

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Mémoire de Magister

Présenté à l'Université de Guelma
Faculté des sciences et sciences de l'ingénierie

Département de : **Informatique**
Ecole Doctorale Nationale Science et Technologie de l'Information et de la Communication
Spécialité : **Informatique**
Option : **SIC**

Présenté par : **Mr. Brahim FAROU**

Systeme hybride utilisant les réseaux de neurones et les modèles de Markov cachés pour la reconnaissance de mots manuscrits Arabe

JURY

Président	M.C	Khadir Mohamed Tarek	Université d'Annaba
Rapporteur	Pr	Seridi Hamid	Université de Guelma
Examineur	M.C	Fareh Nadir	Université d'Annaba
Examineur	Dr	Lafifi Yacine	Université de Guelma

2009

REMERCIEMENTS

En premier lieu, je tiens à exprimer ma profonde gratitude à mon encadreur Monsieur Hamid SERIDI de m'avoir proposé ce sujet et diriger constamment de près mon travail. Je le remercie pour sa disponibilité, ses encouragements, son soutien moral, sa sympathie et son amitié sincère dont il a fait preuve le long de ce travail. J'apprécie la confiance qu'il m'a témoignée et les conseils avisés qu'il m'a prodigués. Qu'il trouve ici mes sincères reconnaissances.

Je tiens aussi à remercier Monsieur Amine BENZENACHE pour son aide, ses inestimables conseils, ses explications claires et ses orientations précises. Qu'il trouve ici mes sincères reconnaissances.

Je suis très reconnaissant à Mr. Fareh Nadir d'avoir accepté de faire partie du jury, je le remercie pour l'intérêt qu'il a porté à mon travail.

Merci à Mr. Khadir Mohamed Tarek de m'avoir fait l'honneur de présider le jury, je lui suis très reconnaissant pour l'intérêt qu'il a porté à mon travail.

Un grand merci à Mr. Lafifi Yacine d'avoir accepté de faire partie du jury, je le remercie pour l'intérêt qu'il a porté à mon travail.

Je remercie également tous les membres de l'équipe SERIDI pour leur soutien moral et leur encouragement pour l'accomplissement de ce travail.

Je ne serais sans doute pas arrivé jusqu'ici sans avoir suivi les cours d'enseignants exceptionnels qui ont participé à ma formation, je pense en particulier aux enseignants de l'École Doctorale de Guelma.

Je remercie également le chef de département d'informatique, Monsieur Ali SERIDI, ainsi que tout le personnel du département d'informatique pour leur soutien moral et leur commodité.

Enfin, je remercie mes collègues : Hallaci Samir, Toualbia Ilyes, Halimi Khaled, Khebizi Ali, Messolhi Said, Gouadjlia Toufik, Zedadra Ouarda, Legrini Samira pour l'amitié et le soutien qu'ils m'ont apportés durant la période d'étude et de préparation de ce travail.

Dédicace

Je dédie ce modeste travail

À mes très chers parents que j'aime beaucoup

À mes frères Dhia Eldinne, Housseme et Fakaria

À ma chère et tendre épouse Halima

À tous mes amis et camarades

À toute ma famille

À Mr. Seridi Hamid

À Mr. Benxenache Amine

À tous ceux qui m'ont aidé de près ou de loin

Brahim

Résumé

La reconnaissance de l'écriture manuscrite est le vieux rêve de tous ceux qui ont eu besoin d'insérer des données dans un ordinateur. Aujourd'hui, il existe plusieurs domaines dans lesquels la reconnaissance de l'écriture manuscrite est attendue avec impatience, par exemple dans le tri automatique du courrier, le traitement automatique de dossiers administratifs, des formulaires d'enquêtes, ou encore l'enregistrement des chèques bancaires. Dans ce cadre, la reconnaissance hors ligne des textes manuscrits cursifs écrits en arabe, reste cependant un problème ouvert. Ceci est dû aux difficultés inévitables auxquelles sont confrontés les auteurs des systèmes de reconnaissance automatique, tels que la normalisation de l'écriture, la segmentation des mots en éléments de base ainsi que la modélisation de ces éléments.

Pour remédier à ce problème, plusieurs classifieurs ont été testés. Les Modèles de Markov Cachés ont eu beaucoup de succès avec le domaine de la reconnaissance de l'écriture manuscrite, ce succès vient de leur capacité d'intégration du contexte, d'absorption du bruit et le pouvoir de classifier les mots sans avoir le besoin de les segmenter en caractères. Cependant, les informations locales sont quelque peu négligées et ils souffrent d'un faible pouvoir discriminatif. Les classifieurs à base des Réseaux de Neurones ont prouvé leur puissance en ce qui concerne la reconnaissance de caractère ou chiffre isolé grâce à leur pouvoir de généralisation à partir d'un ensemble réduit d'échantillons d'apprentissage, mais la faiblesse des méthodes de segmentation influence sur la performance de ce classifieur.

Nous proposons une manière de coopérer des MMC et des réseaux neuronaux dans une architecture probabiliste en tirant avantage des deux outils : la génération d'une liste des N meilleures hypothèses de mots ainsi que leurs segmentations en caractères par un classifieur MMC et les propriétés de modélisation des réseaux neuronaux appliquées aux caractères. Le classifieur RN utilise la segmentation du classifieur MMC afin de retourner à l'image du mot et d'extraire les caractéristiques convenables à la reconnaissance de caractères isolés. Le classifieur RN réévalue chacune des N meilleures hypothèses des mots et les scores générés sont combinés avec ceux du classifieur MMC. Finalement, la liste des N meilleures hypothèses est réordonnée selon les nouveaux scores faisant ainsi ressortir la meilleure hypothèse.

Concevoir un système pour la reconnaissance de texte manuscrit hors ligne des 48 wilayas algériennes est l'objectif des travaux de recherche de ce mémoire. Avec un taux de réussite de 91,77 %, le système proposé a montré une bonne performance par rapport aux systèmes référencés dans l'état de l'art.

Mots clés : *Reconnaissance de l'écriture manuscrite, modèles de Markov cachés, réseaux de neurones, Algorithme de Baum Welch, Algorithme de Viterbi*

Abstract

Handwriting recognition is the old dream of all those who needed to insert data into a computer. Today, there are several areas where the handwriting recognition is impatiently awaited, for example in the automatic sorting of mail, the automatic processing of administrative records, forms, surveys, or registering checks. In this context, the recognition of offline cursive manuscripts written in Arabic, still, remains an open problem. This is unavoidable due to the difficulties facing the authors of the automatic recognition systems, such as standardization of the writing, the segmentation of words into basic elements and the modeling of these elements.

To overcome this problem several classifiers have been tested. The Hidden Markov Models have had a great success with the field of handwriting recognition; this success comes from their ability to integrate the context, noise absorption and the power to classify words without the need to segment it into characters. However, the local information are somewhat ignored and suffer from a low discriminating power. The classifiers based neural networks have proved their power with regard to the recognition of isolated character or digit through their power of generalization from a reduced set of data learning, but the weakness of the segmentation methods influence on the performance of this classifier.

We propose a way to cooperate HMM and neural networks in a probabilistic architecture taking advantage of two tools: generating a list of N best word hypotheses and their segmentations by a character HMM classifier and modeling properties of neural networks applied to characters. The ANN classifier uses the HMM classifier segmentation to return to the image of the word and extract the suitable characteristics to the recognition of isolated characters. ANN classifier reassesses each of N best word hypotheses and generated scores are combined with those of HMM classifier. Finally, the list of N best hypotheses is rearranged according to the new scores and highlights the best hypothesis.

Designing a system for recognition of offline handwritten text of the 48 Algerian states is the research purpose of this thesis. With a success rate of 91.77%, the proposed system showed good performance compared to the references systems in the state of the art.

Keywords: Handwriting recognition, hidden Markov models, neural networks, Baum Welch algorithm, Viterbi algorithm

ملخص

التعرف على الكتابة المخطوطة باليد حلم قديم لجميع أولئك الذين هم في حاجة لإدخال البيانات في جهاز الكمبيوتر. اليوم ، هناك العديد من المجالات التي ينتظر فيها التعرف على الكتابة المخطوطة باليد بفارغ الصبر ، على سبيل المثال في الفرز الآلي للبريد ، والتجهيز التلقائي للسجلات الإدارية ، وعمليات المسح ، أو تسجيل الشبكات. في هذا السياق التعرف على المخطوطات المكتوبة باللغة العربية لا يزال مشكلة مفتوحة. ويرجع هذا الأمر بسبب الصعوبات التي يواجهها الباحثون في النظم التلقائية للتعرف ، مثل توحيد الكتابة ، وتقسيم الكلمات إلى عناصر أساسية أو وضع نماذج لهذه الكلمات.

للتغلب على هذه المشكلة تم اختبار العديد من المصنفات الآلية . إن نماذج ماركوف المخفية كان لها نجاحا كبيرا مع مجال التعرف على الكتابة المخطوطة باليد ، وهذا النجاح راجع إلى قدرتها على الاندماج وامتصاص التشويشات والقدرة على تصنيف الكلمات دون الحاجة لتجزئتها الى حروف. ومع ذلك ، فإن المعلومات المحلية مهمة وتعاني من انخفاض في القدرة على التمييز. إن المصنفات القائمة على الشبكات العصبية أثبتت قوتها في مجال التعرف على الحروف و الأرقام المعزولة بواسطة القدرة على التعميم من خلال مجموعة قليلة من نماذج التعلم ، ولكن ضعف أساليب التجزئة تأثر على أداء هذا المصنف.

نقترح طريقة لتعاون نماذج ماركوف المخفية والشبكات العصبية في وسط احتمالي بنية الاستفادة منهما : توليد قائمة أفضل N فرضية للكلمة وتقسيماتها الناتجة عن نماذج ماركوف المخفية ، وتطبيق المزايا التمييزية للشبكات العصبية على الحروف. إن الشبكات العصبية تستخدم تجزئة نماذج ماركوف المخفية للعودة الى الصورة الاصلية واستخلاص الخصائص المناسبة لتعرف على الحروف المعزولة. تعيد الشبكات العصبية تقييم كل فرضية من قائمة أفضل N فرضية لكل كلمة ، ويتم توحيد النتيجة مع نتائج مصنف ماركوف. وأخيرا ، نقوم بإعادة ترتيبها وفقا للحسابات الجديدة ، فضلا عن تسليط الضوء على أفضل فرضية.

تصميم نظام التعرف على الكتابة المخطوطة باليد لـ 48 ولاية في الجزائر هو الغرض من البحث في هذه الرسالة. مع نسبة نجاح تقارب الـ 91.77 % ، يكون النظام المقترح قد أظهر أداء جيد مقارنة مع النظمة المدروسة في هذه الرسالة.

الكلمات الرئيسية : التعرف على الكتابة المخطوطة باليد، نماذج ماركوف المخفية ، الشبكات العصبية ، خوارزمية Baum Welch ، خوارزمية Viterbi

TABLE DES MATIERES

Page

Liste des Figures

Liste des Tableaux

Introduction Générale _____ 1

Chapitre I : Reconnaissance Automatique de L'écriture

1. Introduction _____ 5

2. Reconnaissance en ligne ou hors-ligne _____ 5

2.1. La reconnaissance en ligne (on line) _____ 6

2.2. La reconnaissance hors-ligne (off line) _____ 6

3. Méthodes de reconnaissance _____ 6

3.1. L'approche globale _____ 7

3.2. L'approche analytique _____ 7

3.3. L'approche pseudo-analytique _____ 8

3.4. L'approche perceptuelle _____ 8

4. Organisation générale d'un système de reconnaissance _____ 8

4.1. Acquisition _____ 9

4.2. Prétraitements _____ 10

4.2.1. Binarisation _____ 10

4.2.2. Filtrage _____ 11

4.2.3. Localisation de l'information manuscrite _____ 11

4.2.4. Redressement de l'écriture _____ 13

4.2.5. Lissage _____ 14

4.2.6. Squelettisation _____ 15

4.2.7. Normalisation de la taille des images _____ 15

4.2.8. Extraction des contours	16
4.2.9. Extraction des composantes connexes	18
4.3. Segmentation	18
4.3.1. Segmentation implicite	18
4.3.2. Segmentation explicite	19
4.4. Extraction des primitives (caractéristiques)	21
4.4.1. Caractéristiques globales	23
4.4.2. Le développement en série	23
4.4.3. Concavités, convexités et occlusions	24
4.4.4. Allongement horizontaux et verticaux	24
4.4.5. Caractéristiques locales	24
4.4.6. Intersection avec des droites	24
4.4.7. Caractéristiques géométriques	24
4.4.8. Caractéristiques de densités et de zoning	25
4.5. Classification	25
4.5.1. Les approches de classification	26
4.5.1.a. Les approches statistiques	26
4.5.1.b. Les approches structurelles	27
4.5.1.c. Les approches hybrides	28
4.6. Post traitement	28
4.7. La prise de décision	29
4.7.1. Evaluation des performances d'un classifieur	29
4.7.2. Taux de reconnaissance et taux d'erreurs	29
5. Problèmes liés à l'OCR	30
6. Les difficultés de reconnaissance de l'écriture manuscrite	31
6.1. La disposition spatiale du manuscrit	31
6.2. Nombre de scripteurs	32
6.3. Taille du vocabulaire	32
6.4. Contraintes morphologiques	33
7. Conclusion	33

Chapitre II : Travaux Existants

1. Introduction	34
2. Système de Miled et al. 1998	34
3. Système d'Essoukhri Ben Amara 1999	36
4. Système d'Amin 2000	37
5. Système de Benouareth 2000	37
6. Système d'Azizi et al. 2002	39
7. Système de Al Ohali 2002	39
8. Système de Kanoun et al., 2002	41
9. Système de Khorsheed 2003	42
10. Système de Pechwitz et al. 2003	43
11. Système de Al-Ma'adeed et al. 2004	43
12. Système de Kammoun et al. 2004	44
13. Système de Farah et al. 2005	45
14. Système de Sari et al. 2005	45
15. Système de Benzenache 2007	46
16. Système de Zaghdoudi 2008	47
17. Conclusion	48

