'amélioration des algorithmes actuelles peuvent apporter un enseignement dans le domaine de la biologie.[32]

IV.3.2.1 Cycle de l'évolution

L'évolution biologique a engendré des systèmes vivants extrêmement complexes. Elle est le fruit d'une altération progressive et continue des êtres vivants au cours des générations et s'opère en deux étapes : la sélection et la reproduction.

La sélection naturelle est le mécanisme central qui opère au niveau des populations, en sélectionnant les individus les mieux adaptés à leur environnement. La reproduction implique une mémoire : l'hérédité, sous la forme de gènes. Ce matériel héréditaire subit, au niveau moléculaire, des modifications constantes par mutations et recombinaisons, aboutissant ainsi à une grande diversité. (Voir figure IV.15)[33]

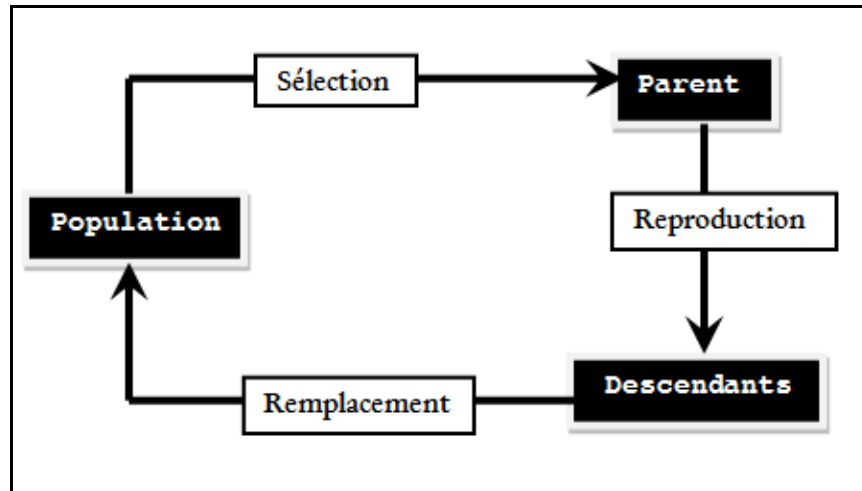


Figure IV.15 : Cycle de l'évolution

IV.3.2.2 Définition

Les algorithmes génétiques sont basés sur une analogie avec le phénomène de l'évolution et de la sélection naturelle. Le pionnier du domaine est John Holland qui a le premier tenté d'implanter artificiellement des systèmes évolutifs basés sur le processus de la sélection naturelle. Ainsi, les algorithmes génétiques sont une abstraction de la théorie de l'évolution. L'idée fondatrice est simple : si l'évolution a optimisé les processus biologiques, l'utilisation du terme de l'évolution pour trouver des solutions optimales dont le cadre de traitements informatiques possède un sens [32]

Les algorithmes génétiques (AG) sont des systèmes qui s'appuient sur les principes de sélection de Darwin et sur les méthodes de combinaison des gènes introduites par Mendel pour traiter des problèmes d'optimisation.

Ils peuvent surpasser d'autres méthodes classiques avec leur robustesse et sont fondamentalement différentes selon quatre axes principaux [32]:

- Les AG utilisent un codage des paramètres, et non les paramètres eux-mêmes.
- Les AG travaillent sur une population de points, au lieu d'un point unique.
- Les AG n'utilisent que les valeurs de la fonction, pas sa dérivée, ou une autre connaissance auxiliaire.
- Les AG utilisent des règles de transition probabilistes, et non déterministes.

IV.3.2.3 Le mécanisme

L'évolution se produit sur des chromosomes qui représentent chacun des individus dans une population. Le processus de sélection naturelle fait en sorte que les individus les mieux adaptés se reproduisent plus souvent et contribuent davantage aux populations futures (reproduction).

Lors de la reproduction, l'information contenue dans les individus des parents est combinée et mélangée pour produire les individus des enfants (croisement).

Le résultat de croisement peut à son tour être modifié par des perturbations aléatoires (mutations).

Donc un AG simple se compose en 3 opérateurs : reproduction, croisement et mutation.

[33]

a) Reproduction : Consiste à sélectionner les individus proportionnellement à leur performance. Un individu ayant une forte valeur d'adaptation a alors plus de chances d'être sélectionné qu'un individu mal adapté à l'environnement. La fonction d'adaptation ou bien « Fitness » est une mesure de qualité que l'on souhaite maximiser et elle est définie sur \mathbb{R}^+ . L'opérateur de reproduction peut être mis en œuvre sous forme algorithmique de différentes façons. On peut créer une roue de loterie pour laquelle chaque individu de la population occupe une section de la roue proportionnelle à son adaptation (voir figure IV.16). Il suffit de tourner la roue biaisée N fois (tel que N est la taille de la population) pour réaliser l'opération.

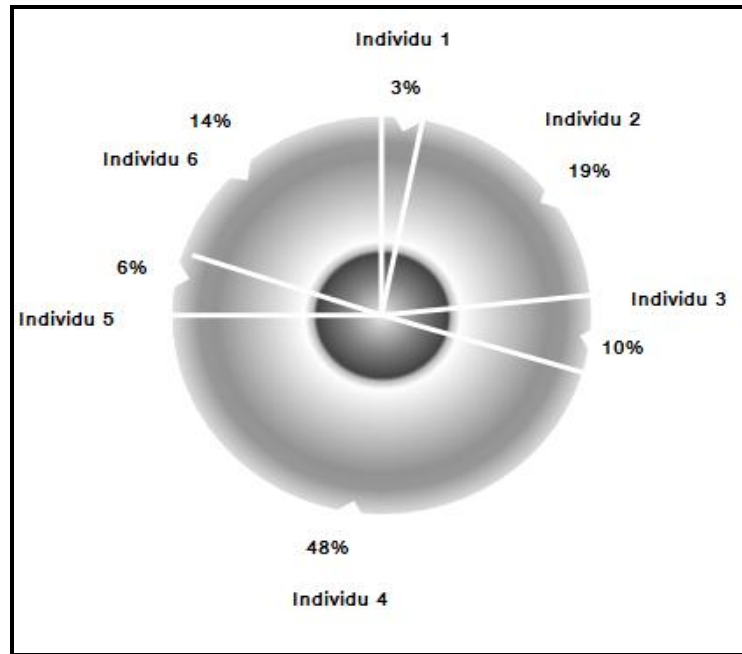


Figure IV.16 : *Modèle de roulette pour une population de 6 individus*

b) Croisement : Généralement, cet opérateur est vu comme l'opérateur de recherche essentiel d'un AG. Il consiste à sélectionner aléatoirement une position de croisement (locus) et de permuter les parties droites des deux parents. On distingue ici la probabilité de croisement P_c . Il existe deux types de croisement : Croisement à 1 seul locus : c'est le plus utilisé, il contient un seul point de croisement et il ne permute que deux parties des deux parents (voir figure IV.17). Croisement à plusieurs locus (uniforme) : peut être vu comme un croisement multipoint dont le nombre de locus est indéterminé a priori. Réellement, on utilise un « masque de croisement », qui est une chaîne binaire de même longueur que les individus. Un « 0 » à la $n^{\text{ième}}$ position du masque laisse inchangé les symboles à la $n^{\text{ième}}$ position des deux individus, un « 1 » déclenche un échange des symboles correspondants. Le masque est engendré aléatoirement pour chaque couple d'individus. (voir figure IV.18).

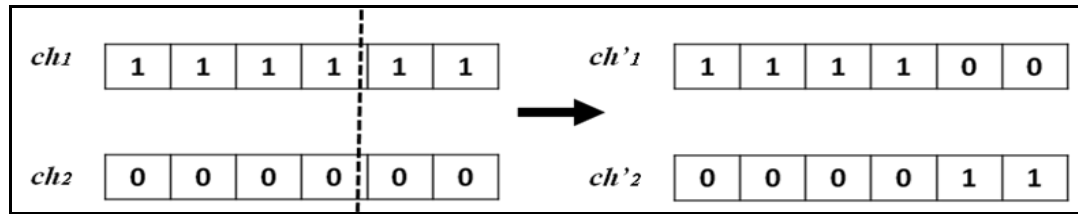


Figure IV.17 : Croisement à un seul locus

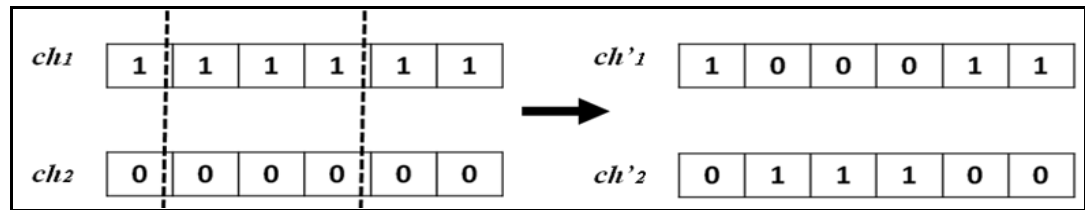


Figure IV.18 : Croisement à deux locus

c) Mutation : Consiste à prendre des gènes aléatoires dans le chromosome et à les altérer selon leur type. Un gène binaire i est muté en non(i) (voir figure IV.19). On distingue ici la probabilité qu'un gène peut être muté dans un chromosome « P_m ». Typiquement P_m est de faible valeur de 0.01 à 0.001 donc il a une faible influence en terme d'optimisation mais d'autre part la mutation assure une recherche locale aléatoire autour de chaque individu.

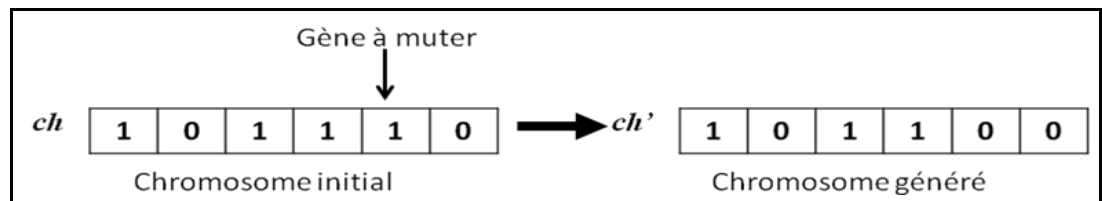


Figure IV.19 : opérateur de Mutation

IV.3.2.4 Principes généraux et paramètres nécessaires

a) Le codage des individus : Pour adapter un AG à un problème particulier, il faut d'abord définir une méthode de codage pour les solutions. Traditionnellement, les premiers algorithmes génétiques utilisaient une représentation binaire où chaque solution est codée sous la forme d'une chaîne de bits: [32]. Si cette représentation est tout à fait appropriée pour certains problèmes d'optimisation, il est maintenant courant d'utiliser d'autres représentations qui tirent partie de la structure naturelle des solutions comme le codage réel par exemple.

b) La fonction à optimiser : L'objectif est exprimé plus naturellement sous la forme de la minimisation d'une fonction de coût $g(x)$ plutôt que la maximisation d'une fonction d'utilité ou de profit $u(x)$. Même si le problème est exprimé sous la forme d'une maximisation, cela ne garantit pas que la fonction d'adaptation doit être une mesure de mérite positive. Aussi, il est souvent nécessaire de transformer la fonction à optimiser d'origine en une fonction d'adaptation, en une ou plusieurs étapes. La dualité entre la minimisation du coût et la maximisation du profit est bien connue. En recherche opérationnelle classique, pour transformer un problème de minimisation en un problème de maximisation, on multiplie tout simplement la fonction de coût par -1. Avec les AG, cette opération seule est insuffisante car la fonction ainsi obtenue n'est pas nécessairement positive sur tout son domaine. Avec les AG, les transformations qui suivent sont souvent utilisées :

$$F(x) = \begin{cases} C_{\max} - g(x) & \text{quand } g(x) < C_{\max} \\ 0 & \text{sinon} \end{cases} \quad \text{IV-28}$$

On affecte à C_{\max} la plus grande valeur de g observée. Mais comme on a déjà dit, il est cependant possible d'avoir des problèmes avec des valeurs négatives de la fonction d'utilité $u(x)$. Pour palier cela, il suffit de transformer la fonction d'adaptation de la façon suivante :

$$F(x) = \begin{cases} u(x) + C_{\max} & \text{quand } u(x) + C_{\min} > 0 \\ 0 & \text{sinon} \end{cases} \quad \text{IV-29}$$

On peut prendre C_{min} comme la valeur absolue de la plus faible valeur de u .

c) La Convergence : La convergence peut être mesurée par la performance statistique : si à une certaine génération, une forte concentration d'individus s'agglutine autour du meilleur (étude de la dispersion), on peut conclure à la convergence. On peut aussi l'observer en mesurant comment les individus changent à chaque position de gène. Un gène est dit avoir convergé lorsque 95% de la population partage la même valeur. La population est qualifiée d'avoir convergé si tous les gènes ont convergé.

d) Taille de la population : Pour appliquer un algorithme génétique avec une convergence optimale, la population doit contenir des individus qui représentent au mieux l'ensemble du domaine de recherche, pour cela la taille de la population doit être suffisamment grande.

e) Critère d'arrêt : Si le temps alloué pour un AG est dépassé ou bien le nombre de générations maximal est atteint, le processus s'arrête et on obtient la dernière génération comme résultat final.

f) Probabilités des opérateurs génétiques : Dans un algorithme génétique simple les deux opérateurs : croisement et mutation ont comme paramètres d'entrée les deux probabilités P_c et P_m et ils doivent être définis avant le lancement de l'algorithme.

P_c : choisi généralement entre 0.6 et 1.

P_m : très faible entre 0.01 et 0.033.

IV.3.2.5 Avantages des AG

Contrairement à la recherche opérationnelle, l'AG n'exige aucune connaissance de la manière dont résoudre le problème. Il est seulement nécessaire de pouvoir évaluer la qualité de la solution. Également, dans le cas de recherche d'optimum de fonctions analytiques, on n'a besoin ni de dérivabilité ni de continuité.

La mise en œuvre d'un AG est aisée : le "moteur" est commun; il y a donc peu de programmation spécifique au problème. Aussi, plus le problème est complexe, plus l'AG montre sa supériorité.

La performance des AG dépend plus du codage et de la fonction d'adaptation que de l'influence des paramètres. Cependant, un choix judicieux de ces derniers est essentiel pour l'efficacité de l'algorithme.

IV.3.3 Hybridation Neuro-Génétique

IV.3.3.1 Modélisation

La réalisation de l'hybridation consiste à définir les différentes interfaces entre l'algorithme de rétro-propagation du gradient (recherche locale) et un AG (recherche globale) .

Il faut considérer le mot interface au sens large et comprendre « les différentes modifications nécessaires à l'AG pour manipuler des PMC ». Il faut en particulier déterminer un codage pour rendre le PMC manipulable par l'AG. Nous devons également définir les différentes étapes de l'AG et construire des opérateurs (croisement, mutation) adaptés aux types de codage de la structure donnée définie.

La première étape pour permettre cette manipulation est de définir sous quelle forme au sens structure de données, l'AG verra le PMC en tant qu'individu d'une population.

Considérons un réseau ayant trois couches de neurones avec deux neurones en entrée, deux en couche cachée et un en sortie. Soit W_{ij} le poids d'une connexion du neurone i vers le neurone j .

La figure IV.20 donne une représentation de ce *perceptron* et le *chromosome* associé.[34][35][36]

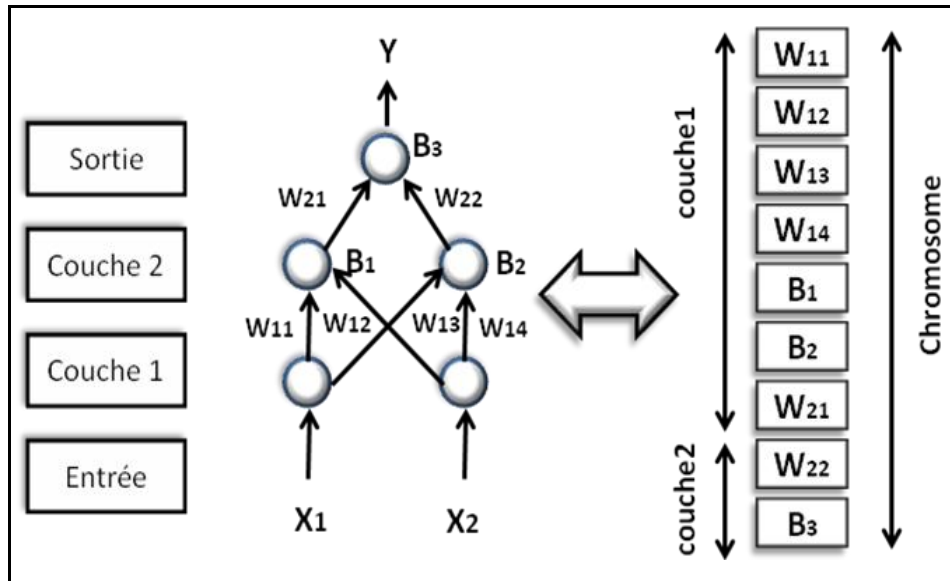


Figure IV.20 : La représentation génétique d'un PMC

IV.3.3.2 Adaptation « fitness »

L'adaptation d'un individu (appelée fitness) décrit l'adéquation de celui-ci à son environnement. Plus précisément, l'individu aura une adaptation d'autant moins élevée qu'il sera une bonne solution dans le problème de l'optimisation d'un NN. On cherche à quantifier la capacité d'un NN à apprendre. On estime que l'apprentissage est d'autant meilleur que le NN a une faible valeur, l'adaptation de l'individu sera donc l'erreur quadratique moyenne.[37]

IV.3.3.3 Codage « représentation des individus »

On est confronté ici au problème de choix entre une représentation binaire et une représentation réelle concernant le codage de l'individu.

Traditionnellement les AG travaillent sur des espaces de recherche binaire dont $[0,1]^n$.

Si l'on choisit une représentation binaire, il est nécessaire de décoder les poids en valeurs réelles avant chaque opération d'apprentissage ou de reconnaissance des PMCs.

IV.3.3.4 Politique de sélection « élitiste »

Nous avons testé différentes politiques de sélection, celle qui s'est révélée être la meilleure est la sélection élitiste. Cela signifie simplement que l'on applique la règle : «ce sont les plus forts qui survivent», une forte partie des meilleurs individus « survit » d'une génération à l'autre en pratique, à une époque t , on sélectionne une forte partie de la population (90% par exemple), cette partie est constituée des meilleurs individus, au sens du critère recherché (l'erreur quadratique moyenne). Le reste de la population sera remplacé par d'autres individus à l'époque $t+1$. Ces nouveaux individus sont obtenus par combinaison des individus sélectionnés.

Cette opération de combinaison est constituée de plusieurs sous opérations : croisement, mutation sont les plus courantes. La politique élitiste apporte généralement une vitesse de convergence assez grande par rapport à d'autres politiques.