Chapitre III : Les Réseaux De Neurones

1. Introduction	49
2. Définition	49
3. Historique	49
4. Le neurone biologique	51
5. Les neurones formels	52
6. Architecture des réseaux de neurones	54

6.1. Les réseaux à couches	55
6.2. Les réseaux entièrement connectés	56
7. L'apprentissage	57
7.1. Apprentissage supervisé	57
7.2. Apprentissage non supervisé	58
8. Les règles d'apprentissage	58
8.1. La règle de HEBB	58
8.2. Règle de Windrow-Hoff	59
9. Les modèles des réseaux de neurones	59
9.1. Adaline de Windrow-Hoff	59
9.2. Le perceptron de ROSENBLATT	60
9.3. Le perceptron multicouche	61
9.4. Le modèle de HOPFIELD	62
9.5. Le modèle de KOHONEN	63
10. Propriétés des réseaux de neurones	64
11. Conclusion	64

Chapitre IV : Les Modèles De Markov Cachés

1. Introduction	65
2. Définitions préliminaires	65
3. Le modèle de Markov à temps discret	66
3.1. Propriétés de Markov du premier ordre	67
3.2. Paramètres d'un modèle de Markov discret	67
4. Le modèle de Markov caché discret (DHMM)	68
4.1. Définitions	68
4.2. Paramètres d'un modèle de Markov caché discret	69
5. Topologie d'HMMs	70

5.1. Modèle ergodique	70
5.2. Modèle gauche-droite	70
6. Les trois problèmes associés aux modèles de Markov Cachés	71
6.1. Solution du problème d'évaluation	71
6.1.1. Calcul de la probabilité d'émission d'une séquence d'observations	72
6.1.2. Procédure récursive directe (l'algorithme Forward)	73
6.1.3. Procédure récursive inverse (algorithme Backward)	74
6.2. Solution du problème d'apprentissage (estimation des paramètres)	74
6.2.1. Maximum de vraisemblance	75
6.2.2. Ré-estimation du modèle	75
6.2.3. Autres critères d'apprentissage (MAP et MMI)	77
6.3. Solution du problème de décodage (reconnaissance)	78
7. Conclusion	80

Chapitre V : Système Hybride RN/MMC Pour La Reconnaissance De Mots Manuscrits arabe

1. Introduction	81
2. La langue Arabe	82
3. Architecture du Système proposé	86
3.1. Sous Système basé sur les Modèles de Markov cachés	87
3.1.1. Acquisition	87
3.1.2. Prétraitement	87
3.1.2.a. Binarisation	87
3.1.2.b. Lissage	88
3.1.2.c. Normalisation	88
3.1.2.d. Cadrage	88
3.1.2.e. Squelettisation	89
3.1.2.f. Correction de la déformation des caractères	91
3.1.2.g. Estimation de la ligne de base	93
3.1.3. Segmentation	93

3.1.4. Extraction des primitives	94
3.1.5. Quantification vectorielle	95
3.1.6. Classifieur MMC	97
3.2. Sous Système basé sur les réseaux de neurones	98
3.2.1. Segmentation	98
3.2.2. Extraction des primitives	98
3.2.3. Classifieur RN	100
3.3. Combinaison des résultats des classifieurs MMC/RN	100
4. Tests et Résultats	101
5. Conclusion	105
Conclusion Générale	106

Bibliographie

LISTE DES FIGURES

<u>Figure</u>	<u>Titre</u>	<u>Page</u>
Figure I.1 :	Système de Reconnaissance des caractères _____	9
Figure I.2 :	Histogramme des niveaux de gris _____	11
Figure I.3 :	Histogramme de la segmentation lignes _____	12
Figure I.4 :	Correction de l'inclinaison (20°) _____	14
Figure I.5 :	Exemple du mot « قالمة » avant et après l'extraction du contour _____	17
Figure I.6 :	Exemple du mot « قالمة » avant et après l'extraction du contour extérieur _____	17
Figure III.1 :	Le neurone biologique _____	51
Figure III.2 :	Le neurone formel _____	52
Figure III.3 :	Fonction de Heaviside _____	53
Figure III.4 :	Fonction Signe _____	53
Figure III.5 :	Fonction Linéaire multi seuils _____	53
Figure III.6 :	Fonction à Seuils _____	53
Figure III.7 :	Fonctions sigmoïdes _____	54
Figure III.8 :	Réseau à couches _____	56
Figure III.9 :	Réseaux entièrement connectés _____	56
Figure III.10 :	Le neurone d'Adaline _____	60
Figure III.11 :	Représentation du Perceptron _____	61
Figure III.12 :	Perceptron Multicouches _____	62
Figure III.13 :	Le réseau de HOPFIELD _____	62
Figure III.14 :	Carte de Kohonen avec deux entrées _____	63
Figure IV. 1 :	Diagramme d'états d'un processus de Markov discret _____	68

Figure IV.2 :	(1) Modèle ergodique à 4 états, (2) Modèle gauche-droite (a) le modèle séquentiel, (b) le modèle parallèle_____	70
Figure V. 1 :	Les classes de reconnaissance_____	82
	Exemples d'écritures arabes reflétant ses caractéristiques morphologiques. (a) Ecriture manuscrite,	
Figure V. 2 :	(b) Exemple d'élongations horizontales, (c)Ecriture voyellée, (d) Variations dans des bandes, (e) Différentes combinaisons verticales possibles de caractères_____	85
Figure V. 3 :	Architecture du Système_____	86
Figure V. 4 :	L'opération de Binarisation et ses effets sur l'image_____	87
Figure V. 5 :	Opération de Lissage_____	88
Figure V. 6 :	Opération de Cadrage_____	89
Figure V. 7 :	Algorithme de squelettisation de Zhang-Suen appliqué sur des mots manuscrits arabes_____	89
Figure V. 8 :	Résultat de la squelettisation des mots manuscrits arabes en utilisant l'algorithme de Zhang-Suen modifié_____	91
Figure V. 9 :	Déformation des mots après la squelettisation (création de concave et convexe non désirées)_____	91
Figure V. 10 :	Résultats de l'opération de correction de la déformation sur des mots manuscrits arabes_____	92
Figure V. 11 :	Détection de la ligne de base en utilisant l'histogramme de projection horizontale_____	93

- Figure V. 12 :** Segmentation non uniforme utilisant
l'analyse de l'histogramme de projection verticale _____ **93**
- Figure V. 13 :** Modèle HMM de type droite-gauche du mot **فالمة** _____ **97**

LISTE DES TABLEAUX

Tableau I.1 :	Répartition des exemples dans la base de test cas général_____	30
Tableau V. 1 :	Les formes des caractères arabes dans des différentes positions_____	83
Tableau V. 2 :	Structure du vecteur de caractéristiques pour le sous-système basé sur le classifieur MMC_____	95
Tableau V. 3 :	Structure du vecteur de caractéristiques pour le sous-système basé sur le classifieur RN_____	99
Tableau V. 4 :	Quelques échantillons de la base des mots_____	101
Tableau V. 5 :	Quelques échantillons de la base des caractères_____	102
Tableau V. 6 :	Résultats de la reconnaissance_____	103
Tableau V. 7 :	Comparaison des résultats de notre système avec celui de Benzenache et Zaghdoudi_____	104

Introduction générale

L'écriture fût durant ce dernier siècle le mode de collecte, de stockage et de transmission de l'information le plus couramment utilisé. Elle demeure le mode de communication le plus privilégié par l'être humain. Utiliser à la fois à des fins personnelles et dans un cadre professionnel, elle ne se limite plus à l'échange entre les êtres humains, mais elle est aussi utilisée comme interface entre l'homme et la machine. Ainsi, la lecture automatique de l'écriture fût durant ces trente dernières années l'objet d'intenses recherches.

Toutefois, les premiers ordinateurs souffraient du manque de puissance et d'espace mémoire ce qui a influencé sur l'avancement des recherches dans le domaine. Mais avec l'explosion des technologies de l'information qui a eu lieu depuis le début des années 80, la quantité de recherche a énormément augmenté.

La reconnaissance automatique de l'écriture manuscrite est le fruit des apports de plusieurs disciplines. Poussé par un besoin croissant d'automatisation, le traitement de l'écrit a donné lieu à une activité de recherche importante. En effet, chaque jour plus d'un milliard de documents commerciaux et financiers sont traités par ordinateur. La grande majorité d'entre eux sont traités manuellement par des opérateurs humains qui doivent lire puis retranscrire le contenu de chaque document ce qui représente une tâche fastidieuse et très coûteuse en temps.

Il est donc naturel de chercher à automatiser ce processus en remplaçant l'opérateur humain par un système de reconnaissance capable d'effectuer le même travail. Les avancées récentes dans le domaine de l'analyse et de la reconnaissance des documents, ainsi que la baisse des coûts de la puissance de calcul permettent aujourd'hui le développement de systèmes efficaces qui tentent de minimiser les interventions humaines.

Cependant, la route est encore longue avant d'arriver à l'obtention d'une machine capable de simuler parfaitement le processus de la lecture chez l'être humain, tout particulièrement dans le cas de la reconnaissance hors ligne d'écriture manuscrite non contrainte. En effet, l'objectif ultime d'un système de reconnaissance automatique de l'écriture manuscrite est de réaliser un système qui soit capable de comprendre le sens d'un texte écrit par n'importe quelle personne.

Pour réaliser un système pareil, il doit être non seulement, capable de reconnaître n'importe quelle écriture, mais il faudrait également mieux exploiter les différents niveaux

d'interprétations depuis le niveau graphique jusqu'au niveau lexical et syntaxique, voire sémantique.

Parmi les nombreuses études présentées dans la littérature, la plupart portent sur la reconnaissance de sous-unités isolées d'écriture, tels que les caractères, les mots ou les chaînes de chiffres. Pour reconnaître ces sous-unités, deux types d'approches de reconnaissances sont utilisées à savoir les approches analytiques et les approches globales. Généralement, les approches globales sont dédiées à la reconnaissance des mots dans un vocabulaire limité où le nombre de mots à reconnaître constitue un lexique de taille restreinte, cependant lorsque le lexique est de taille étendue une phase de segmentation s'avère indispensable.

Vu que l'arabe est une langue parlée par les Arabes dans plus de 20 pays, elle a bénéficié de nombreuses recherches dans le domaine de la reconnaissance automatique, cependant, les chercheurs se sont retrouvés face à de nouveaux problèmes qui n'existaient pas lors des précédentes recherches dans la reconnaissance d'autres langues. En effet, l'arabe se distingue des autres langues par plusieurs caractéristiques propres à cette dernière. L'une des différenciations les plus importantes provient de sa semi-cursivité aussi bien dans sa forme imprimée que manuscrite, d'où la notion de sous-mot ou pseudo-mot. Nous retenons également la forte dépendance de la calligraphie du caractère de son contexte, la complexité et la multiplicité des graphies des lettres, la variabilité des liaisons intra-caractères ou des ligatures horizontales et verticales ainsi que la présence de chevauchements.

La nature calligraphique de l'arabe rend délicat le choix des procédures de prétraitement, fait de la segmentation de textes arabes un problème épineux et complique la sélection des primitives et leur mode d'extraction.

La reconnaissance de l'écriture arabe souffre, en dehors de sa nature, de plusieurs problèmes qui concernent essentiellement le manque d'outils de test et de validation, généralement présents dans les systèmes de reconnaissance. Ces problèmes sont, en fait, une conséquence immédiate du manque d'applications porteuses. Une sensibilisation des utilisateurs potentiels telle que les banques ou les postes est une action primordiale afin de faire franchir la reconnaissance de l'écriture arabe les frontières de la pure recherche à l'exploitation réelle.

Les progrès réalisés ces dernières années dans le domaine de la reconnaissance de l'écriture manuscrite arabe sont dus en grande partie à l'utilisation d'approches statistiques. Parmi celles-ci, deux techniques ont plus particulièrement été mises à contribution. Il s'agit d'une part des approches basées sur les modèles de Markov cachés et d'autre part des modélisations basées sur les modèles connexionnistes.

Les Modèles de Markov cachés ont eu beaucoup de succès avec le domaine de la reconnaissance de l'écriture manuscrite, ce succès vient de leur capacité d'intégration du contexte, d'absorption du bruit et le pouvoir de classifier les mots sans avoir le besoin de les segmenter en caractères. Cependant, les informations locales sont quelque peu négligées et ils souffrent d'un faible pouvoir discriminatif. Les classifieurs à base des Réseaux de neurones ont prouvé leur puissance en ce qui concerne la reconnaissance de caractère ou chiffre isolé grâce à leur pouvoir de généralisation à partir d'un ensemble réduit d'échantillon d'apprentissage, mais la faiblesse des méthodes de segmentation influence sur la performance de ce classifieur.

C'est dans cette voie que s'inscrit le cadre de notre travail et qui a comme objectif l'élaboration d'un système de reconnaissance de l'écriture manuscrite à vocabulaire limité, car l'application visée derrière cette architecture est la classification des noms des wilayas de l'Algérie. Pour cela, nous avons adopté une approche hybride qui fait interagir les deux types de classifieur mentionnés précédemment. Afin d'atteindre notre objectif, nous avons divisé ce mémoire en cinq chapitres :

Nous avons présenté dans le premier chapitre une vue générale du domaine de la reconnaissance automatique de l'écriture. Nous avons commencé dans un premier temps par décrire les types et les méthodes de reconnaissance de l'écriture, ensuite construire une organisation générale d'un système de reconnaissance et détailler chaque module, et nous avons terminé par les problèmes liés à l'ORC.

Le deuxième chapitre est un état de l'art regroupant plusieurs travaux antérieurs traitant la reconnaissance de l'écriture dans ses deux formes manuscrites et imprimées et en utilisant les différentes stratégies de reconnaissance (globale et analytique) appliquant divers types de classifieurs.

Le troisième chapitre est une étude générale sur les réseaux neuronaux, les algorithmes d'apprentissage ainsi que les fonctions de seuillage. Nous abordons aussi l'architecture et les

types des réseaux neuronaux, et nous terminons par quelques propriétés des réseaux de neurones.

Le quatrième chapitre contient des notions théoriques de base sur les modèles de Markov cachés en tant qu'outils statistiques et nous spécifions dans notre étude les modèles de Markov caché discret qui sont l'un des deux modules de notre système. Ensuite, une description des différentes topologies utilisées est illustrée. Nous citons les problèmes liés à ce type de classifieur et les solutions proposées pour les résoudre en mettant la lumière sur quelques critères d'apprentissage.

Le cinquième chapitre est réservé à la conception et l'implémentation du système proposé. Nous avons d'abord abordé la langue arabe et ses caractéristiques, nous avons ensuite proposé une architecture pour notre système et détaillé chaque module. Cette partie va décrire aussi les différents points d'interaction entre les deux classifieurs et détailler la conception de chacun d'eux sachant que notre système est une hybridation entre un perceptron multicouche et un modèle de Markov caché discret de type droite-gauche. Ce chapitre va contenir aussi les tests et les résultats enregistrés par notre système au cours de notre expérimentation.

Finalement, nous terminons ce mémoire par une conclusion générale et quelques perspectives dans le domaine.

Chapitre I

Reconnaissance Automatique de L'écriture

1. Introduction

Durant les vingt dernières années, la lecture optique des textes et la reconnaissance d'écriture manuscrite ont été l'objet de nombreux travaux de recherches suivis de nombreuses publications dans le monde entier. On a ainsi peu à peu dégagé une classification des différentes approches possibles selon les applications envisagées et mis en évidence les difficultés spécifiques qu'a chacune de ces approches depuis l'acquisition des données jusqu'à l'interprétation finale. Il existe deux principaux axes de recherche : la reconnaissance de l'écriture imprimée et de l'écriture manuscrite. Si dans le premier cas, on peut constater que les principales difficultés ont été surmontées, dans le cas contraire en ce qui concerne la reconnaissance du manuscrit ces difficultés existent toujours.

L'objectif de la reconnaissance de l'écriture manuscrite est de développer un système qui se rapproche le plus de l'être humain dans sa capacité de lire. Malgré que les études en cours soient nombreuses, il n'existe encore aucun système capable de traiter de façon fiable ce problème, et cela à cause de sa complexité et sa diversité et des difficultés de réalisation des systèmes universels de reconnaissance, donc il est apparu que la seule voie à court terme consistait à particulariser le problème traité.

Nous allons essayer, dans chapitre, de donner une vue générale sur la reconnaissance automatique de l'écriture. Nous commençons dans un premier temps par les types de reconnaissance de l'écriture et les méthodes de reconnaissance, ensuite nous donnons une organisation générale d'un système de reconnaissance et nous détaillons chaque module et nous terminons par les problèmes liés à l'ORC.

2. Reconnaissance en ligne ou hors-ligne

La reconnaissance de l'écriture manuscrite se devise en deux grandes catégories, selon le mécanisme d'acquisition de l'écriture, à savoir en ligne (on line) ou hors ligne (off line) dont nous allons donner la définition ainsi que quelques caractéristiques de chaque catégorie.

2.1. La reconnaissance en ligne (on line)

La reconnaissance en ligne ou on line est une reconnaissance dynamique et en temps réel, les symboles sont reconnus au fur et à mesure que l'utilisateur écrit dans un écran tactile ou dans une surface spécialisée (tablette graphique) à l'aide d'un stylo électronique, ce mode de reconnaissance est réservé à l'écriture manuscrite et ne présente pas beaucoup de handicap à cause des caractéristiques suivantes :

- Données mono dimensionnelles
- Écriture présentée par un ensemble de points ordonné dans le temps
- Système interactif où chaque erreur peut être corrigée immédiatement
- Reconnaissance de graphèmes presque évidente à cause du marquage du posé et levé de stylo
- Accès direct au tracé (l'information utile)

2.2. La reconnaissance hors ligne (off line)

Contrairement aux systèmes en ligne, les systèmes hors ligne disposent en entrée d'une image caractérisée par sa résolution et le nombre de niveaux de gris qui lui sont associés. Cette méthode est convenable dans le cas des documents contenant une écriture manuscrite ou imprimée. Aujourd'hui, la reconnaissance de textes imprimés offre des résultats tout à fait satisfaisants quand la numérisation est de bonne qualité. Il n'en est pas de même, pour la reconnaissance de l'écriture manuscrite où le problème est d'une tout autre complexité.

De plus, contrairement à la reconnaissance en ligne de l'écriture manuscrite où l'on accède directement au tracé, la reconnaissance hors ligne doit faire face à un premier problème qui consiste à traiter l'image afin d'en extraire le message écrit. Pour cela une suite d'opérations très délicates sur l'image sont nécessaires pour pallier ce problème.

Le second problème qui avantage les systèmes en ligne sur les systèmes hors ligne c'est l'absence d'information sur la dynamique du tracé ce qui rend la tâche de segmentation plus difficile et ambiguë.

3. Méthodes de reconnaissance

Les approches de reconnaissance de mots peuvent être divisée en quatre catégories principales que nous présentons dans cette section, la première globale et considère le mot comme une entité indivisible, la seconde est analytique et se base sur la segmentation du mot

en caractères, tandis que la troisième est une approche qui combine les deux premières. La quatrième est perceptuelle et inspirée des humains.

3.1. L'approche globale

L'approche globale a une vision générale du mot, elle reflète le processus humain de lecture mot par mot et elle est basée sur une description unique de l'image du mot et sur l'extraction des paramètres caractérisant l'allure globale du mot. Vue comme une entité indivisible disposant de beaucoup d'information elle absorbe plus facilement les variations au niveau de l'écriture, dans ce cadre il existe plusieurs stratégies basées sur l'allure des contours inférieurs et supérieurs, leurs comparaisons avec des formes d'un dictionnaire, d'autres stratégies sont basées sur des descriptions des mots sous forme d'un graphe reliant les points caractéristiques du squelette ou bien sur l'orientation des concavités. Cette approche est applicable sur des mots cursifs appartenant à un vocabulaire réduit, et l'apprentissage des modèles nécessite une grande quantité d'échantillons qui est souvent difficile à réunir.

3.2. L'approche analytique

Cette approche, appelée aussi basée segmentation, permet de s'affranchir des limites de la précédente, mais nécessite une interprétation locale basée sur un découpage (segmentation) du mot. La difficulté d'une telle approche est la segmentation, car pour reconnaître les lettres, il faut segmenter le tracé et pour segmenter le tracé, il faut y reconnaître des lettres [BEL 01].

Il s'ensuit qu'un processus de reconnaissance selon cette approche doit nécessairement se concevoir comme un processus de relaxation alternant les phases de segmentation et d'identification des segments.

Cette approche est la seule applicable dans le cas de grands vocabulaires et peut s'adapter facilement à un changement de vocabulaire. Elle permet théoriquement une discrimination plus fine des mots, car elle se base sur la reconnaissance des lettres qui la composent et il est possible de récupérer l'orthographe du mot reconnu. Son inconvénient principal demeure la nécessité de l'étape de segmentation avec les problèmes de sous-segmentation ou de sur segmentation que cela implique.

3.3. L'approche pseudo analytique

C'est une approche hybride combinant les caractéristiques des deux premières (globale et analytique). Elle opère sur le pseudo mot qui est nettement visible dans les scriptes tels que l'arabe.

3.4. L'approche perceptuelle

Cette approche est utilisée dans les systèmes orientés perception inspirés de la lecture humaine. Elle se base sur des modèles psychologiques de la lecture et de la perception humaine des mots.

4. Organisation générale d'un système de reconnaissance

L'objectif de la reconnaissance de l'écriture manuscrite est de développer un système qui se rapproche le plus de l'être humain dans sa capacité de lire. Cependant, cette reconnaissance de l'écriture consiste à extraire d'une forme inconnue (mot, lettres, chiffres) une description plus simple et à établir sur celle-ci une décision. Cette décision est effectuée généralement en mesurant la ressemblance d'une forme inconnue avec un ensemble de références stockées en mémoire et décrites dans une représentation analogue. Les références sont obtenues lors d'une phase antérieure qualifiée d'apprentissage. Cette phase est très importante dans tout système de reconnaissance de l'écriture. Autrement dit, c'est un passage de l'espace observable vers un espace de décision d'appartenance à une classe.

La construction d'un système de reconnaissance de l'écriture comprend plusieurs étapes distinctes représentées par la Figure suivante :

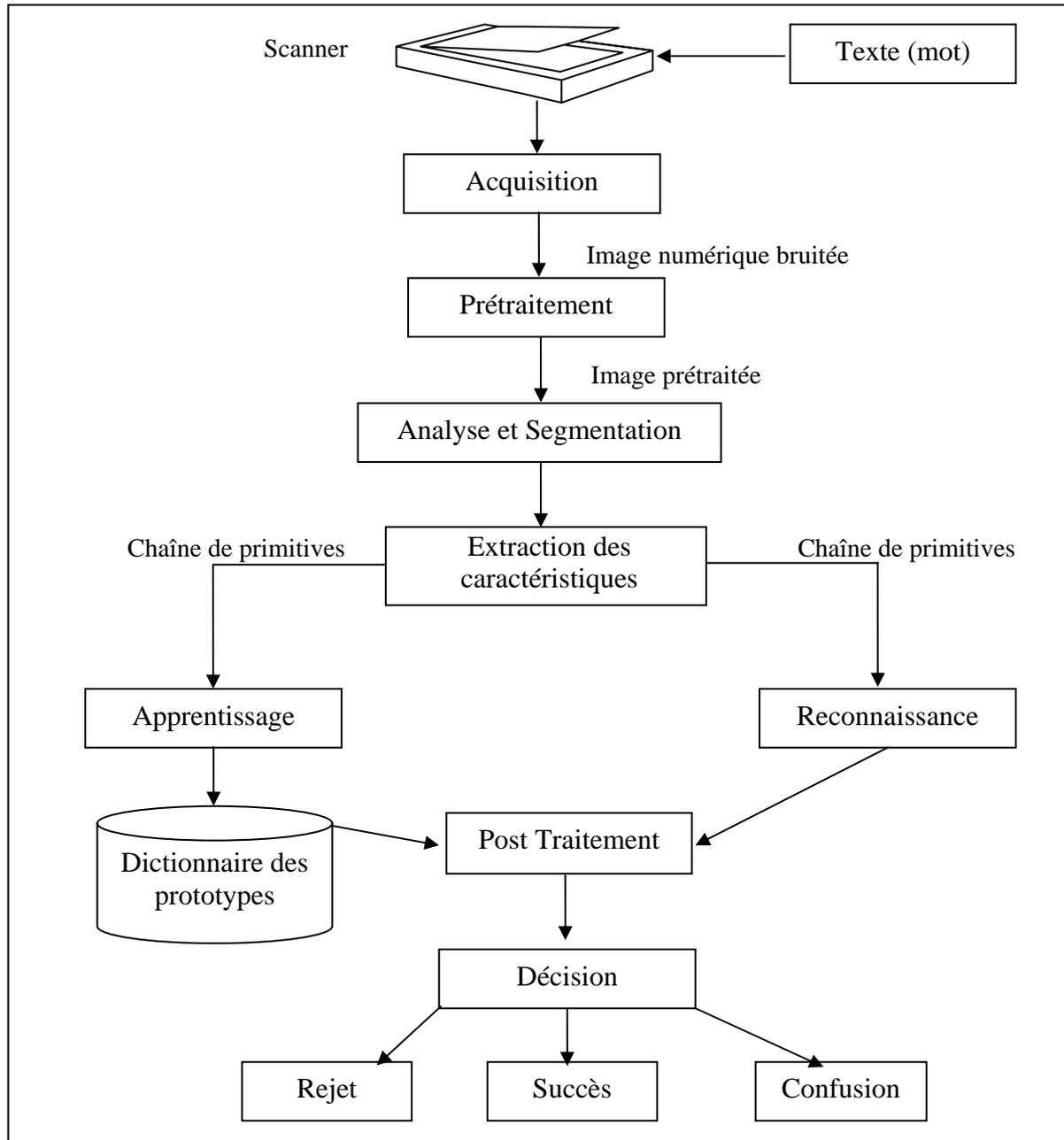


Figure I.1 : Système de Reconnaissance des caractères

4.1. Acquisition

C'est la première étape du processus de reconnaissance de forme, car elle permet de transformer l'objet d'une forme réel en une forme numérique compressible par les ordinateurs (Image) d'où l'appellation de digitalisation ou encore numérisation, l'opération se résume par la transformation d'un signal analogique en un signal numérique. Les systèmes d'acquisition les plus courants sont des scanners équipés de caméras linéaires (couleur ou monochrome). La

résolution des caméras varie entre 200 dpi et 800 dpi, une résolution suffisante pour conserver un trait d'écriture d'une épaisseur très petite.

La qualité du matériel d'acquisition peut influencer directement sur le pouvoir décisionnel du classifieur, car le bruit dans une image transmise par ce dernier peut fausser les résultats et même les opérations de prétraitement les plus évolués ne permettent pas une élimination totale du bruit.

Il existe deux types d'acquisition selon la manière d'utiliser l'image :

- **Acquisition physique** : Le système d'acquisition, le scanner en l'occurrence, est relié à l'ordinateur et est piloté par un logiciel. Ce dernier permet de digitaliser l'information analogique et stocke les données numériques dans un fichier graphique.
- **Acquisition Logique** : Le deuxième niveau d'acquisition, dit logique, consiste à exploiter le fichier graphique pour en extraire l'image. Elle se présente alors sous forme d'un tableau M à n lignes et p colonnes.

4.2. Prétraitements

Le prétraitement est une suite d'opérations sur l'image pour essayer de réduire le volume de donnée à traiter et d'éliminer le bruit introduit par les outils d'acquisition, la qualité du papier, l'encre ou encore l'effet du temps dans les documents anciens.