IV.3.3.5 Algorithme hybride détaillé

L'algorithme neuro-génétique proposé a été détaillé dans la figure suivante : [35]

```

g := 0 ;
Initialisation d'un ensemble de poids aléatoire P(g) :=W(g)    %l'ensemble des poids synaptiques
Evaluation E(P(g) )                                         %erreur quadratique moyenne
While g < Maxg do                                       %Maxg le nombre maximum de génération
  For i=1,...,N/2 do                                     %N : la taille de Population P
    Sélectionner deux parents de la population P(g)         %Sélection
    Appliquer le croisement sur les deux parents avec une probabilité Pc %Croisement
    Appliquer la mutation sur les descendants avec une probabilité Pm %Mutation
    Evaluation E(P(g) )
    Insérer les deux nouveaux individus à P(g+1)           %Reproduction
  end
  g :=g+1                                                  %Incrémentation de générations
end

```

Figure IV.21 : Algorithme hybride neuro-génétique détaillé

IV.4. Conclusion

Pour faciliter davantage l'utilisation des ordinateurs, de nouvelles approches ont été proposées et créées. Elles n'utilisent plus le clavier ni la souris mais un stylet relié à une tablette électronique.

Dans la recherche de meilleurs algorithmes capables de résoudre une classe de données de problèmes, une tendance actuelle veut que l'on bâtit des méthodes complexes incorporant des connaissances et des techniques issues de plusieurs horizons.

Si l'on est exigeant sur la qualité de la solution, et que l'on souhaite un fonctionnement automatique, des critères pertinents sont traités et améliorés.

De par la littérature, grâce à leurs propriétés d'amélioration des performances, la combinaison des algorithmes génétiques avec les réseaux neuronaux nous offre un modèle hybride performant.

Chapitre V : RÉSULTATS, ANALYSE ET DISCUSSIONS

V.1 Introduction

Ce dernier chapitre est consacré pour présenter les outils matériels et logiciels utilisés dans la phase d'implémentation.

Ensuite, quatre expériences sont exposées en détail fournissant des résultats. Ces derniers sont discutés et comparés à la fin de cette section.

V.2 Dispositifs matériels

V.2.2 Ordinateur

Pour la réalisation des expériences, un ordinateur de bureau a été utilisé avec :

- Un processeur Pentium Core2Duo avec vitesse d'horloge 2.2 GHz
- Mémoire vive RAM installée 2 Go.

Le système d'exploitation utilisé est le Windows 7 (32 bits) édition intégrale.

Toutes les expériences sont développées sous MATLAB version 7.8.0.347 (R2009a).

V.2.3 Tablette PC

Pour la construction de la base de données « NOUN », une tablette WACOM BAMBOO version 5.08-6 a été exploitée et illustrée dans la figure suivante :

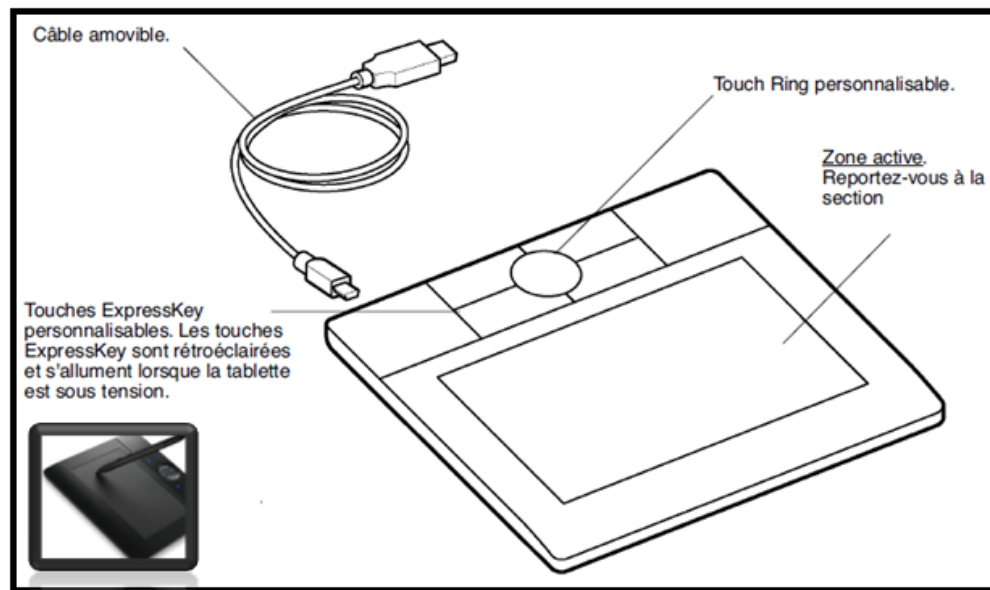


Figure V.1: *Fonctionnalités de la tablette WACOM*

La tablette dispose d'un stylet sans fil et sans pile, contribue à réaliser le travail en activant ses fonctions d'encrage numériques. Le stylet est un outil à main levée sensible à la pression. Le stylet permet d'effectuer quatre opérations de base : le pointage, le clic, le double-clic et le déplacement.

La figure V.2 montre les composants de ce stylet et le réglage de paramètres.

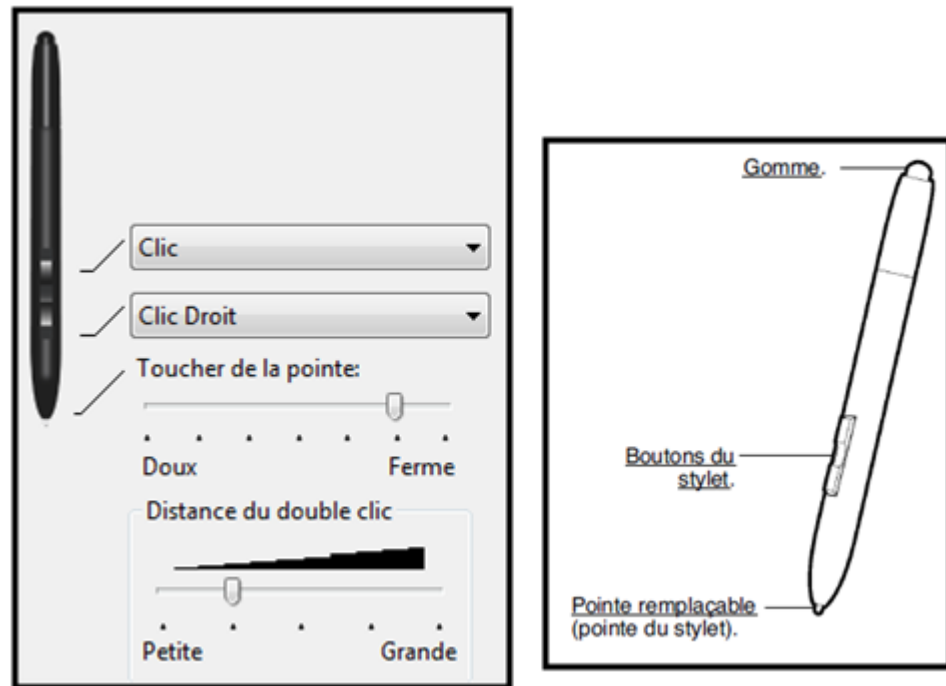


Figure V.2 : Composants de stylets et réglage de paramètres

Il faut tenir le stylet comme un crayon classique. Le stylet peut être incliné pour travailler. Les boutons de stylet sont positionnés à pouvoir appuyer facilement dessus avec la pouce ou l'index. (Voir figure V.3)

Avant l'utilisation de la tablette, il faut organiser l'espace de travail de manière à travailler confortablement. Il faut placer la tablette, le stylet et le clavier de façon qu'ils soient aisément accessibles. Il faut aussi placer le moniteur de manière à le voir avec un minimum de fatigue oculaire. Pour une utilisation optimale, la tablette doit être orientée de sorte que le curseur de l'écran se déplace dans le même sens de la main du scripteur. (Voir la figure V.4)

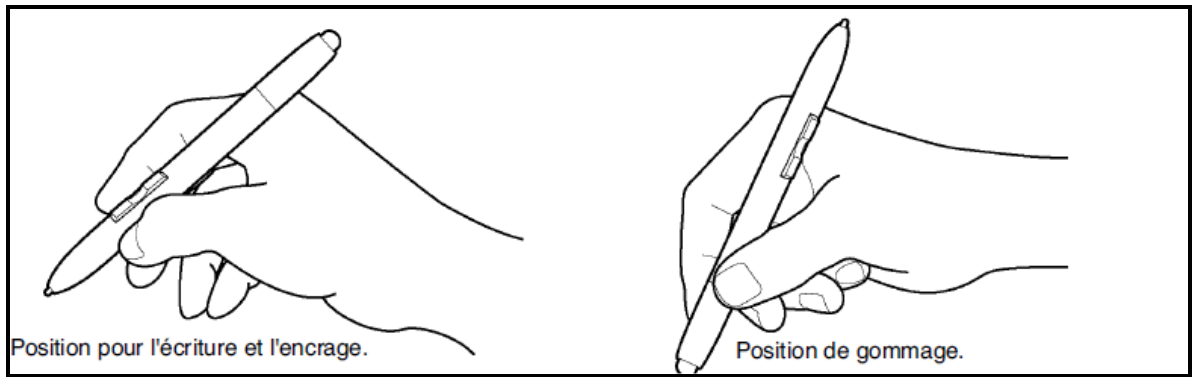


Figure V.3 : *Positionnement de stylet*

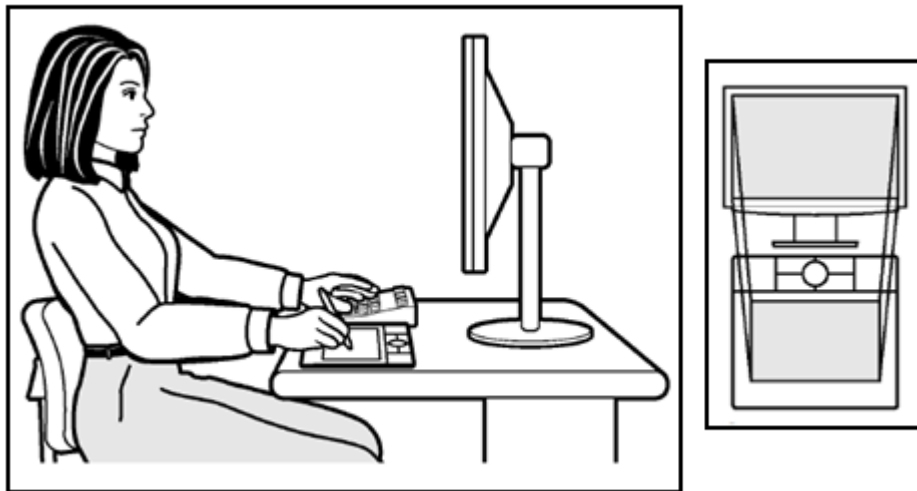


Figure V.4 : *Zone de travail sur tablette*

V.3 Base de données NOUN

V.3.1 Fiche d'écriture Scripteur

V.3.1.1 Alphabet

L'alphabet Arabe est composé de 28 lettres avec leurs formes isolées. Le scripteur est invité à saisir tout l'alphabet un nombre N d'occurrences. Il doit respecter la fiche mentionnée sur la figure suivante :