4.2.1. Binarisation

La binarisation est une opération qui permet de distinguer le fond de l'image et l'information manuscrite contenue dans l'image. Elle consiste à attribuer à chaque pixel de l'image une valeur de 0 ou 1 : '0' représente le noir (le texte), et le '1' représente le blanc (le fond de la page). Pour cela, elle applique en premier lieu l'opération de seuillage [THO 81].

Seuillage : Il consiste à déterminer la valeur du seuil à partir duquel tous les pixels ayant un niveau de gris inférieur à cette valeur sont représentés par un zéro "0" le noir, et tous les pixels de niveau de gris supérieur auront la valeur un "1" (le blanc).

La valeur du seuil est déterminée soit à partir de l'histogramme de niveau de gris de tous les pixels de l'image (analyse globale) si le contraste texte/fond est suffisamment marqué, alors elle sera égale à la valeur du niveau de gris se trouvant dans la vallée entre les deux pics de l'histogramme *figure II.2*. Soit par une analyse des contrastes locaux si les documents présentent des variations locales de contraste.

Le seuil doit être calculé d'une manière adéquate, car les composantes du texte liées des traits fins peuvent se déconnecter, ce qui modifie la forme originale du texte. En effet, le mauvais choix du seuil peut détruire une grande part d'information contenue dans l'image de l'enveloppe. Alors, une bonne binarisation doit être capable de conserver à la fois tous les caractères et les gravures sans récupérer trop de bruit.

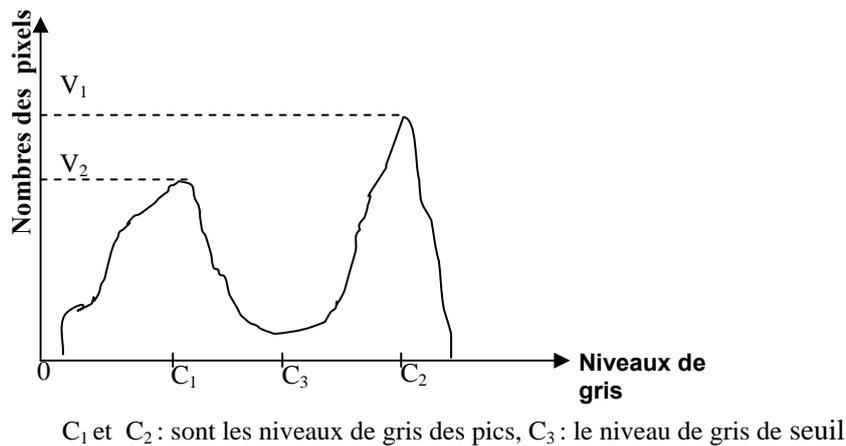


Figure 1.2 : Histogramme des niveaux de gris

4.2.2. Filtrage

Cette étape permet d'améliorer la qualité de l'image, pour cela il faut effectuer une opération de réévaluation de la valeur d'un pixel en fonction des pixels voisins.

4.2.3. Localisation de l'information manuscrite

Pour pouvoir reconnaître l'écriture manuscrite, il faut d'abord la localiser. Cette étape se devise généralement en trois niveaux différents de segmentation, correspondant à la localisation des zones d'intérêt sur une image, la localisation des lignes et enfin la localisation des mots des caractères ou même des graphèmes [OLL 99].

Le premier niveau consiste à déterminer les zones d'intérêt présentes sur un document en utilisant des connaissances a priori sur la nature du document traité (chèques, adresses, formulaires...) et en s'appuyant sur les règles classiques de disposition spatiale de l'écriture. La complexité du processus de localisation des informations manuscrites dépend alors du niveau d'information dont on dispose lors du traitement du document. Ce processus peut être relativement simple si la structure du document est fixe et connue (cas des chèques bancaires), en se basant sur les positions et les dimensions estimées des différents champs.

Lorsque la structure de document est plus complexe, la localisation des champs peut s'avérer beaucoup plus difficile.

Le deuxième niveau consiste à localiser les lignes d'écriture au sein des champs ou blocs de texte préalablement extraits. Pour cela, on s'appuie le plus souvent sur un modèle physique de disposition de l'écriture. Lorsque ce modèle est respecté, c'est-à-dire que les lignes d'écriture sont plus ou moins parallèles et horizontales, le processus d'extraction est relativement simple et des méthodes de détection basées généralement sur l'analyse d'histogrammes de projection déterminés sur l'ensemble du bloc de texte donnent des résultats très satisfaisants.

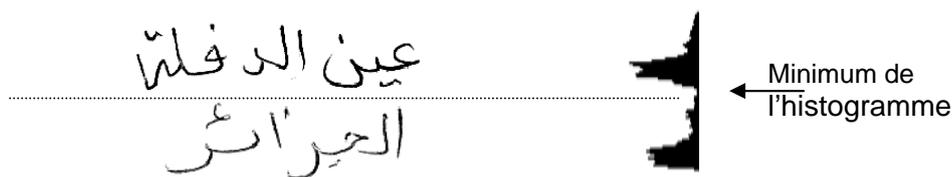


Figure 1.3 : Histogramme de la segmentation lignes.

Toutefois, lorsque le document manuscrit présente des dispositions variables, des inclinaisons différentes ou des lignes de bases fluctuantes, des techniques spécifiques doivent être mises en œuvre [HEU 03]. Dans tous les cas et quelles que soient les approches envisagées, des problèmes subsistent notamment lorsque les lignes d'écriture se superposent partiellement en présence d'extensions hautes ou basses qui s'étendent jusqu'à la ligne d'écriture inférieure ou supérieure.

Le troisième niveau consiste à localiser les mots se trouvant à l'intérieur des lignes précédemment extraites. L'écriture manuscrite étant naturellement découpée en mots, cette opération se base généralement sur la détection d'espaces à l'intérieur d'une ligne par une recherche de composantes connexes, ou par des histogrammes de projection. Comme il existe deux types d'espaces : les espaces intra mot et les espaces inter mots, on fait souvent usage d'une hypothèse de départ qui consiste à considérer que les espaces entre les mots sont plus grands que les espaces entre les lettres. L'utilisation d'heuristiques simples, telles que la comparaison à un seuil de décision intra-inter mots, permet de différencier entre ces deux types d'espaces.

Cependant, la localisation des mots sur une ligne d'écriture reste une opération délicate dans les systèmes de reconnaissance hors ligne d'écriture manuscrite non contrainte, car l'hypothèse évoquée ci-dessus est loin d'être vérifiée. Ainsi, le moyen idéal de résoudre ce problème serait de valider la segmentation en mots par la reconnaissance de la phrase et donc de se placer dans le cadre d'un principe de segmentation-reconnaissance de la phrase ou de la ligne de texte. Si cette approche est envisageable pour la reconnaissance de mots (où le problème se pose de la même façon pour localiser les lettres dans un mot), elle est en revanche difficilement applicable dans le cadre de la segmentation en mots à cause des innombrables combinaisons qu'elle engendrerait.

Ainsi, la majorité des approches proposées dans la littérature procèdent à une segmentation préalablement à la reconnaissance. L'objectif est alors de détecter tous les espaces dans une ligne sans distinction des espaces inter mots et des espaces inter caractères puis de réaffecter les espaces. Deux approches sont alors possibles : la première concerne les méthodes qui ont recours à une métrique adaptée au problème afin d'ordonner de la meilleure façon possible, les espaces détectés dans les lignes pour qu'une simple technique de seuillage puisse permettre de séparer les espaces inter mots des espaces inter caractères. La seconde approche met en œuvre une étape de pré reconnaissance pour attribuer les espaces détectés dans la phrase à l'une des deux classes, espace inter mots ou espace intra-mot, en utilisant des caractéristiques relatives à l'aspect de deux graphèmes consécutifs [HEU 03].

4.2.4. Redressement de l'écriture

Cette opération consiste à corriger la pente d'un mot ou à redresser l'inclinaison des lettres dans un mot afin de faciliter la segmentation. L'opération est effectuée à l'aide d'une transformation ligne par ligne où chaque pixel noir de coordonnées (x,y) est remplacé par les coordonnées (x',y') données par :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & -\tan \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (\text{I.1})$$

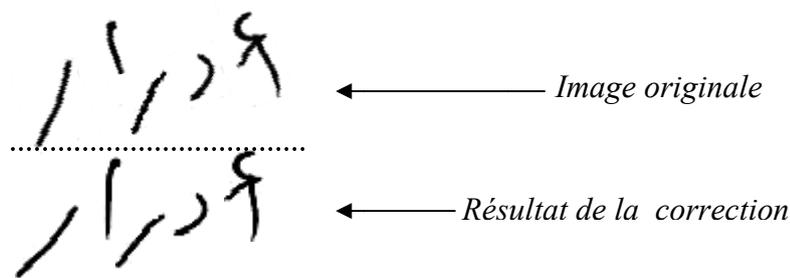


Figure I.4 : Correction de l'inclinaison (20°)

Le principe sur lequel repose ces deux étapes est le même et consiste, dans un premier temps, à détecter une inclinaison par rapport à l'horizontale (respectivement à la verticale) puis d'appliquer une transformation (en cisaillement) dans la direction perpendiculaire.

La détection de l'inclinaison moyenne d'un mot selon la direction horizontale est effectuée, soit globalement sur le mot, soit par morceaux en se basant sur la détection de la ligne de référence de l'écriture. La détermination des pentes des caractères peut être basée soit sur la localisation des portions de traits verticaux et le calcul de la moyenne des inclinaisons locales, soit sur les profils des projections verticales sur plusieurs directions, soit sur les chaînes de codage des contours [HEU 03, PAQ 00].

4.2.5. Lissage

L'image du caractère peut être entachée de bruit dont l'origine est la qualité du document ou le mode d'acquisition utilisé, ce qui conduit soit à des absences de points (trous), soit des points en plus (des empâtements). Les techniques de lissage permettent de réduire au maximum les discontinuités introduites par les bruits et de rétablir la régularité et la continuité du contour du mot. Cette technique (opération de nettoyage et bouchage) consiste à examiner le voisinage d'un pixel et de lui attribuer la valeur 1 si le nombre de pixels noirs dans cette zone est supérieur à un certain seuil.

Nous utilisons les notions suivantes pour la description des masques de transformation de voisinage : '1' pour un point de la forme et '0' pour un point du fond et X pour un point quelconque.

L'opération de nettoyage conduit à supprimer les petites tâches et les excroissances de la forme. Elle est réalisée de différentes manières suivant le type de bruit à enlever :

- Par élimination des points de la forme isolés ou situés à l'extrémité des contours, en appliquant sur l'image le masque suivant dans les huit directions

0	0	0
X	1	0
0	0	0

- Par élimination des points formant des angles droits ou des croisements du contour, en appliquant le masque suivant dans les huit directions :

X	0	0
1	1	0
0	0	0

Pour le bouchage, il s'agit d'égaliser les contours et de boucher les trous internes à la forme des caractères en lui ajoutant des points noirs. Nous pouvons citer :

- Le bouchage de trous isolés. Si le voisinage d'un point du fond correspond au masque suivant, alors ce point est mis à 1.

1	1	1
1	0	1
1	1	1

4.2.6. Squelettisation

Cette étape permet de ramener l'image du mot à une écriture linéaire d'une épaisseur égale à un pixel, en préservant la forme, la connexité et la topologie du tracé. La squelettisation permet une réduction considérable du nombre de données et optimise les étapes suivantes (temps de traitement) du traitement.

4.2.7. Normalisation de la taille des images

C'est une technique qui permet de ramener les caractères à la même taille, en effet à cause de la variation des fontes ou des opérations d'agrandissement ou de réductions de la taille des images, les caractères peuvent subir une légère déformation dans la taille, ce qui complique les tâches de segmentation et influence sur la stabilité des paramètres.

Cette technique opère sur deux phases. La première normalise le caractère en hauteur et la seconde en largeur. L'ordre de normalisation ainsi choisi évite que les caractères fins ne se déforment pas.

Il s'agit de transformer l'image du caractère de dimensions h_i, l_i en une image de dimensions h, l . Une étape intermédiaire consiste à produire une image de dimension h, l' , avec :

$$p = \frac{h}{h_i} \text{ et } l' = p.l \quad (\text{I.2})$$

La normalisation en hauteur est exécutée en transformant chaque pixel (x, y) noir de l'image du caractère en $(p.x, p.y)$. La normalisation en largeur de l'image ainsi obtenue se fait par l'examen de deux cas :

Si $l' < l$, alors l'image normalisée en hauteur est centrée dans une surface de dimension h, l .

Si $l' > l$, chaque pixel noir de l'image (x, y) subit la transformation en $((x / l'), l, y)$.

4.2.8. Extraction des contours

On appelle le point de contour, un pixel qui se situe sur la frontière séparant deux régions voisines. Dans les images à niveaux de gris, il est intéressant d'extraire le contour à l'aide d'un calcul de gradient. Ce contour est alors d'autant plus marqué que le niveau des pixels résultant du gradient est élevé. En revanche, dans les images binaires, il est plus avantageux d'utiliser un algorithme de suivi de contour, car il fournit directement une liste ordonnée de points.

a) Le gradient :

Le contour se manifeste dans l'image par des variations locales importantes des valeurs de niveaux de gris mis en évidence par des élévations de la dérivée première de la fonction image.

Le gradient d'une fonction $f(x, y)$ est défini par :

$$\nabla f(x, y) = \frac{\partial f}{\partial x} \vec{i} + \frac{\partial f}{\partial y} \vec{j} \quad (\text{I.3})$$

\vec{i} et \vec{j} sont les vecteurs unitaires sur x et y .

Dans le domaine discret les dérivées suivantes x et y peuvent être exprimées par les approximations suivantes :

$$A_x = f(x+1, y) - f(x, y) \text{ suivant } x \quad (\text{I.4})$$

$$A_y = f(x, y+1) - f(x, y) \quad \text{suivant } y \quad (I.5)$$

L'amplitude du gradient est donc

$$A = \sqrt{A_x^2 + A_y^2} \quad (I.6)$$

La *Figure suivante* montre l'exemple d'un mot avant et après l'extraction de son contour.



Figure I.5 : Exemple du mot « قالمة » avant et après l'extraction du contour.

b) Procédure de suivi de contour

L'algorithme de traçage du contour utilise le concept de 8-voisinage, ou 4-voisinage. Il peut être décrit comme étant un observateur qui traverse les pixels de l'image du texte (la composante connexe) d'un pixel à un autre, et sélectionné le plus à droite.

Le point de départ peut être détecté en parcourant la composante de droite à gauche et de haut en bas. L'algorithme s'achève quand le pixel courant coïncide avec celui de départ. (L'algorithme est détaillé dans [THO 81]. Ce type d'algorithme retrouve seulement le contour extérieur de la composante connexe, Figure suivante :



Figure I.6 : Exemple du mot « قالمة » avant et après l'extraction du contour extérieur.

Le contour est également utilisé comme étape préalable à un changement de représentation de l'information, en tant qu'empreinte des formes contenant une quantité réduite de données.

4.2.9. Extraction des composantes connexes

L'extraction des composantes connexes, procédure également appelée capture des connexités ou étiquetage des pixels, est largement utilisée en reconnaissance des Formes (RdF) pour segmenter les images binaires. La technique consiste à regrouper les pixels voisins dans un ensemble appelé composante connexe. Chaque ensemble est disjoint des autres et peut ensuite être aisément isolé. La 4-connexité est distinguée de la 8-connexité suivant que le critère de voisinage comprend les 4 ou les 8 voisins d'un pixel.

Il existe deux principaux algorithmes pour accomplir cette tâche :

- le premier est basé sur une procédure de suivi de contour : en parcourant le contour d'un objet et en revenant au point de départ, une composante connexe est délimitée, à l'exclusion cependant des contours intérieurs correspondant aux éventuels trous.
- Le second algorithme procède par une propagation d'un étiquetage des pixels lorsque l'on effectue un balayage des lignes et des colonnes de l'image.

4.3. Segmentation

La segmentation est une étape qui permet de passer du niveau du document au niveau de la ligne puis au niveau du mot pour finir au niveau le plus bas (caractère ou graphème). Les trois premiers types de segmentation étaient présentés dans la partie **Localisation de l'information manuscrite**. Cette partie consiste à localiser les lettres ou les graphèmes à l'intérieur des mots extraits dans les niveaux précédents. C'est une étape optionnelle, car il existe des méthodes de reconnaissance qui utilisent l'entité mot comme l'unité de base dans la classification. L'utilisation de telles méthodes est due à la difficulté de déterminer avec exactitude la frontière de séparation qui existe entre les caractères. Malheureusement, la segmentation du mot en caractères ou graphèmes est indispensable lorsque la taille du vocabulaire est importante.

Les techniques de segmentation proposées dans la littérature se divisent en deux types à savoir la segmentation implicite et la segmentation explicite dont nous allons les détailler par la suite.

4.3.1. Segmentation implicite

En s'inspirant des approches utilisées dans le domaine de la reconnaissance de la parole, la technique de la segmentation implicite a été introduite dans le domaine de la reconnaissance de texte. Cette méthode se base sur la division du signal d'entrée, qui est dans

notre cas l'écriture manuscrite, a un ensemble d'intervalles dans le pas et la dimension est fixée précédemment (un ou plusieurs pixels), la segmentation est totalement guidée par la reconnaissance des caractères ou des graphèmes (pas de pré segmentation). Dans ce cas le système recherche dans l'image des composantes connexes ou des groupements de graphèmes qui correspondent à ces classes de lettre.

Dans la littérature, on trouve deux manières d'utiliser cette méthode. La première appelée méthode par fenêtrage utilise une fenêtre mobile de largeur variable (d'où l'appellation) pour trouver des séquences de points de segmentations potentiels qui seront confirmés ou non par la reconnaissance de caractères. La variation de la taille et l'inclinaison de la fenêtre donnent plusieurs séquences qui seront traitées par la suite par la reconnaissance. Cette méthode efficace, s'avère trait difficile dans le paramétrage de la fenêtre mobile (hauteur, largeur, inclinaison), ce qui nécessite une expérimentation au préalable de toutes les hypothèses existantes.

La deuxième méthode, appelée méthode par recherche de primitives, utilise la détection des combinaisons de primitives qui permettent de donner par la suite une meilleure reconnaissance.

Dans les systèmes qui utilisent cette approche, il n'y a pas de module de segmentation parce qu'en réalité c'est le classifieur qui segmente d'une manière indirecte les mots. Les classifieurs les mieux adaptés à cette architecture sont les Modèles de Markov Caché.

4.3.2. Segmentation explicite

Dans cette approche le module de segmentation est nettement visible dans l'architecture des systèmes de reconnaissance. La communication avec le classifieur est unidimensionnelle c.-à-d. dès que la phase de segmentation est terminée, le système sort totalement de ce module.

Les études dans le domaine de la segmentation explicite du mot en caractère sont principalement basées sur l'analyse du contour. Les critères de segmentation que nous avons répertoriés sont liés à la détection des minimums du contour supérieur du mot. Les hypothèses de base sont les suivantes :

- les caractères composant le mot ne sont reliés que par un seul trait ;

- la segmentation peut être effectuée par une coupure verticale située au minimum local du trait de connexion suivi par une procédure de capture des composantes connexes.

La détection de ce minimum local est obtenue par le calcul de la dérivée du contour c'est-à-dire, dans l'espace discrétisé, de la différence des ordonnées de deux points d'abscisses consécutives. Puis le minimum est validé en fonction des critères d'unicité du trait de connexion et de son épaisseur, afin de ne pas couper un caractère comportant un minimum local à l'intérieur de son tracé.

Afin de renforcer les hypothèses de base, plusieurs autres critères ont été proposés dans différentes études incluant la segmentation des mots manuscrits : on peut aussi appliquer la technique des composantes connexes, procédure également appelée capture des connexités ou étiquetage des pixels.

La technique consiste à regrouper les pixels voisins dans un ensemble appelé composante connexe. Chaque ensemble est disjoint des autres et peut ensuite être aisément isolé.

Généralement dans les méthodes comportant une segmentation explicite du mot en lettres (méthodes de reconnaissance analytique), la première étape se résume dans la détermination de la zone médiane du mot, c'est-à-dire, par convention dans la suite du texte, la zone qui permet de distinguer les lettres à hampes ou à jambages, de celles qui en sont dépourvues. La méthode la plus simple utilisée est basée sur l'analyse de l'histogramme horizontal de l'image. L'histogramme permet de mettre aisément en évidence la zone médiane du mot, car la contribution des minuscules sans hampe ni jambage y est déterminante.

On cherche dans un premier temps la ligne de base où la ligne de l'écriture. On calcule la somme $S(i)$ suivante :

$$S(i) = \sum h(j) \quad (\text{I.7})$$

L'indice i correspondant à la ligne de base, où la somme $S(i)$ est maximum, notée « M ». Dans la plupart des cas, cette ligne d'indice i se trouve à l'intérieur de la zone médiane du mot, même si elle est parfois plus près d'un bord de la zone que de l'autre.

Dans un deuxième temps, on recherche dans la partie supérieure à la ligne d'indice M ainsi que dans la partie inférieure, les indices des minima de l'histogramme respectivement m_h

et m_b . Dans le cas idéal, ces deux minima délimitent la zone médiane. Mais, en pratique, l'histogramme est souvent étalé et dessine une lente décroissance autour de la zone médiane. On recherche donc plutôt les maxima de la dérivée de la fonction h afin d'obtenir l'indice pour lequel la variation est la plus importante.

La ligne de séparation entre les hampes et la zone médiane est obtenue au maximum de la dérivée de la fonction h_1 comprise entre m_h et M , tandis que la ligne de séparation entre la zone médiane et les jambages est obtenue au maximum de la valeur absolue de la dérivée de h_1 comprise entre M et m_b .

Lorsque la zone médiane du mot est obtenue, les lettres à hampe ou à jambage sont distinguées des lettres médianes quand elles dépassent les limites de la zone médiane d'une grandeur supérieure à une marge fixée.

Les deux modes de segmentation (externe et interne) des connaissances (lexicales, syntaxiques ou sémantiques) sont généralement utilisés afin d'appuyer la reconnaissance. Pour plus d'informations sur les techniques de segmentation, le lecteur peut se reporter aux références suivantes [KNE 01] [AKM 98].

Par ailleurs, les méthodes analytiques, par opposition aux méthodes globales, présentent l'avantage de pouvoir se généraliser à la reconnaissance d'un vocabulaire sans limites a priori, car le nombre de caractères est naturellement fini. De plus, l'extraction des primitives est plus aisée sur le caractère que sur la chaîne, le nombre de modèles à considérer est fini, les règles grammaticales associées sont plus faciles à établir, la complexité des calculs est généralement réduite. Cependant, la performance de segmentation affecte directement la fiabilité du système global : outre les erreurs de confusion, certaines erreurs de segmentation peuvent engendrer même des rejets.

4.4. Extraction des primitives (caractéristiques)

Les systèmes de reconnaissance utilisent des vecteurs de caractéristique pour pouvoir reconnaître les formes ou chaque vecteur se comporte comme une signature d'une forme. La difficulté de cette étape est de déterminer quelles caractéristiques employées pour obtenir un bon résultat de classification.

L'objectif de l'extraction et de la sélection des caractéristiques est d'identifier les caractéristiques qui sont importantes dans la discrimination de classes de formes. Cela signifie

de trouver des primitives les plus robustes possible, les plus pertinentes (afin de diminuer la taille du vecteur de primitives) et les moins coûteuses en calcul.

Le résultat de cette phase d'extraction de primitives est une séquence de symboles qui représente l'image binaire et qui traduit, soit la présence ou l'absence (cas booléen) ou la valeur associée (cas réel) à la caractéristique concernée.

En fonction de l'objectif fixé et de la méthode d'extraction choisie, l'approche de l'extraction des primitives peut être systématique ou heuristique.

La modélisation et le codage conduisent à une approche systématique dans la mesure où l'objectif fixé est la détermination d'une représentation complète de la forme, même de façon approximative. Dans la modélisation, les primitives sont obtenues a posteriori, par le résultat de l'approximation, en ce qui concerne le codage, les catégories de primitives sont définies a priori. Un test, qui est par exemple réalisé à l'aide d'une sonde, permet de valider la présence de chacune des primitives sur l'ensemble de la forme.

Le paramétrage conduit plutôt à une approche heuristique. Dans ce cas, on ne cherche pas nécessairement une représentation complète, mais seulement des indices significatifs. De même que dans le cas du codage, ces indices sont des primitives définies a priori.

L'une des raisons pour lesquelles cette étape pose un problème est que la plupart des techniques d'extraction s'accompagnent d'une perte d'information irrémédiable. De ce fait, il faut effectuer un compromis entre quantité et qualité de l'information, mais peu d'études théoriques sont faites sur ce sujet où l'intuition prévaut.

Pour un problème de classification donné, la principale qualité recherchée pour un ensemble de caractéristiques est sa faculté de rassembler les objets appartenant à une même classe dans une même partition de l'espace de représentation, tout en éloignant autant que possible les autres. Cette qualité est communément appelée pouvoir discriminant de l'ensemble de caractéristiques.

Dans la littérature, il existe un très grand nombre de travaux concernant l'extraction de caractéristiques pour la reconnaissance de l'écriture [TRI 96]. Nous allons citer les techniques les plus utilisées.

4.4.1. Caractéristiques globales

C'est une technique classique utilisée en traitement d'image, elle consiste à utiliser la caractéristique la plus simple dans une image, c'est-à-dire l'intensité de chaque pixel. Dans cette méthode il n'y a pas une extraction réelle des caractéristiques puisque le vecteur est construit à partir de tous les pixels de l'image. Une étape de normalisation de la taille de l'image est nécessaire afin de pouvoir effectuer la comparaison avec les modèles appris. Le degré de similarité entre les modèles décidera de l'appartenance de l'image à une classe.

4.4.2. Le développement en série

On peut distinguer dans cette catégorie trois types de technique :

- **La transformée de Fourier** : c'est l'une des techniques les plus utilisées dans la reconnaissance des formes et spécialement les caractères. Les caractéristiques extraites sont les descripteurs de Fourier basés sur les coefficients complexes des séries de Fourier.

La transformée peut être calculée directement sur l'image du caractère (bidimensionnelle) ou appliquée sur le contour du caractère (mono dimensionnelle). Cette technique est caractérisée par la propriété d'invariance aux rotations ce qui implique des problèmes de reconnaissance de certains caractères.

- **La transformation en Ondelette** : Une autre transformation assez proche de celle de Fourier est celle en Ondelette. Le principal intérêt des Ondelette est qu'elles permettent d'obtenir une information fréquentielle localisée concernant un signal ou une fonction de base choisie. Ce type de données est particulièrement intéressant pour la classification. Malgré certains avantages, cette technique est peu utilisée en reconnaissance de formes. La raison est que les caractéristiques extraites ne sont pas invariantes à la translation. En d'autres termes, un léger décalage du caractère (ou mot) donnera des coefficients d'Ondelette totalement différentes.
- **Les moments invariants** : L'invariance recherchée est celle à la rotation, à la translation et au changement d'échelle. Les caractéristiques extraites par ces techniques sont considérées comme le résultat d'une transformation globale appliquée uniquement aux pixels de la forme analysée. De par leur nature, elles sont très peu sensibles aux variations locales de la forme, ce qui explique leur grande utilisation. Il existe plusieurs formulations des moments invariants, comme celles de Hu et celle de Li. Cependant, les plus utilisées actuellement sont celles dérivées des polynômes

complexes de Zernike, car elles ont des performances supérieures en termes d'invariance.