خ	ح	ج	ث	ت	ب	ا
Kha	Ha	Jim	Tha	Ta	Ba	Alef
خ	ح	ج	ث	ت	ب	ا
ص	ش	س	ز	ر	د	ذ
Sad	Chin	Sin	Zain	Ra	Dhal	Dal
ص	ش	س	ز	ر	د	ذ
ق	ف	ع	ع	ظ	ط	ظ
Qaf	Fa	Ghain	Am	Thah	Itah	Dhad
ق	ف	ع	ع	ظ	ط	ظ
ي	ه	و	ن	م	ل	ك
Ya	He	Waw	Noun	Mim	Lam	Kaf
ي	ه	و	ن	م	ل	ك

Figure V.5 : Alphabet saisie par le scripteur

V.3.1.2 Interface d'acquisition

L'interface de construction des échantillons de caractères arabes isolés est la suivante :

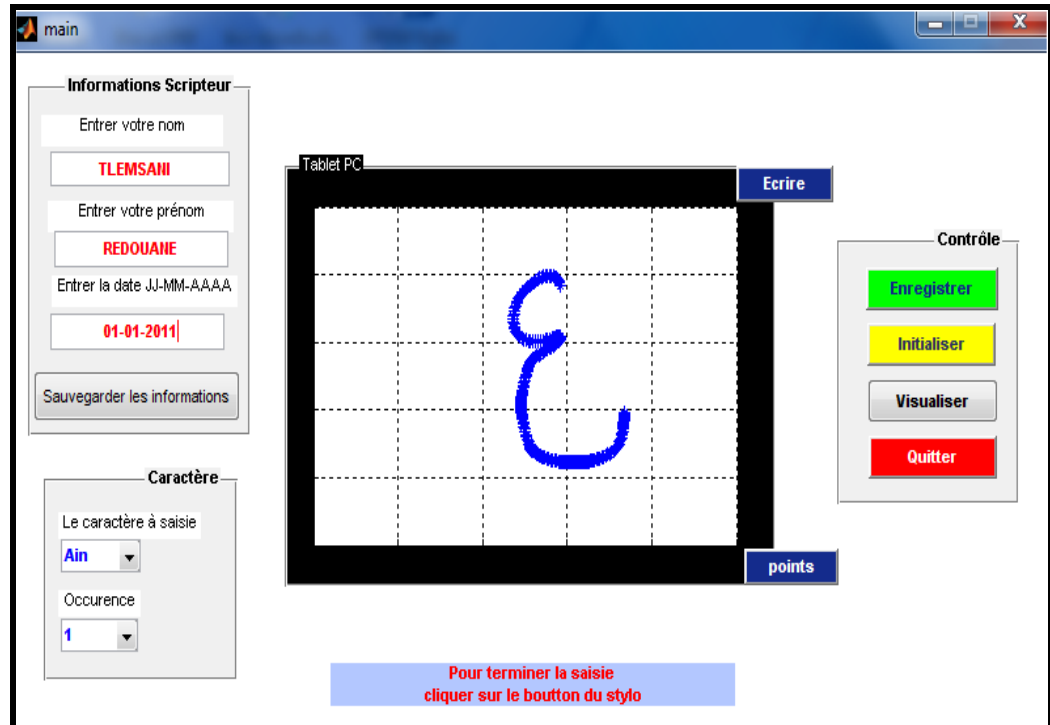


Figure V.6 : Interface d'acquisition

Le scripteur doit inscrire son *nom*, *prénom* et la *date de saisie* dans l'onglet « **Information Scripteur** » mentionné à gauche dans l'interface.

En bleu, les deux boutons : « *Écrire* » et « *points* » sont réservés pour saisir le caractère et les points des caractères respectivement.

Le scripteur doit faire une sélection du caractère à saisir et quelle occurrence sans l'onglet « **caractère** ».

L'onglet « **contrôle** » à droite sert à appliquer quelques contrôles utiles pour la construction des échantillons. En vert, le bouton « *enregistrer* » pour la sauvegarde, en jaune le bouton « *initialiser* » pour la remise à zéro de toutes les informations sur l'interface. En gris pour la visualisation du caractère sauvegardé et en rouge pour quitter l'interface.

V.2.2 Description de la base

La base de données NOUN dans sa première version est développée pour initier la recherche et le développement des systèmes de reconnaissance du manuscrit arabe en ligne. Elle contient 2800 caractères isolés, en arabe. Elle a été collectée auprès d'environ 25 scripteurs différents, chacun a écrit les lettres de l'alphabet arabe cinq fois. Cette base de données a été développée entre : le laboratoire SIMPA (Signal Image Parole) à l'Université des Sciences et de la Technologie d'Oran USTO-MB et le laboratoire LaRATIC (Laboratoire de Recherche Appliquée et des TICs) à l'Institut National des Télécommunications et des Technologies d'Information et de la Communication INTTIC.

V.3.3 Traitement et segmentation des caractères isolés:

Notre étude concerne la reconnaissance en-ligne c'est-à-dire acquise dynamiquement à l'aide d'une tablette graphique à digitaliser. Celle-ci possède une résolution précise et échantillonnée à une vitesse d'écriture choisie. Un temps d'adaptation est nécessaire au scripteur pour pouvoir écrire à peu près correctement. Malgré tout pour son traitement le caractère devra être segmenté en tracé c'est-à-dire en strokes élémentaires.

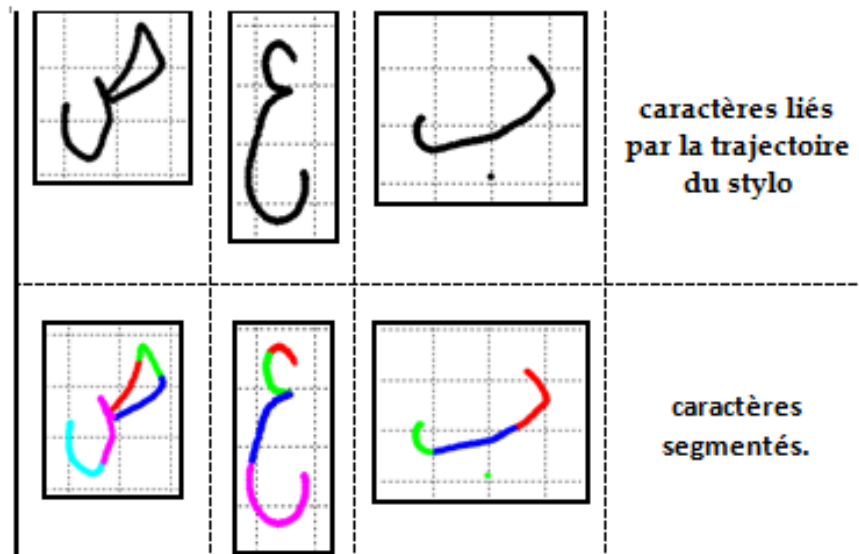


Figure V.7 : Traitement et segmentation des caractères « Ba », « Ain » et « Sad »

La figure ci-dessus montre les différents changements implémentés et appliqués par exemple sur trois caractères choisis.

V.4 Résultats et expériences :

Durant la recherche et la spécification de la problématique à traiter, une division en quatre expériences a été désignée pour cerner tous les paramètres pertinents et les approches implémentées citées dans le chapitre précédent.

V.4.1 Expérience 1 : DBN

La figure V.7 représente l'architecture globale du système bayésien implémenté pour l'apprentissage et la reconnaissance à partir de la base de données construite.

Les différents modules du système sont répartis comme suit:

1. Préparation et réorganisation des sous corpus
2. Phase d'apprentissage
3. Phase de reconnaissance

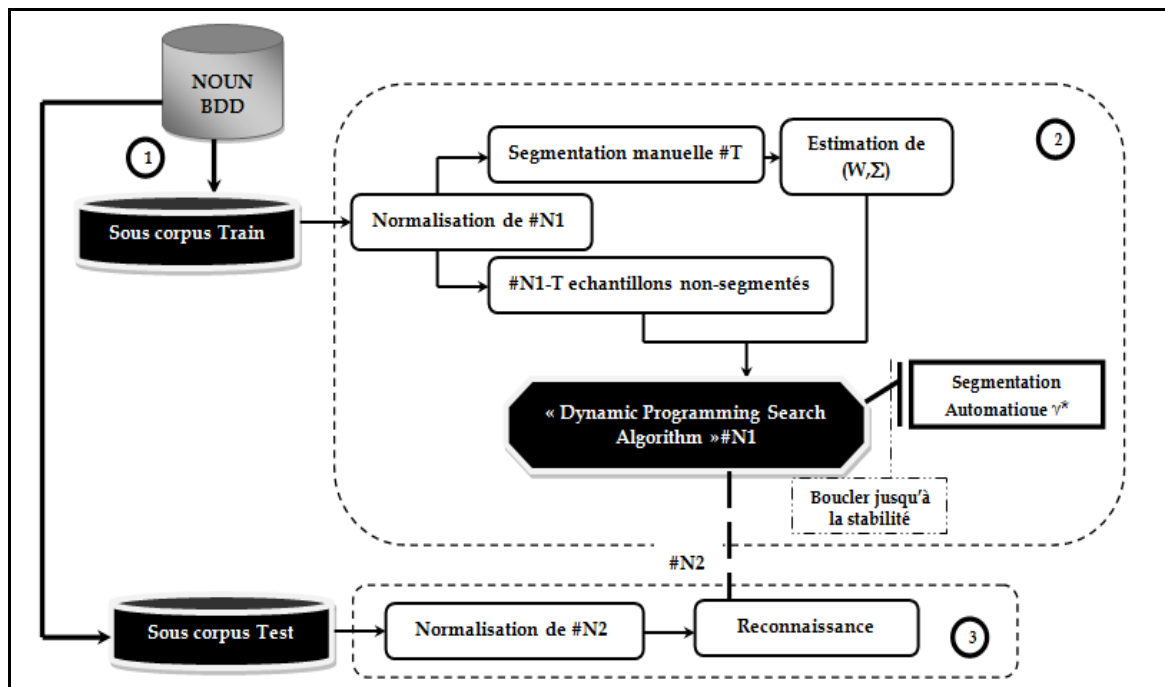


Figure V.8 : Schéma général du modèle implémenté

Pour nos expériences, on a implémenté et vérifié notre modèle bayésien sur un ensemble de lettres d’**Alef** à **Ya** de l’alphabet arabe. La table V.4 montre la matrice de confusion des taux de reconnaissance; ce sont des taux relativement importants sur un tel sous corpus. L’efficacité revient aux éléments spatiaux sauvegardés et aux vraisemblances des gaussiennes de chaque modèle de point. Le taux global de reconnaissance atteint **66.78%**.

V.4.4 Expérience 2: DBN profondeur influence

Avant toute amélioration, cette expérience a été réalisée, pour but de tester l’influence de la profondeur du réseau bayésien d sur les différents taux de reconnaissance sur chaque caractère. Les profondeurs $d=2$, $d=3$ et $d=4$ ont été prises à base d’augmentation de complexité comme montré dans la figure suivante :

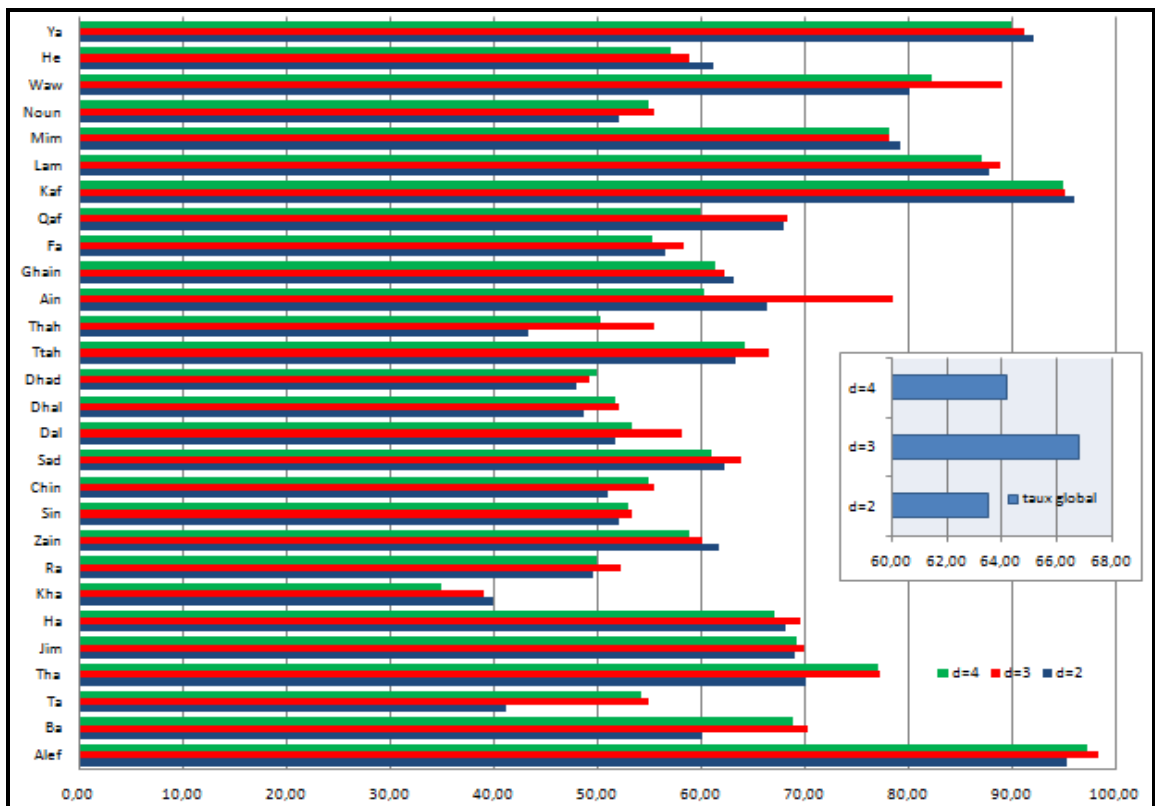


Figure V.9 : *histogramme de variation de taux de reconnaissance par rapport la profondeur d*

V.4.3 Expérience 3 : DBN-NNGA

La méthode implémentée dans l'expérience 1 est un modèle probabiliste qui se résume au concept des réseaux bayésiens dynamiques et après avoir obtenu γ^* le vecteur des vraisemblances par l'algorithme de programmation dynamique, au lieu de prendre la valeur maximale, les chercheurs en reconnaissance des formes utilisent souvent un vérifieur à ce niveau pour adapter le système.

Les réseaux de neurones sont les classificateurs les plus utilisés dans le domaine de l'intelligence artificielle et on a tenté à intégrer ce mécanisme discriminant comme outil de vérification des formes. Les paramètres utilisés dans notre réseau de neurone sont les suivants :

Tableau V.1 : Paramètres neuronaux

Paramètres	Valeurs
Activation du neurone	sigmoïde
Nombre max d'itérations	100
Erreur désirée	0.001
Pas d'apprentissage	0.7

L'architecture neuronale utilisée est montrée dans la figure ci-dessous (28 neurones en entrée, 16 dans la couche cachée et 28 neurones dans la couche de sortie), en entrée affectation des vecteurs des vraisemblances obtenus par le réseau bayésien dynamique. En sortie un vecteur binaire indique la classe de caractère associé.

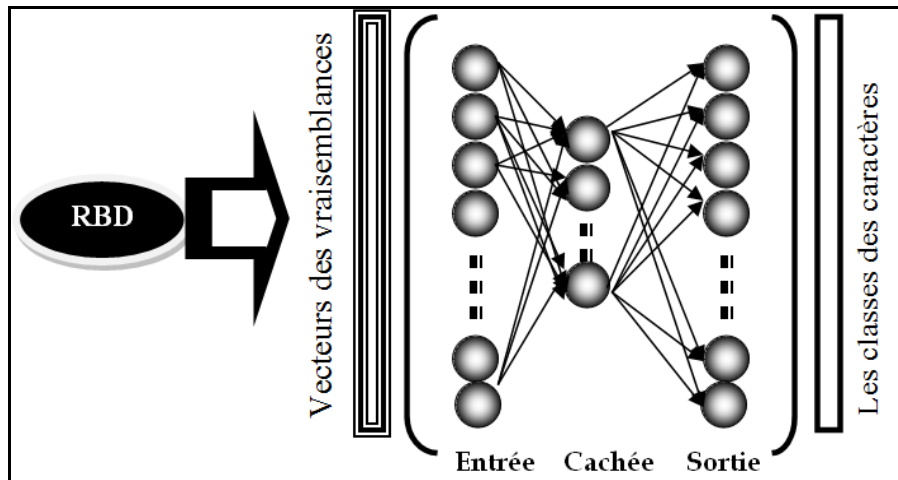


Figure V.10 : Vérifieur neuronale

Dans la figure V.9, on remarque clairement la convergence de l'erreur quadratique moyenne obtenue à partir d'un apprentissage neuronale.

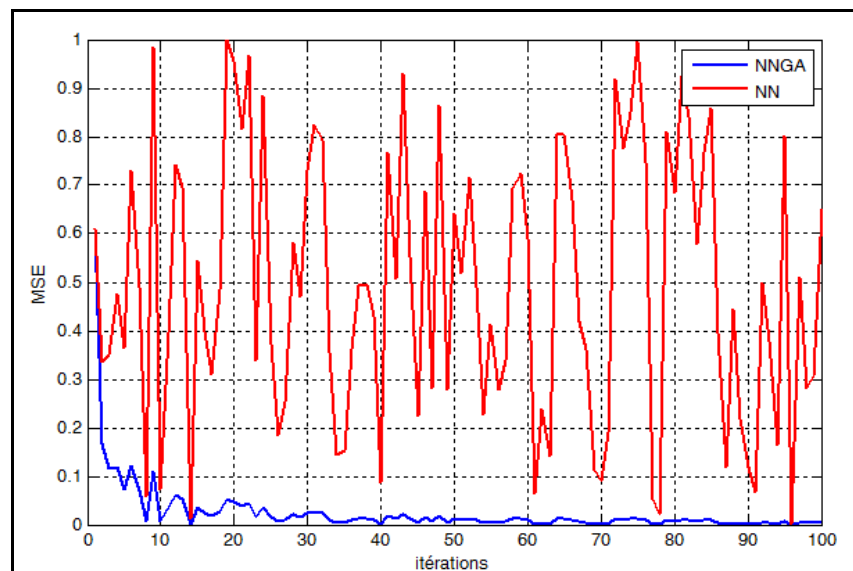


Figure V.11 : graphe de l'évolution de l'erreur par itération

Par la suite, une amélioration génétique a été appliquée sur les poids synaptiques des neurones associés en fixant les paramètres suivants :

Tableau V.2 *Paramètre génétique*

<i>Paramètres</i>	<i>Valeur</i>
Opérateur de sélection	roulette
Croisement	Pc=0.7 à un seul locus
Sélection avec élitisme	el=0.9
Mutation	Pm=0.01 gaussienne
Taille de population	40
Nombre maximum de génération	1000

Sur le même graphe, la convergence a été meilleure en appliquant l'optimisation génétique sur ce vérifieur et cette approche a vraiment donnée des résultats remarquables. La table V.5 montre la matrice de confusion des taux de reconnaissance pour le modèle bayésien avec un vérifieur neuro-génétique (NNGA) et on a obtenu un taux plus élevé **73,47%** c'est-à-dire une augmentation de **6,69%**. Ce résultat démontre l'efficacité d'utilisation un vérifieur combiné au modèle implémenté.