4.4.3. Concavités, convexités et occlusions

Dans cette catégorie on utilise les propriétés topologiques et géométriques de la forme analysée en effectuant l'extraction des concavités, des convexités et des occlusions. Il y a deux techniques principales pour l'extraction de caractéristiques dans cette catégorie :

- La première utilise une image binaire et procède par étiquetage du fond de cette image (pixels blancs), elle consiste à attribuer à chaque pixel une étiquette de la vision de la concavité (ouverte, fermer, direction).
- La deuxième est appliquée sur l'image du contour des formes à analyser. Le suivi du contour extérieur permet de détecter les parties concaves et convexes de la forme. La présence de contours intérieurs correspond à une détection d'occlusion.

4.4.4. Allongements horizontaux et verticaux

En se basent sur les propriétés structurelles des caractères, la technique utilise une projection des pixels du caractère sur un axe perpendiculaire a la direction de recherche des allongements, la position et la valeur de ces derniers est obtenu par détection des maxima sur l'histogramme. Cette technique présente l'inconvénient de détecter des maxima qui ne sont pas des allongements.

4.4.5. Caractéristiques locales

Cette catégorie utilise les fins de trait, les jonctions et les intersections pour extraire la structure des caractères. Elle se base sur l'application de plusieurs masques sur l'image entière dans le cas de l'écriture manuscrite afin d'extraire les particularités citées.

4.4.6. Intersection avec des droites

Cette catégorie utilise l'intersection des caractères avec une ou plusieurs droites afin d'extraire les propriétés projectives des caractères. La technique est très difficile à paramétrer, en effet pour obtenir des résultats significatifs il faut ajuster le nombre de droite, la position des droites, la longueur des droites ainsi que l'orientation de ces derniers.

4.4.7. Caractéristiques géométriques

Cette catégorie permet de prendre les propriétés métriques des caractères en utilisant les mesures de distance entre les pixels d'un caractère les plus éloignés suivant les axes

horizontaux ou verticaux. Une étape de normalisation de l'image est nécessaire pour une bonne classification.

4.4.8. Caractéristiques de densités et de zoning

Cette catégorie de technique utilise la densité de pixel des caractères dans l'image ou dans certaines parties. Elle consiste à effectuer un découpage de l'image en zone dans laquelle on effectue une estimation de la densité de pixels. La technique qui se base sur la perception humaine présente la difficulté du choix du nombre de zones à utiliser.

4.5. Classification

La classification est l'élaboration d'une règle de décision qui transforme les attributs caractérisant les formes en appartenance à une classe (passage de l'espace de codage, vers l'espace de décision). Avant qu'un modèle de décision ne soit intégré dans un système de reconnaissance de l'écriture, il faut avoir procédé auparavant à deux étapes : l'étape d'apprentissage et l'étape de test.

L'étape d'apprentissage consiste à caractériser les classes de formes de manière à bien distinguer les familles homogènes de formes. C'est une étape clé dans le système de reconnaissance. On distingue deux types d'apprentissage : apprentissage supervisé et apprentissage non supervisé [KOE 02].

Dans le cas de l'apprentissage supervisé, un échantillon représentatif de l'ensemble des formes à reconnaître est fourni au module d'apprentissage. Chaque forme est étiquetée par un opérateur appelé professeur, cette étiquette permet d'indiquer au module d'apprentissage la classe dans laquelle le professeur souhaite que la forme soit rangée. Cette phase d'apprentissage consiste à analyser les ressemblances entre les éléments d'une même classe et les dissemblances entre les éléments de classes différentes pour en déduire la meilleure partition de l'espace des représentations. Les paramètres décrivant cette partition sont stockés dans une table d'apprentissage à laquelle le module de décision se référera ensuite pour classer les formes qui lui sont présentées.

Dans le cas de l'apprentissage non supervisé, on fournit au système de reconnaissance un grand nombre de formes non étiquetées. L'étape de la classification va se charger d'identifier automatiquement les formes appartenant à une même classe.

Dans le domaine de la reconnaissance de l'écriture, ce sont les méthodes basées sur un apprentissage supervisé qui sont le plus souvent utilisées ; et plus particulièrement pour les caractères manuscrits isolés, car les classes naturelles sont connues et en nombre limité.

L'étape de test permet d'évaluer la performance du classifieur pour un apprentissage donné. C'est une étape importante, car elle peut mettre en cause le choix des primitives ou le choix de la méthode d'apprentissage. En effet, il est difficile de trouver a priori les primitives pertinentes et la méthode d'apprentissage la plus adaptée au problème posé, d'où l'utilité de procéder par itérations successives. Ces itérations consistent à extraire des primitives jugées utiles au problème de reconnaissance à résoudre et à tester la performance du système avec cet ensemble de primitives. Au fur et à mesure que les performances du système souhaitées ne sont pas atteintes alors il suffit de trouver à nouveau une nouvelle famille de primitives ou de combiner les primitives extraites avec de nouvelles primitives.

Ces phases d'apprentissage et de test sont réalisées préalablement à l'intégration du module de décision dans le système de reconnaissance. Dans tous les cas, on peut permettre au système de reconnaissance d'itérer les phases d'apprentissage et de test tant qu'on n'a pas atteint les performances désirées. Le calcul de cette performance est le résultat du classifieur utilisé. Pour construire un classifieur, il existe trois approches : structurelle, statistique et hybride.

4.5.1. Les approches de classification

Dans cette section nous allons donner un bref aperçu sur les principales méthodes de classification utilisées en reconnaissance de l'écriture, et particulièrement celle des mots cursifs.

4.5.1.a. Les approches statistiques

Cette approche consiste à déterminer des caractéristiques extraites d'une forme pour les caractériser d'une manière statistique. Elle a besoin d'un nombre élevé d'exemples afin de réaliser un apprentissage correct des lois de probabilité des différentes classes. Autrement dit, cette approche bénéficie des méthodes d'apprentissage automatique s'appuyant sur des bases théoriques connues telles que la théorie de la décision bayésienne, les méthodes de classification non supervisées et l'analyse en composantes principales. Les deux principales familles de méthodes utilisées sont les méthodes paramétriques et les méthodes non paramétriques.

- Les méthodes paramétriques opèrent sous l'hypothèse que les classes étudiées suivent une distribution de probabilité d'une certaine forme connue a priori. La prise de décision consiste à déterminer la classe pour laquelle la forme inconnue présente la probabilité d'appartenance maximale. Elles exigent des bases d'apprentissage assez importantes pour une estimation correcte des paramètres de la distribution supposée. L'approche statistique englobe : la règle de Bayes [BEL 92], les méthodes neuronales et les chaînes de Markov [ELY 96].

- Les méthodes non paramétriques, les lois de probabilité sont inconnues pour une des classes. Le problème revient à établir des frontières de décision entre les classes. Les techniques les plus utilisées en reconnaissance de formes sont : la méthode du plus proche voisin [BEL 92], et la méthode d'appariement de graphes [BEL 92]. Pour plus d'amples et d'informations, Kunt dans [KUN 00] décrit un ensemble de méthodes statistiques en reconnaissance de formes.

Malgré leur nature différente, les approches statistiques et structurelles peuvent être combinées aux mêmes domaines d'application. Le choix d'une approche peut être lié à des contraintes matérielles telles que la taille de la base d'apprentissage disponible, le temps de calcul requis et la taille mémoire nécessaire. L'utilisation conjointe des deux approches peut être une solution optimale pour le problème de reconnaissance de l'écriture.

4.5.1.b. Les approches structurelles

Ces approches consistent à mettre en relation la structure des formes analysées et la syntaxe d'un langage formel. La description des formes est réalisée par l'intermédiaire de phrases et le problème de classification est ramené à un problème d'analyse de grammaire (parsing). De manière générale, les approches syntaxiques ou structurelles permettent la description de formes complexes à partir de formes élémentaires. Ces dernières, encore appelées caractéristiques, sont extraites directement des données présentes en entrée du système. La différence principale entre ces méthodes et les méthodes statistiques est que ces caractéristiques sont des formes élémentaires et non pas des mesures. Une autre différence est qu'elles introduisent la notion d'ordre dans la description d'une forme. Les méthodes les plus répandues utilisent le calcul de distance d'édition entre deux chaînes et la programmation dynamique.

4.5.1.c. Les approches hybrides

Une dernière approche est envisageable : l'approche hybride. La reconnaissance par modèles de Markov cachés on est une. En effet, ils utilisent une approche statistique tout en ayant la possibilité d'utiliser des descriptions structurelles. Leur application dans le domaine de la reconnaissance de l'écriture est de plus en plus présente. [ELY 96] [LER 97], après leur succès en reconnaissance de la parole. Une description complète de l'application de ces modèles à la reconnaissance de mots manuscrits est présentée dans [ZER 00]. Il existe également des approches couplant différentes techniques :

Modèle de Markov et réseaux de neurones, modèles de Markov cachés et k- plus proches voisins.

4.6. Post traitement

Le post traitement correspond au traitement ultérieur à la reconnaissance, pour prendre en compte les nombreuses relations d'ordre logique, lexical, syntaxique ou sémantique entre les différents composants d'un document. Le rôle de la phase de post traitement consiste à corriger les erreurs de l'étape de reconnaissance, en tenant compte des informations lexicales, syntaxiques, et sémantiques propres au document [OLL 99].

- **Utilisation de l'information lexicale.** Le contrôle lexical vérifie la présence d'un mot dans un lexique de référence. Si la reconnaissance s'est effectuée au niveau caractère, le post traitement consistera, par exemple à vérifier à l'aide d'un dictionnaire l'exactitude des mots obtenus en combinant les différentes hypothèses fournies par l'étape de reconnaissance. On attribue ensuite à chaque mot présent dans le lexique une valeur de confiance qui dépend de celles attribuées par l'étape de reconnaissance aux différents caractères composant ce mot.
- **Utilisation de l'information syntaxique.** L'utilisation d'un ensemble de règles grammaticales permet de prendre en compte le contexte syntaxique d'un document afin de confirmer ou infirmer la séquence de mots proposés. Cette approche est souvent appliquée dans les systèmes de lecture des montants de chèques bancaires, pour lesquelles les règles syntaxiques ne sont pas trop nombreuses [OLL 99].
- **Utilisation de l'information sémantique.** Lorsque la nature du problème le permet, il est également possible de mettre en corrélation des résultats obtenus sur des champs différents de l'image, mais codant la même information (cas des chèques bancaires, des bons de commande ou des enveloppes postaux). Il existe, à l'heure actuelle, un certain nombre de travaux visant à exploiter au mieux la redondance d'information

entre les champs numériques et littéraux des chèques bancaires ou entre le code postal et le nom de ville dans les adresses postales.

4.7. La prise de décision

Les méthodes de reconnaissance vues précédemment sont utilisées pour décrire les règles de prise de décision, cette dernière peut-être :

- **Caractère ou mot reconnu** : Si le système arrive à associer un et un seul prototype au caractère ou mot à reconnaître, il prend une décision unique.
- **Caractère ou mot ambigu** : Si le système associe plusieurs prototypes au caractère ou mot à reconnaître, il propose ainsi plusieurs choix avec une confiance de même ordre.
- **Caractère ou mot rejeté** : Si le système n'arrive pas à associer un prototype au caractère ou mot à reconnaître : il ne prend aucune décision de classification. On définit ainsi plusieurs facteurs de performance de système tel que :
 1. **le taux de reconnaissance** : qui représente le pourcentage de caractères ou mot reconnus parmi ceux présentés au système.
 2. **Le taux de substitution** : qui présente le pourcentage de caractères ou mots acceptés par le système, mais classés de façon incorrecte (à cause d'un problème d'ambiguïté).
 3. **Le taux de rejet** : qui représente le pourcentage de caractère ou mots rejetés (non reconnus) parmi ceux représentés à l'entrée du système.

4.7.1. Évaluation des performances d'un classifieur

Nous décrivons dans ce paragraphe les critères les plus souvent employés pour évaluer la performance des systèmes de reconnaissance en phase de généralisation. L'objectif est d'obtenir une estimation la plus fidèle possible du comportement du système dans des conditions réelles d'utilisation. Pour cela des critères classiques comme les taux de reconnaissance et taux d'erreurs sont presque systématiquement utilisés [RAG 03]. Mais d'autres critères comme la fiabilité ou la complexité apportent aussi des informations utiles.

4.7.2. Taux de reconnaissance et taux d'erreurs

Les taux de reconnaissance et d'erreurs permettent d'évaluer la qualité du classifieur f_M par rapport au problème pour lequel il a été conçu. Ces taux sont évalués grâce à une base de test qui contient des formes d'écritures dans le même espace de représentation E que celles

utilisées pour l'apprentissage. Elles sont aussi étiquetées par leur classe réelle d'appartenance afin de pouvoir vérifier les réponses du classifieur. Pour que l'estimation du taux de reconnaissance soit la plus fiable possible, il est primordial que le reconnaisseur n'ait jamais utilisé les échantillons de cette base pour faire son apprentissage (la base de test ne doit avoir aucun individu en commun avec la base d'apprentissage et les éventuelles bases de validation). De plus, cette base de test doit être suffisamment représentative du problème de classification.

En général quand les échantillons étiquetés à disposition sont suffisamment nombreux ils sont séparés en deux parties disjointes et en respectant les proportions par classes de la base initiale. Une partie sert pour former la base d'apprentissage et l'autre pour former la base de test. Le découpage le plus courant est de 2/3 pour l'apprentissage et le 1/3 restant pour la base de test. Les performances en terme de taux de reconnaissance sont alors déterminées en présentant au classifieur chacun des exemples e_j de la base de test et en comparant la classe donnée en résultat $f_M(e_j) = s$ à la vraie classe de e_j . En considérant que la base de test contient N individus et que sur ceux-ci $N_{corrects}$ sont bien classés par le système le taux de reconnaissance τ_{reco} est simplement défini par :

$$\tau_{reco} = \frac{N_{corrects} \times 100}{N} \quad (\text{I.8})$$

Le taux d'erreur τ_{err} est défini à partir du nombre d'individus N_{err} mal classés.

$$\tau_{err} = \frac{N_{err} \times 100}{N} \quad (\text{I.9})$$

La répartition des exemples bien et mal classés est illustrée sur le *Tableau* suivant :

Base de test : N exemples	
$N_{corrects}$	N_{err}

Tableau I.1 Répartition des exemples dans la base de test cas général

5. Problèmes liés à l'OCR

La tâche de l'OCR n'est pas aisée, divers problèmes compliquent le processus de reconnaissance, parmi lesquels nous citons les plus importants [AKR 98] :

- ❖ **La qualité du document** : un document télécopié ou photocopie plusieurs fois est plus difficile à traiter que la copie originale. L'écriture devient plus mince ou au contraire plus

épaisse, dégradée avec les parties de texte qui manquent ou des tâches qui apparaissent, des ouvertures ou des bouchages de boucles.

- ❖ **L'impression** : un document composé est de meilleure qualité qu'un document dactylographié qui, à son tour, est plus clair qu'un texte issu d'une imprimante matricielle. Une imprimante à jet d'encre peut introduire des taches d'encre et un étalement des caractères, un laser peut générer des lignes ou des fonds.
- ❖ **Le support de l'information** : tel que le papier, joue également sur les performances de la reconnaissance par sa qualité : son grammage, sa granulation et sa couleur.
- ❖ **L'acquisition** : la numérisation en temps réel introduit souvent des distorsions dans l'image. Dans le cas hors ligne, la qualité du texte numérisé est un compromis entre les variations de la position du papier, la propriété de la vitre du dispositif de numérisation et sa résolution.

Les problèmes posés par la reconnaissance optique de l'écriture manuscrite, objet de notre travail, sans aucune contrainte sont plus complexes que ceux liés à l'écriture imprimée. Les erreurs de lecture dans le cas du manuscrit sont dues aux variations infinies de l'écriture de nature aléatoire, qui dépendent de plusieurs facteurs.

6. Les difficultés de reconnaissance de l'écriture manuscrite

Sur le plan méthodologique, l'évaluation de la complexité de certains problèmes à traiter doit prendre en compte plusieurs critères :

6.1. La disposition spatiale du manuscrit

La disposition spatiale peut être influée par des contraintes externes, qui peuvent être imposées au scripteur, et des contraintes internes à l'écriture proviennent des habitudes propres à chaque scripteur.

Les contraintes externes conduisant à une écriture :

- Prés casés : l'utilisation doit s'efforcer d'écrire à l'intérieur des cases prédéfinies.
- Zonée : le scripteur écrit dans des zones bien limitées.
- Guidée : dans ce cas, l'écriture est guidée par l'existence d'une ligne support.
- Générale : correspondant à une écriture à emplacement totalement libre.

Parmi les contraintes internes, on trouve :

- Lettres séparées.
- Groupes de lettres liées à l'intérieur d'un mot.
- Mots entièrement liés.
- Plusieurs de ces cas à l'intérieur d'un même texte manuscrit.

6.2. Nombre de scripteurs

La difficulté de mise en œuvre d'un système de reconnaissance d'écriture croît également avec sa portée d'utilisation, c'est-à-dire avec le nombre d'utilisateurs ou scripteurs.

Donc, nous pouvons citer les systèmes suivants :

a) Système mono-scripteur

C'est le système le plus simple avec apprentissage de l'écriture propre à l'utilisateur considéré.

b) Système multi-scripteur

Ce système présente un degré de complexité supérieur, dans ce cas le système doit s'adapter à l'écriture de plusieurs scripteurs.

c) Système multi-mono-scripteurs

C'est une simple extension du cas mono-scripteurs, le système s'adapte à l'écriture de chaque utilisateur après que celui-ci se soit fait connaître par son nom.

d) Système omni-scripteurs

C'est le système le plus général, car il s'agit de reconnaître l'écriture de n'importe quel utilisateur sans que la notion d'identité du scripteur n'intervienne.

6.3. Taille du vocabulaire

Dans ce cas le problème à résoudre dépend de la taille du vocabulaire à reconnaître. Tel qu'on peut distinguer deux catégories d'applications :

a) Les applications à vocabulaire limité

Où le nombre de mots à reconnaître constitue un lexique de taille réduite (<1000 mots), la complexité est moindre, car la réduction du nombre limite l'encombrement mémoire et favorise l'utilisation de méthodes de reconnaissance directes et donc rapides, par balayage systématique de l'ensemble des mots du lexique.

b) Les applications à vocabulaire très étendu

Où le nombre de mots à reconnaître constitue un lexique étendu (>1000 mots) formant un dictionnaire, cela va provoquer des problèmes d'encombrement de la mémoire et de temps d'accès à chaque mot. Dans ce cas, les seules méthodes efficaces sont des approches de type

recherches arborescentes avec raffinements successifs.

D'autres types de critères peuvent influencer sur la complexité des systèmes de reconnaissances. Ils sont relatifs aux variations intrinsèques de l'écriture, dans un contexte cursif. Parmi ces variations, on peut noter celles :

- ❖ **Propres au scripteur** : traduisant le style personnel en termes de rapidité, de continuité et de régularité. Tous ces éléments influent sur la forme des lettres (écriture penchée).
- ❖ **Propres à l'écriture manuscrite** : la forme d'une lettre dépend de sa position dans le mot (début, milieu, fin) ainsi que des lettres voisines. Toutes ces variations vont conduire à des formes morphologiques différentes d'une même lettre, appelées allographes.

6.4. Contraintes morphologiques

Les difficultés de mise en œuvre d'un système de reconnaissance d'écriture augmentent en fonction de sa capacité plus ou moins grande à prendre en considération plusieurs types de modèles distincts, et à tolérer des variantes par rapport à ces prototypes afin d'imposer de moins en moins de contraintes à l'utilisateur. On peut distinguer trois niveaux successifs :

- Un premier niveau correspondant à la reconnaissance d'un nombre limité d'entités isolées.
- Un second niveau correspondant à la reconnaissance d'entités plus globales.
- Le troisième niveau est celui de la reconnaissance globale des textes manuscrits qui peuvent comporter plusieurs types d'écritures différentes comme des retouches.

7. Conclusion

Dans ce chapitre nous avons donné les différents types de reconnaissance, ensuite nous avons cité d'une manière générale les méthodes utilisées dans la reconnaissance de l'écriture manuscrite. Nous avons également détaillé chaque étape du processus de reconnaissance de mot manuscrit avec les approches utilisées dans chaque étape. Nous avons aussi donné une description des règles de prise de décisions, et comment évalué les performances d'un système de reconnaissance. Et enfin, nous avons énuméré les principaux problèmes rencontrés par l'OCR.

CHAPITRE I RECONNAISSANCE AUTOMATIQUE DE L'ECRITURE

1. Introduction	5
2. Reconnaissance en ligne ou hors-ligne	5
2.1. La reconnaissance en ligne (on line)	6
2.2. La reconnaissance hors-ligne (off line)	6
3. Méthodes de reconnaissance	6
3.1. L'approche globale	7
3.2. L'approche analytique	7
3.3. L'approche pseudo-analytique	8
3.4. L'approche perceptuelle	8
4. Organisation générale d'un système de reconnaissance	8
4.1. Acquisition	9
4.2. Prétraitements	10
4.2.1. Binarisation	10
4.2.2. Filtrage	11
4.2.3. Localisation de l'information manuscrite	11
4.2.4. Redressement de l'écriture	13
4.2.5. Lissage	14
4.2.6. Squelettisation	15
4.2.7. Normalisation de la taille des images	15
4.2.8. Extraction des contours	16
4.2.9. Extraction des composantes connexes	18
4.3. Segmentation	18
4.3.1. Segmentation implicite	18
4.3.2. Segmentation explicite	19
4.4. Extraction des primitives (caractéristiques)	21
4.4.1. Caractéristiques globales	23
4.4.2. Le développement en série	23
4.4.3. Concavités, convexités et occlusions	24

4.4.4. Allongement horizontaux et verticaux	24
4.4.5. Caractéristiques locales	24
4.4.6. Intersection avec des droites	24
4.4.7. Caractéristiques géométriques	24
4.4.8. Caractéristiques de densités et de zoning	25
4.5. Classification	25
4.5.1. Les approches de classification	26
4.5.1.a. Les approches statistiques	26
4.5.1.b. Les approches structurelles	27
4.5.1.c. Les approches hybrides	28
4.6. Post traitement	28
4.7. La prise de décision	29
4.7.1. Evaluation des performances d'un classifieur	29
4.7.2. Taux de reconnaissance et taux d'erreurs	29
5. Problèmes liés à l'OCR	30
6. Les difficultés de reconnaissance de l'écriture manuscrite	31
6.1. La disposition spatiale du manuscrit	31
6.2. Nombre de scripteurs	32
6.3. Taille du vocabulaire	32
6.4. Contraintes morphologiques	33
7. Conclusion	33

Chapitre II

Travaux existants

1. Introduction

Le domaine de la reconnaissance de l'écriture et spécialement l'écriture manuscrite a eu beaucoup d'intérêt scientifique dans ces dernières années. Le développement des systèmes capable de reconnaître des mots manuscrits arabes n'a pas eu cette chance, de tels systèmes n'ont vu le jour que récemment avec des études théoriques superficielles et des systèmes ORCA qui n'ont pas quitté le stade de l'expérimentation malgré les taux élevés de reconnaissance.

Ce retard par rapport aux langues latines peut s'expliquer par l'absence de base de données de large vocabulaire unifié. En effet, chaque laboratoire utilise sa propre base ce qui empêche une amélioration de ces systèmes par des chercheurs en dehors de ces labos. Un autre inconvénient vient de l'incapacité de comparer les travaux réalisés vu qu'ils ont été utilisés dans des circonstances variées. Nous avons essayé dans ce chapitre de donner les différents systèmes réalisés pour faire la reconnaissance automatique de l'écriture arabe.

2. Système de Miled et al. 1998

Dans [MIL 98a], Miled décrit une stratégie de reconnaissance de mots manuscrits qui peut être qualifiée d'hybride, car la modélisation est basée sur les trois niveaux perceptifs : global pseudo-analytique et analytique, représentés, respectivement par une modélisation markovienne des mots, sous-mots et caractères.

Au niveau global, un mot est vu comme étant formé de deux zones: les tracés principaux et les diacritiques.

Des caractéristiques de type indices visuels [CHE 98] sont utilisées, en distinguant deux catégories d'indices: ceux correspondant aux tracés principaux (alifs, boucles, ascendants, descendants, poches, vallées) et ceux correspondant au tracé secondaire (points simples et multiples). Les mots sont décrits par une séquence chronologique (de droite à gauche) d'indices visuels codés. Cette séquence se transforme en une séquence d'observations

émise par le modèle markovien du mot. La topologie choisie est gauche-droite, avec un nombre d'états par modèle égal à l'espérance mathématique des longueurs des séquences d'observations par classe de mots.

Le traitement dans chacun des deux autres niveaux perceptifs, analytique et pseudo-analytique, peut être divisé en trois procédures:

- Segmenter les tracés en graphèmes et fournir la représentation vectorielle de ces graphèmes. Dans [MIL 98a, MIL 98b, MIL 98c, MIL 00], chaque graphème est représenté par un ensemble de 19 composantes dont 9 correspondent à des caractéristiques perceptuelles: boucle, ouverture, dépassement haut, dépassement bas..., et 10 correspondent aux 10 premiers descripteurs de Fourier.
- Passer au niveau logique du traitement, en affectant, à chaque graphème, une observation au sens logique des MMC. Ainsi, chaque graphème est remplacé par l'étiquette de sa classe d'appartenance.
- Élaborer une modélisation markovienne adaptée à chacun des deux niveaux perceptifs.

Dans la modélisation pseudo-analytique, chaque sous-mot qui est considéré comme une suite de graphèmes. Il est modélisé par un MMC gauche droite et un mot sera le résultat de la concaténation des sous-mots qui le composent.

Dans la modélisation analytique, c'est le caractère qui est considéré comme une suite de graphèmes. Il est modélisé par un MMC gauche droite à trois états (avec l'hypothèse qu'un caractère peut être composé, au maximum, de trois graphèmes). Un sous-mot sera, dans ce cas, le résultat de la concaténation des modèles de caractères qui le composent et un mot sera le résultat de celle des sous-mots qui le composent.

Testés sur une base de 5900 mots pour un lexique de 232 villes tunisiennes, les taux de reconnaissance des classifieurs globaux, pseudo-analytique et analytique sont, respectivement: 67,8%, 72,5% et 81,8%.

Le système proposé dans [MIL 98a] est basé sur une combinaison de classifieurs issus des trois niveaux évoqués précédemment. Plusieurs méthodes de combinaison de classifieurs ont été évaluées, mais la stratégie adoptée pour la coopération est plutôt hiérarchique, fondée sur une accumulation d'informations d'un niveau perceptif à l'autre. Une stratégie hiérarchique descendante (du niveau global au niveau analytique) a permis d'atteindre un taux de

reconnaissance de 89,4%, soit près de 8% de plus que le meilleur des trois classifieurs, pris individuellement.

3. Système d'Essoukhri Ben Amara 1999

Dans [ESS 99], l'auteur propose un système de reconnaissance de mots arabes imprimés fondé sur une modélisation globale, pseudo-bidimensionnelle de sous-mots par des modèles de Markov cachés planaires (MMCP ou PHMM). Les sous-mots considérés sont extraits du vocabulaire des noms de villes tunisiennes.

L'image d'un mot est d'abord divisée en sous-mots, chacun est modélisé par un MMCP dont la topologie permet de modéliser les variations telles que les élongations des ligatures horizontales et la présence des ligatures verticales.