V.4.4 Expérience 4 : DBN Time Segmentation « plage de points »

On a implémenté les trois approches (ARD, ARD par plage de points statiques et ARD par plage de points dynamiques) appliquées sur l'ensemble des caractères d'Alef à Ya et en sortie on a obtenue le temps de segmentation de l'observation indiquée (voir table V.3).

Le résultat montre efficacement que la méthode de segmentation basée sur la plage de points statiques est plus rapide par rapport à l'ancien algorithme ARD. Aussi l'approche dynamique de la segmentation par plage de points a montré une rapidité mieux remarquée globalement. On peut démontrer cette efficacité à cause du concept de base expliqué dans le chapitre précédent, qui donne une possibilité de partition par un ensemble de points et non pas un point unique. La matrice de confusion confirme cette amélioration. (voir tables V.6).

Tableau V.3 : Temps de segmentation des observations pour les trois algorithmes

Alphabet	Temps de segmentation (exprimé en seconde)		
	ARD	ARD Statique	ARD Dynamique
Alef	73.472	4.062	1.367
Ba	186.765	8.964	1.562
Ta	37.797	2.329	1.407
Tha	22.468	1.625	1.172
Jim	77.718	4.235	1.735
Ha	32.078	2.063	1.641
Kha	71.485	3.766	1.781
Dal	88.882	4.744	1.484
Dhal	55.233	3.372	1.272
Ra	45.203	2.375	1.375
Zain	39.195	2.329	1.407
Sin	11.152	1.621	1.172
Chin	77.718	4.231	1.735
Sad	98.071	2.011	1.666
Dhad	188.063	8.878	1.547
Ttah	244.907	11.401	1.9793
Thah	245.161	11.522	2.215
Ain	192.163	8.913	1.527
Ghain	245.366	11.502	2.415
Fa	103.679	5.430	1.542
Qaf	235.998	11.062	1.575
Kaf	161.550	7.898	1.370
Lam	140.787	7.058	1.440
Mim	322.879	17.434	1.526
Noun	46.999	2.763	1.321
Waw	55.221	2.663	1.451
He	71.755	3.923	1.417
Ya	319.061	14.606	1.693

V.4.5 Comparaison et Discussion

La première à réaliser est la construction de base contenant des échantillons de lettres arabes. En effet, dans notre première version de base de données, une stratégie minimale a été effectuée pour réaliser le système de reconnaissance par la suite. Des boîtes de bondissement normalisées ont été préparées pour faciliter la tâche au scripteur lors de la saisie.

Nous avons implémenté dans la première expérience notre modèle DBN non amélioré et les résultats ont été acceptables mais pas totalement fiables. Nous avons distingué par la suite

un paramètre pour traiter son influence sur l'apprentissage et le test relative qui est la profondeur du réseau relative à la complexité architecturale. Une valeur de $d=3$ a été sélectionnée dans la deuxième expérience comme valeur adéquate.

Pour l'apprentissage d'un système hybride DBN-NNGA, une amélioration remarquable à base de test a été citée. Ce qui nous démontre, l'utilité d'utilisation de vérifieur neuronale avec une optimisation de mécanisme interne par les algorithmes génétiques.

Une autre approche d'optimisation de notre système a été soulignée et exécutée au niveau de l'algorithme de segmentation elle-même. La segmentation par plage de points est un concept robuste pour un gain de temps lors de la réalisation informatique.

Dans un lieu de comparaison, la figure V.12 a bien montré la progression du système d'une amélioration à une autre en fixant la profondeur du réseau à $d=3$ comme meilleur paramètre de complexité de topologie. Le Modèle hybride DBN-NNGA a réalisé l'augmentation vénienne des taux de reconnaissances. Efficacement, Le modèle DBN-SPPD a repoussé les taux avec des pas justifiables et remarquables.

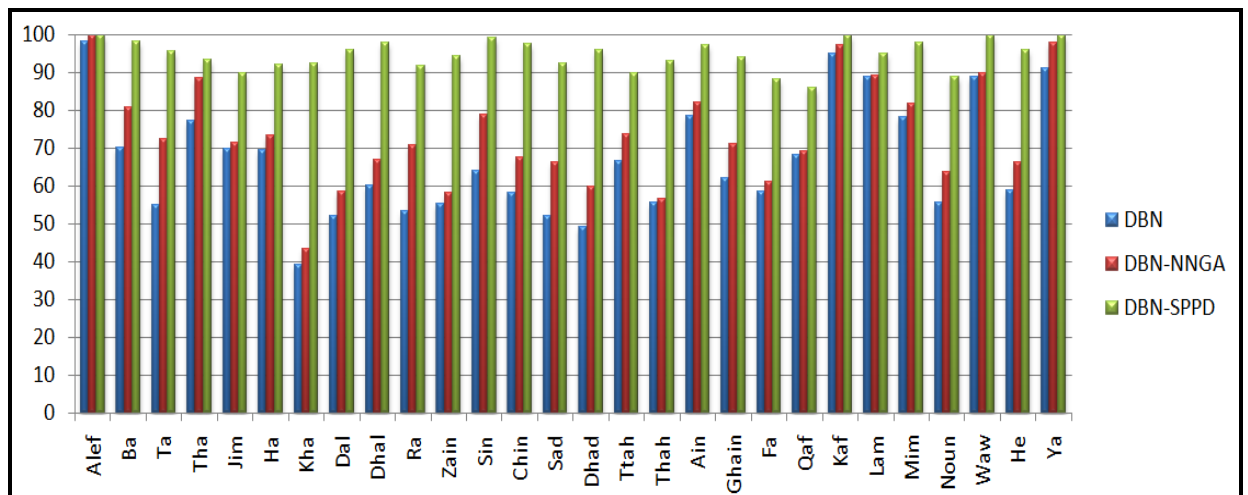


Figure V.12 : Taux de reconnaissance pour l'ensemble de caractères arabes en utilisant les trois approches d'amélioration implémentées

Nous avons également montré qu'une amélioration du système de reconnaissance, il était possible suivant le mécanisme du modèle et l'algorithme de recherche et de segmentation

élaborée. Le taux global atteint à proximité la totalité de tous les échantillons à une classification complète dans la dernière expérience comme montré dans la figure V.13.

Nos expériences valident l'utilisation des réseaux bayésiens dynamiques améliorées dans le cadre d'une tâche de reconnaissance de lettres arabes acquises en ligne.

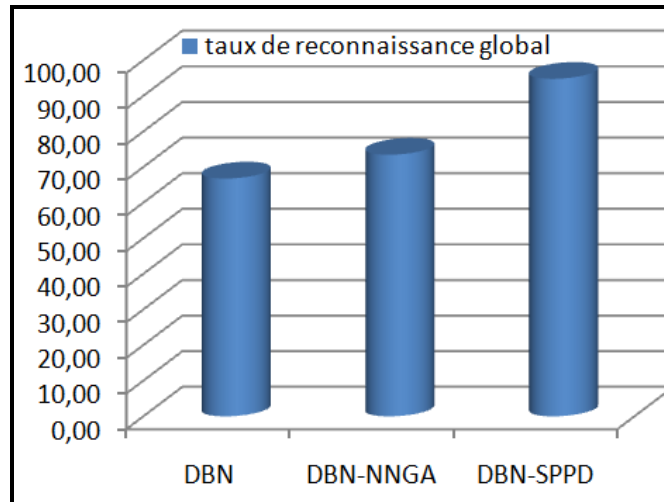


Figure V.13 : *Taux de reconnaissance global en utilisant les trois approches d'amélioration implémentées*

Les taux de reconnaissance obtenus dans la première expérience débute par 39,03% pour le caractère Kha et passe pour le caractère Qaf en moyenne en 68,32% et atteint le taux de reconnaissance maximum 98,33% pour le caractère Alef. Il est intéressant de noter qu'au niveau de la plus part des caractères le taux n'a pas dépassé les 70%. En effet du point de vue modélisation, l'approche acceptera d'autres ajouts.

Évidement et par la suite, nous avons pensé à une démarche hybride d'amélioration qui apparut efficace mais pas suffisante pour atteindre la fiabilité voulu réalisée au niveau de notre système.

Et elle est affichée avec des petites améliorations des taux par exemple : les caractères Jim, Fa et Waw donnent respectivement du DBN à DBN-NNGA des faibles augmentations 1,41%, 2,60% et 1,02%.

Cette insuffisance a poussée la recherche vers le mécanisme interne de ce réseau bayésien. Une amélioration du DBN-NNGA au DBN-SPPD atteint 49,10% pour le caractère Kha et moyenne par 21,32%. Cette idée d'optimisation au niveau de l'algorithme de segmentation

elle-même a été développée et elle fournie comme attendu des meilleurs résultats par rapport le modèle classique.

En général, les matrices de confusion dans les tableaux (V.4, V.5 et V.6) confirment pratiquement ces conclusions.

A la différence d'autres systèmes comme les travaux de N. Mezghani[1], le taux de reconnaissance global est égal à 95.17% (taux plus élevé au taux fournie par notre système qui égal à 94,79%) mais ce justifie que dans ces travaux, ils n'ont pas utilisé la totalité des caractères et ils ont utilisés les classes de caractères qui présentent la même forme géométrique comme le **Ba, Ta et Tha**. Dans notre système, nous avons gardé les caractères sans regroupement par forme et cela facilitera la tâche de reconnaissance et donnera plus de fiabilité et moins de traitement par la suite.

Deux objectifs de cette thèse sont associés à la création d'une base de données au premier lieu et la réalisation d'un système de reconnaissance de lettres. La deuxième partie du travail au niveau expérimental exprime l'avantage du modèle bayésien comme une architecture probabiliste avec ses propres formules de classification.

Tableau V.4 : Matrice de confusion pour le modèle DBN

%	Alef	Ba	Ta	Tha	Jim	Ha	Kha	Dal	Dhal	Ra	Zain	Sin	Chin	Sad	Dhad	Ttah	Thah	Ain	Ghain	Fa	Qaf	Kaf	Lam	Mim	Noun	Waw	He	Ya
Alef	98,33	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1,67	-	-	-	-	-
Ba	-	70,21	18,8	10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ta	-	25	55	20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Tha	-	10,1	12,7	77,23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Jim	-	-	-	-	70	20	4,88	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ha	-	-	-	-	28,15	69,55	2,21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Kha	-	-	-	-	28,81	28,91	39,03	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Dal	-	-	-	-	-	-	-	52,20	47,80	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Dhal	-	-	-	-	-	-	-	39,88	60,12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ra	-	-	-	-	-	-	-	-	-	53,33	40,34	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6,33	-
Zain	-	-	-	-	-	-	-	-	-	44,60	55,40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sin	-	-	-	-	-	-	-	-	-	-	-	63,88	36,12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Chin	-	-	-	-	-	-	-	-	-	-	-	41,88	58,12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sad	-	-	-	-	-	-	-	-	-	-	-	-	-	52,14	29,35	10,01	08,50	-	-	-	-	-	-	-	-	-	-	-
Dhad	-	-	-	-	-	-	-	-	-	-	-	-	-	19,53	49,22	10,58	20,67	-	-	-	-	-	-	-	-	-	-	-
Ttah	-	-	-	-	-	-	-	-	-	-	-	-	-	14,16	3,90	66,52	15,42	-	-	-	-	-	-	-	-	-	-	-
Thah	-	-	-	-	-	-	-	-	-	-	-	-	-	6,02	10,33	28,11	55,54	-	-	-	-	-	-	-	-	-	-	-
Ain	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	78,58	21,42	-	-	-	-	-	-	-	-	-
Ghain	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37,78	62,22	-	-	-	-	-	-	-	-	-
Fa	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	58,40	41,60	-	-	-	-	-	-	-
Qaf	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	31,68	68,32	-	-	-	-	-	-	-
Kaf	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	95,11	-	-	-	-	-	-
Lam	11,14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	88,86	-	-	-	-	-
Mim	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	78,22	-	-	-	-
Noun	-	8,54	12,13	23,82	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	55,51	-	-	-
Waw	-	-	-	-	-	-	-	-	-	11,02	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	88,98	-	-
He	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	39,17	-	2,01	58,82	-
Ya	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	91,11

Tableau V.5 : Matrice de confusion pour le modèle DBN-NNGA

%	Alef	Ba	Ta	Tha	Jim	Ha	Kha	Dal	Dhal	Ra	Zain	Sin	Chin	Sad	Dhad	Ttah	Thah	Ain	Ghain	Fa	Qaf	Kaf	Lam	Mim	Noun	Waw	He	Ya
Alef	99,77	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ba	-	80.88	19.12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ta	-	-	72.54	27.46	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Tha	-	0.01	11.41	88.58	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Jim	-	-	-	-	71.41	18.42	10.17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ha	-	-	-	-	15.55	73.33	11.12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Kha	-	-	-	-	28,34	28.22	43.44	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Dal	-	-	-	-	-	-	-	58.55	41.12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Dhal	-	-	-	-	-	-	-	33.01	66.99	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ra	-	-	-	-	-	-	-	-	-	70.88	29.12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Zain	-	-	-	-	-	-	-	-	-	41.72	58.28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sin	-	-	-	-	-	-	-	-	-	-	-	78.99	21.01	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Chin	-	-	-	-	-	-	-	-	-	-	-	32.35	67.65	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sad	-	-	-	-	-	-	-	-	-	-	-	-	-	66.33	24.22	8.01	1.44	-	-	-	-	-	-	-	-	-	-	-
Dhad	-	-	-	-	-	-	-	-	-	-	-	-	-	16.21	59.87	11.11	12.81	-	-	-	-	-	-	-	-	-	-	-
Ttah	-	-	-	-	-	-	-	-	-	-	-	-	-	9.22	2.11	73.77	14.90	-	-	-	-	-	-	-	-	-	-	-
Thah	-	-	-	-	-	-	-	-	-	-	-	-	-	6.96	9.46	27.04	56.54	-	-	-	-	-	-	-	-	-	-	-
Ain	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	82.12	17.88	-	-	-	-	-	-	-	-	-
Ghain	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	28.98	71.02	-	-	-	-	-	-	-	-	-
Fa	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	61.00	39.00	-	-	-	-	-	-	-
Qaf	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	30.95	69.05	-	-	-	-	-	-	-
Kaf	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	97.21	-	-	-	-	-	-
Lam	9.11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	89.14	-	-	-	-	-
Mim	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	81.83	-	-	-	-
Noun	-	-	36,44	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	63.56	-	-	-
Waw	-	-	-	-	-	-	-	-	-	8.02	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	90.00	-	-
He	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	33,68	-	-	66.32	-
Ya	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	98.11

Tableau V.6 : Matrice de confusion pour l'amélioration par ARD Dynamique DBN-SPPD

%	Alef	Ba	Ta	Tha	Jim	Ha	Kha	Dal	Dhal	Ra	Zain	Sin	Chin	Sad	Dhad	Ttah	Thah	Ain	Ghain	Fa	Qaf	Kaf	Lam	Mim	Noun	Waw	He	Ya
Alef	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ba	-	98.22	1.78	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ta	-	-	95.55	4.45	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Tha	-	-	6.67	93.33	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Jim	-	-	-	-	89.99	9.22	0.79	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ha	-	-	-	-	4	91.99	4.01	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Kha	-	-	-	-	-	7.46	92.54	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Dal	-	-	-	-	-	-	-	95.96	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Dhal	-	-	-	-	-	-	-	4.04	98.02	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ra	-	-	-	-	-	-	-	-	-	91.91	8.09	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Zain	-	-	-	-	-	-	-	-	-	2.55	94.46	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sin	-	-	-	-	-	-	-	-	-	-	-	99.12	0.88	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Chin	-	-	-	-	-	-	-	-	-	-	-	2.28	97.72	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sad	-	-	-	-	-	-	-	-	-	-	-	-	-	92.52	5.48	2.00	-	-	-	-	-	-	-	-	-	-	-	-
Dhad	-	-	-	-	-	-	-	-	-	-	-	-	-	4.05	95.95	-	-	-	-	-	-	-	-	-	-	-	-	-
Ttah	-	-	-	-	-	-	-	-	-	-	-	-	-	5.08	-	89.92	5.00	-	-	-	-	-	-	-	-	-	-	-
Thah	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6.94	93.06	-	-	-	-	-	-	-	-	-	-	-
Ain	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	97.47	2.53	-	-	-	-	-	-	-	-	-
Ghain	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5.92	94.08	-	-	-	-	-	-	-	-	-
Fa	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	88.11	11.89	-	-	-	-	-	-	-
Qaf	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	13.98	86.02	-	-	-	-	-	-	-
Kaf	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-
Lam	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	95.00	-	-	-	-	-
Mim	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	98.08	-	-	-	-
Noun	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	88.98	-	-	-
Waw	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-
He	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	96.02	-
Ya	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100

CONCLUSION GENERALE & PERSPECTIVES

Pour faciliter davantage l'utilisation des ordinateurs, de nouvelles approches ont été proposées et créées. Elles n'utilisent plus le clavier ni la souris mais un stylet relié à une tablette électronique.

Etudier une problématique de reconnaissance du manuscrit arabe est un défi particulier surtout via la technologie en-ligne sachant que peu de travaux commercialisés dans ce sens ont vu le jour.

Les lettres arabes ont des spécificités liées à ce manuscrit qui est daté de plusieurs siècles et par conséquent son traitement n'était pas une tâche facile.

La recherche effectuée nous a conduit à découvrir beaucoup de problèmes dans le domaine de la reconnaissance d'écriture dont on a essayé de régler, un parmi plusieurs. C'est-à-dire que la construction de notre propre base de données NOUN et le traitement des caractères isolés sans passer à la reconnaissance des mots ou bien des phrases, ce n'est qu'un démarrage vers d'autres complexités.

Dans la recherche de meilleurs algorithmes capables de résoudre une classe de données de problèmes, une tendance actuelle veut que l'on bâtit des méthodes complexes incorporant des connaissances et des techniques issues de plusieurs horizons.

Le choix de l'approche de classification à développer est un autre défi et se justifie par l'utilisation des modèles graphiques de connaissances, tels que les réseaux bayésiens dynamiques, qui nous a permis de traiter efficacement l'ensemble des caractères isolés en gardant leurs informations spatiales et l'ordre d'écriture de chaque caractère du manuscrit arabe.

Pour améliorer notre modèle probabiliste tout en produisant des solutions performantes, une approche neuro-génétique hybride est souvent le meilleur choix à base de test.

Dans notre travail, on a été confronté d'une part au traitement automatique de l'écriture arabe en ligne, nécessitant des connaissances appropriées en écriture et en traitement numérique en ligne et d'autre part au peu d'articles traitant l'écriture arabe en ligne.

Comme travail futur, le passage vers le deuxième stade en traitant les mots et les phrases est vraiment intéressant surtout avec un enrichissement assuré de notre nouvelle base d'écriture arabe en ligne.

Finalement, reconnaître de l'écriture manuscrite consiste à associer une représentation symbolique à une séquence de symboles graphiques. Le but est de pouvoir utiliser cette représentation dans une application informatique. Du point de vue informatique, les difficultés de la reconnaissance d'écriture sont liées bien évidemment à la variabilité des écritures, mais aussi au fait qu'on ne sait pas encore très bien modéliser l'énorme masse de connaissances et d'informations utiles à la reconnaissance. La modélisation bayésienne, avec des améliorations via le temps et via une hybridation neuro-génétique, a montré une efficacité remarquable et approuvée par l'expérience. Notre système pourra mieux s'adapter en jouant sur d'autres paramètres du réseau qui nous offre la motivation à attaquer un autre défi dans ce domaine.

BIBLIOGRAPHIE

- [1] N. Mezghani, A. Mitiche, et M. Cheriet, “A new representation of shape and its use for high performance in online Arabic character recognition by an associative memory,” *International Journal on Document Analysis and Recognition*, vol. 7, n^o. 4, p. 201–210, 2005.
- [2] F. Menasri, “Contributions à la reconnaissance de l’écriture arabe manuscrite,” in *thèse de l’Université Paris Descartes*, 2008.
- [3] K. S. Nathan, H. S. Beigi, J. Subrahmonia, G. J. Clary, et H. Maruyama, “Real-time on-line unconstrained handwriting recognition using statistical methods,” in *International Conference on Acoustics, Speech, and Signal Processing*, p. 2619–2622, 1995.
- [4] H. Mouchère, “Étude des mécanismes d’adaptation et de rejet pour l’optimisation de classifieurs: Application à la reconnaissance de l’écriture manuscrite en-ligne,” in *thèse de l’Institut National des Sciences Appliquées de Rennes*, 2007.
- [5] M. Kleper, “Advanced display technologies: An Investigation of the emerging and developing technologies related to the generation beyond print-on-paper,” *A research monography of the printing industry center at RIT*, 2003.
- [6] B. K. Sin et J. H. Kim, “Ligature modeling for online cursive script recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, n^o. 6, p. 623–633, 2002.
- [7] K. H. Kim, T. J. Seong, et J. I. Doh, “On-line recognition of cursive Hangul characters by modeling extended graphemes,” in *Proceedings of the 4th International Conference on Document Analysis and Recognition*, vol. 2, p. 604–607, 1997.
- [8] R. Tlemsani et A. Benyettou, “On Line Isolated Characters Recognition Using Dynamic Bayesian Networks,” *International Arab Journal of Information Technology*, vol. 08, n^o. 04, Oct. 2011.
- [9] S. Quiniou, “Intégration de connaissances linguistiques pour la reconnaissance de textes manuscrits en-ligne,” in *thèse de L’INSA DE RENNES*, 2007.
- [10] P. Natajan, Z. Lu, R. Schwartz, I. Bazzi, et J. Makhoul, *Multilingual machine printed OCR*. World Scientific Publishing Co., Inc. River Edge, NJ, USA, 2002.

- [11] É. Poisson, “Architecture et apprentissage d’un système hybride neuro-markovien pour la reconnaissance de l’écriture manuscrite en-ligne,” in *thèse de l’Université de Nantes*, 2005.
- [12] A. Delaye, S. Macé, et E. Anquetil, “Hybrid statistical-structural on-line Chinese character recognition with fuzzy inference system,” in *19th International Conference on Pattern Recognition. ICPR 2008.*, p. 1–4, 2008.
- [13] R. Saabni et J. El-Sana, “Hierarchical On-line Arabic Handwriting Recognition,” in *10th International Conference on Document Analysis and Recognition*, p. 867–871, 2009.
- [14] A. Elbaati, M. Kherallah, A. M. Alimi, A. Ennaji, et T. Sfax, “De l’Hors-Ligne Vers un Système de Reconnaissance En-Ligne: Application à la Modélisation de l’Écriture Arabe Manuscrite Ancienne,” in *Semaine du document numérique, SDN*, 2006.
- [15] N. Mezghani, A. Mitiche, et M. Cheriet, “Estimation de densité de probabilité par maximum d’entropie et reconnaissance bayésienne de caractères Arabes en-ligne,” in *15e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle, Tours*, 2006.
- [16] N. Mezghani, A. Mitiche, et M. Cheriet, “Reconnaissance en-ligne de caractères arabes manuscrits par un réseau de kohonen,” in *International Symposium on Visual Computing Las Vegas*, p. 493–500, 2008.
- [17] F. Biadsy, J. El-Sana, et N. Habash, “Online arabic handwriting recognition using hidden markov models,” in *IWFHR 2006 France*, 2006.
- [18] M. Kherallah, F. Bouri, et A. M. Alimi, “Toward an On-Line Handwriting Recognition System Based on Visual Coding and Genetic Algorithm,” *Adaptive and Natural Computing Algorithms Journal*, vol. 12, n°. 3, p. 502–505, 2005.
- [19] M. Kherallah, F. Bouri, et A. M. Alimi, “On-line Arabic handwriting recognition system based on visual encoding and genetic algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 22, n°. 1, p. 153–170, 2009.
- [20] E. H. Ratzlaff, “Methods, report and survey for the comparison of diverse isolated character recognition results on the unipen database,” *Document Analysis and Recognition journal*, vol. 1, n°. 2, p. 623, 2003.
- [21] M. Parizeau, A. Lemieux, et C. Gagné, “Character recognition experiments using UNIPEN data,” in *Proceedings. Sixth International Conference on Document Analysis and Recognition, Seattle*, p. 481–485, 2001.

- [22] H. El Abed, V. Märgner, M. Kherallah, et A. M. Alimi, "Icdar 2009 online arabic handwriting recognition competition," in *2009 10th International Conference on Document Analysis and Recognition*, p. 1388–1392, 2009.
- [23] M. Deviren et K. Daoudi, "Structural learning of dynamic Bayesian networks in speech recognition," in *Seventh European Conference on Speech Communication and Technology, Scandinavia*, 2001.
- [24] F. V. Jensen, *An introduction to Bayesian networks*. UCL press London, 1997.
- [25] T. Starner, J. Makhoul, R. Schwartz, et G. Chou, "On-line cursive handwriting recognition using speech recognition methods," in *Proceedings of IEEE ICASSP Australia*, p. 125-128, 1994.
- [26] N. Friedman, K. Murphy, et S. Russell, "Learning the structure of dynamic probabilistic networks," in *Fourteenth Conf. on Uncertainty in Artificial Intelligence (UAI), Madison*, p. 139–147, 1998.
- [27] K. P. Murphy, "Inference and learning in hybrid Bayesian networks," 1998.
- [28] I. J. Kim et J. H. Kim, "Statistical utilization of structural neighborhood information for oriental character recognition," presented at the Proceedings of the 4 th IAPR International Workshop on Document Analysis Systems, p. 303–312, 2000.
- [29] H. Y. Kim et J. H. Kim, "Hierarchical random graph representation of handwritten characters and its application to Hangul recognition," *Computer Journal of Pattern Recognition*, vol. 34, n°. 2, p. 187–202, 2001.
- [30] C. L. Liu, I. J. Kim, et J. H. Kim, "Model-based stroke extraction and matching for handwritten Chinese character recognition," *Computer Journal of Pattern Recognition*, vol. 34, n°. 12, p. 2339–2352, 2001.
- [31] J. F. Jodouin, *Les Réseaux de neurones: Principes et définitions*. Hermes, 1994.
- [32] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley, 1989.
- [33] A. Spalanzani et J. Caelen, "Algorithmes évolutionnaires pour l'étude de la robustesse des systèmes de reconnaissance de la parole," in *thèse de l'Université Joseph Fourier - Grenoble I*, 1999.
- [34] R. K. Belew, J. McInerney, et N. N. Schraudolph, "Evolving networks: Using the genetic algorithm with connectionist learning," *Artificial life II*, p. 511–547, 1992.

- [35] R. Tlemsani, N. Neggaz, et A. Benyettou, "Amélioration de l'Apprentissage des Réseaux Neuronaux par les Algorithmes Evolutionnaires: application à la classification phonétique," in *3ème Conférence internationale: Sciences Electroniques, Technologies de l'Information et des Télécommunications SETIT2005 TUNIS*, 2005.
- [36] A. Ourdighi et A. Benyettou, "L'Intégration des Algorithmes Génétiques dans l'Apprentissage des Réseaux de Neurones à Délais Temporels Adaptatifs," in *4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications SETIT 2007 TUNIS*, 2007.
- [37] M. Crucianu, "Algorithmes d'évolution pour les réseaux de neurones," *Rapport de recherche*, vol. 187, 1997.

GLOSSAIRE

Nous donnons dans ce glossaire les définitions correspondant à certains sigles, acronymes et termes importants utilisés dans notre manuscrit.

ADAB :	base de données arabe (Arabic DAtaBase)
AG :	Les Algorithmes Génétiques
ARD :	Algorithme de Recherche Dynamique
DBN :	Dynamic Bayesian Network
HMM :	les modèles de Markov cachés
ISR :	Inter Stroke Relationships
IRONOFF:	base de données de LORIA (IReste ON/OFF database)
MSE:	Mean Square Error
NNGA :	Neural Network mixed with Genetic Algorithm
PMC :	Le Perceptron Multi-Couches
SPPS :	Segmentation par Plage de Points Statique
SPPD :	Segmentation par Plage de Points Dynamique
UNIPEN :	base de données latine
WSR :	Within Stroke Relationships

Sous-corpus : un nombre d'échantillon choisi pour notre expérimentation

Stroke : Segment élémentaire

Stylet : Stylo Électronique