Une image d'un sous-mot est décomposée verticalement en une ou plusieurs zones selon la topologie de la forme considérée. Les zones correspondent aux bandes de variation de l'écriture arabe, mettant ainsi en évidence des caractéristiques morphologiques donnant le contexte de type géographique des différents caractères qui constituent le sous-mot à savoir, dans l'ordre (du haut vers le bas) : ascendants, points diacritiques supérieurs, boucles et/ou lieu de ligatures verticales, points diacritiques inférieurs et/ou lieu de ligature verticale et jambage.

Un sous-mot est modélisé par un modèle vertical de Bakis haut-bas strict de super-états correspondants aux zones définies précédemment. Les transitions d'un super-état à l'autre expriment le passage d'une bande d'informations à l'autre. La durée dans un super-état est assimilée au nombre de lignes analysées par le modèle. Pour cela, une fonction de durée spécifique a été définie permettant au modèle, lors de la reconnaissance, de décider de la répartition des lignes dans les super-états.

Chaque zone est modélisée par un modèle horizontal secondaire MMC discret unidimensionnel de type gauche-droite dont le nombre d'états est étroitement lié à la topologie de la bande correspondante, il suit les variations du nombre de transitions noir/blanc dans la bande associée. Ainsi, les observations des modèles horizontaux des bandes sont réalisées sur les différents segments de l'image, un segment étant un ensemble de pixels voisins de la même couleur se trouvant sur la même ligne. La durée dans les modèles secondaires représente l'observation de la longueur du segment.

Les expérimentations effectuées dans [ESS 99] portent sur 88 noms de villes tunisiennes correspondant à un vocabulaire de 100 sous-mots (après élimination des redondances d'un total initial de 314 sous-mots). Le nombre d'états des modèles horizontaux varie de 2 à 10 et celui des super-états de 1 à 8. L'apprentissage est effectué séparément pour chaque modèle de sous-mot sur une base de 240 échantillons par sous-mot. Les 24000 images sont issues de documents imprimés de style Jiza (taille 12). Les tests effectués sur 33168 images de sous-mots (style Jiza, taille 12) ont abouti à un taux de reconnaissance de 99,84%.

Pour reconstituer les noms de villes, la méthode proposée dans [ESS 99] est basée sur un processus de vérification d'erreurs utilisant un dictionnaire des chaînes de sous-mots possibles. Le nombre de sous-mots dans l'image du nom de ville, la position du sous-mot considéré et la réponse du système de reconnaissance de sous-mots sont utilisés pour effectuer la recherche dans le dictionnaire. À chaque reconnaissance d'un sous-mot, les deux premiers candidats sont retenus, les chaînes de sous-mots construites, au fur et mesure, passent par une validation et sont pondérées par la somme des rangs de reconnaissance des sous-mots qui les constituent (rang=1 ou 2 puisque les deux premiers sous-mots candidats sont retenus). À la fin du traitement, si la réponse n'est pas unique, le nom de ville retenu correspond au poids le plus faible.

4. Système d'Amin 2000

Le système proposé en 2000 par Amin A. [AMI 00a] effectue la reconnaissance globale de mots arabes imprimés multi fontes. Une approche d'apprentissage symbolique est utilisée pour la création d'arbres de décision [COR 02] représentant les règles de classification des mots. L'outil choisi pour l'apprentissage est le C4.5 qui utilise les descriptions relatives aux mots en termes de caractéristiques ainsi que les classes correspondantes. Le système a été testé sur 1000 mots arabes imprimés avec différentes fontes à raison de 15 échantillons pour chaque mot (voir figure 2.6). Le taux moyen de reconnaissance atteint 92%.

5. Système de Benouareth 2000

Cette proposition s'intègre dans le cadre des travaux de recherche effectués au LRI de l'université d'Annaba.

Dans [BEN 00], Benouareth propose une méthode de reconnaissance de mots manuscrits qui combine une modélisation globale du mot, basée sur des caractéristiques perceptuelles avec une modélisation analytique utilisant des MMC correspondants aux lettres.

L'image prétraitée du mot à reconnaître passe d'abord par l'étape d'extraction de caractéristiques perceptuelles (le nombre de sous-mots, ascendants, descendants, boucles, points diacritiques et zigzags ou Hamzas). Le vecteur obtenu permettra l'identification du sous-ensemble de tous les mots du lexique ayant les mêmes caractéristiques perceptuelles. La sortie de cette procédure est une liste de mots candidats. Dans le cas où il n'y a qu'un seul candidat, le processus de reconnaissance se terminera et le résultat de la reconnaissance sera ce mot.

La segmentation en graphèmes est invoquée seulement lorsque la liste des mots hypothèses générée par le niveau global d'analyse est constituée de plus d'un mot. Cette opération découpe l'image de chaque composante connexe en graphèmes, qui sont les lettres potentielles (ou seulement des parties de celles-ci) formant les mots. Ce découpage se fait par extraction des points de segmentation potentiels déterminés par une analyse des contours de composantes connexes du mot.

La procédure d'extraction des primitives locales génère, pour chaque graphème dans l'image du mot, un vecteur de caractéristiques à partir duquel un code est affecté au graphème. Ce code est choisi parmi un ensemble de M codes possibles formant un code book représentant les différentes classes de graphèmes de l'écriture arabe. Il est construit par quantification vectorielle (en utilisant la variante LBG de l'algorithme de K-means) de l'ensemble des vecteurs de caractéristiques locales extraites à partir de tous les graphèmes générés à partir des images d'apprentissage. À l'issue de cette étape, une séquence de code correspondant à l'ensemble des classes de graphèmes formant le mot à reconnaître est obtenue.

Le classifieur markovien utilise la séquence de codes de graphèmes du mot en entrée, la liste des mots hypothèses, et un ensemble de modèles de références. Chaque modèle de référence associé à un mot du lexique est un MMC construit par concaténation des MMC lettres correspondant à la séquence des lettres du mot. Les modèles lettres sont obtenus par apprentissage. La classification consiste à mettre en correspondance (Matching) la séquence des codes de graphèmes obtenue à partir de l'image test, et l'ensemble des HMMs associés à

la liste des mots hypothèses générée par analyse globale. Cette mise en correspondance (ou alignement) est faite par l'algorithme de Viterbi. À la sortie de cette étape, on obtient le meilleur mot candidat.

6. Système d'Azizi et al. 2002

Cette proposition s'intègre dans le cadre des travaux de recherche effectués au LRI de l'université d'Annaba. Dans [AZI 02a], les auteurs proposent un système multi-classifieurs (dénommé MCNF) pour la reconnaissance de mots manuscrits en utilisant la combinaison parallèle de trois classifieurs neuronaux basés, chacun sur un type de caractéristiques.

En effet, pour la phase d'extraction de caractéristiques, Azizi et al. [AZI 02a, AZI 02b] optent pour un assortiment de caractéristiques :

Statistiques : leur extraction consiste à diviser l'image d'un mot en zones et à calculer la densité de pixels noirs dans chaque zone.

Perceptuelles : représentées par les ascendants, descendants, boucles et points diacritiques.

Géométriques : représentées par les moments invariants de HU [HEU 94, TRI 96].

Chaque classifieur neuronal est un perceptron à trois couches entraîné pour reconnaître les mots représentés par un ensemble homogène de caractéristiques. Dans la phase de décision, chaque mot en entrée sera représenté par des vecteurs de caractéristiques statistiques, perceptuelles et géométriques. Le système affecte chaque vecteur de caractéristiques au sous réseau correspondant et fournit des sorties de type mesure (indiquant à quel degré le classifieur considère chaque classe comme la vraie classe du mot en entrée).

Le résultat final sera la combinaison de ceux issus des différents sous réseaux neuronaux avec un calcul flou qui prend en compte les sorties et l'importance de chacun des réseaux en fonction des résultats obtenus dans la phase de test.

7. Système de Al Ohali 2002

Dans [ALO 02], Al Ohali propose un système de reconnaissance des sous-mots composant les mots du vocabulaire des montants de chèques en utilisant une combinaison séquentielle entre un classifieur global neuronal et un classifieur analytique markovien. Les sous-mots reconnus sont concaténés pour reconstituer les mots, et, par la suite, les montants littéraux.

Pour une image donnée d'un sous-mot en entrée, le premier classifieur, basé sur des caractéristiques globales, permet de réduire le nombre de sous-mots candidats et le second, basé sur des caractéristiques locales, détermine l'identité du sous-mot en question.

Dans la modélisation basée sur des caractéristiques globales, l'image d'un sous-mot est d'abord normalisée à une taille fixe, puis découpée en zones. La densité de pixels est calculée pour chaque zone, produisant ainsi un vecteur dont la taille sera réduite après une étape d'analyse en composantes principales pour produire des vecteurs non corrélés. Les vecteurs de caractéristiques globales sont présentés en entrée à un réseau de Kohonen [DAV 93, HER 94, JOD 94a, JOD 94b] à trois couches, dont la sortie correspond est un nombre, entre 1 et 20, correspondant au numéro du groupe gagnant de la compétition au niveau de la deuxième couche du réseau. Notons que les expérimentations effectuées, en utilisant les descriptions globales des sous-mots, ont permis de définir 20 groupes contenant, chacun, 24 classes en moyenne. Ainsi, la modélisation globale permet d'effectuer une réduction lexicale et fournit une liste de sous-mots candidats au classifieur markovien.

Dans la modélisation basée sur des caractéristiques locales, l'image du mot est soumise à une squelettisation, un graphe représentant le squelette du sous-mot est construit, puis transformé en arbre. La trajectoire de stylo (pen-trajectory) est ensuite estimée à partir de cet arbre. Chaque segment de la trajectoire curviligne est représenté par une séquence de codes de Freeman, puis une approximation linéaire est appliquée. Ainsi, le squelette du sous-mot est transformé en une suite de segments rectilignes, représentés, chacun, par deux paramètres: sa longueur et sa direction. Une quantification vectorielle est appliquée pour transformer chaque couple (longueur, direction) issu de la base d'apprentissage en une observation, la taille optimale du codebook a été fixée expérimentalement à 170. Un regroupement (clustering) a aussi été appliqué sur les séquences de caractéristiques pour partitionner chaque classe en groupes partageant les mêmes vecteurs. Le nombre de clusters obtenus est 150, représentant les 67 classes traitées.

Un modèle de Markov caché (MMC) gauche droite est utilisé pour modéliser chaque cluster. Le nombre d'états de chaque MMC est défini en fonction du nombre de lettres dans le sous-mot, tous les MMC correspondant à la même classe ont donc le même nombre d'états. Les modèles sont ensuite entraînés en utilisant l'algorithme de Baum-Welch.

Testé sur un lexique de 67 sous-mots, le classifieur neuronal effectue une réduction lexicale avec un taux de réussite de 99,04%. Sa combinaison avec le classifieur markovien a augmenté le taux de reconnaissance de ce dernier de 3,68% puisqu'il est passé de 69,85 à 73,53%.

Ce système de reconnaissance de sous-mots s'intègre dans un système de reconnaissance de montants littéraux de chèques. Les mots sont reconstruits par une concaténation de sous-mots consécutifs (4, au maximum, dans ce vocabulaire) et une procédure de recherche dans un dictionnaire. Comme certains sous-mots sont communs à plusieurs mots, la séquence de sous-mots n'est pas forcément unique. Une analyse syntaxique, basée sur la grammaire des montants, permet de vérifier la validité des propositions de montants obtenues.

8. Système de Kanoun et al., 2002

Les travaux de Kanoun et al. [KAN 02a, KAN 02b, ENN 04] ont porté essentiellement sur la validation de l'approche affixale proposée, pour la reconnaissance de mots arabes imprimés. Le prototype développé a permis de mettre en valeur l'intérêt de la procédure à la fois en terme de fiabilisation et d'amélioration du processus de reconnaissance, mais surtout, dans les perspectives que cela ouvre pour l'utilisation de grands lexiques et la préparation des textes reconnus à des traitements de plus haut niveau, en réalisant une annotation ou étiquetage morpho-syntaxique préalable des entités reconnues [ENN 04].

L'approche affixale procède par un processus dynamique de segmentation-reconnaissance d'affixes et du radical du mot (racine et infixes). Ce processus utilise des concepts linguistiques stables et liés à la structure morphologique du vocabulaire pour diriger la reconnaissance et filtrer les hypothèses non valides au cours du traitement. Le système proposé par S. Kanoun et al., commence par une segmentation explicite du mot en segments élémentaires. Une reconnaissance d'affixes (préfixe et suffixe) est alors effectuée en associant les premiers segments pour former des hypothèses de préfixes, et les derniers pour constituer des hypothèses de suffixes. Le résultat intermédiaire est un premier ensemble d'hypothèse de couples (préfixe, suffixe) sans aucune reconnaissance du radical. Cet ensemble est filtré par la restriction affixale au niveau des préfixes et des suffixes. Ce processus continue dans ce premier espace de solutions par la reconnaissance du radical pour chaque hypothèse cohérente de couples (préfixe, suffixe). Cette reconnaissance consiste à décomposer le radical en

hypothèses de racines et d'infices, qui sont validées ensuite en utilisant respectivement le dictionnaire de racines et le lexique d'infices. Un ensemble d'hypothèses de triplets (préfixe, infixe, suffixe) est alors constitué, sur lequel un deuxième niveau de filtrage de cohérence affixale est appliqué afin de ne garder que les triplets (préfixe, infixe, suffixe) cohérents. Enfin, la restriction sémantique entre ces derniers triplets et les hypothèses de racines est appliquée pour sélectionner à la fin des hypothèses de mots valides.

Ce premier prototype, utilisant des outils de reconnaissance relativement simples, a été testé sur un vocabulaire de 1000 mots issus d'un lexique de 200 racines. Les résultats préliminaires conduisent à un taux de bonne reconnaissance des mots de 74% lorsqu'une seule hypothèse de mot est retenue, et un taux de 87% de présence du bon mot dans une liste de 10 hypothèses (Top-1 et Top-10). Les situations d'erreur où le mot correct n'apparaît pas sont pour la plupart dues à des problèmes liés principalement à de mauvaises reconnaissances des affixes. Ce point est essentiel puisqu'il a permis de valider la démarche en mettant clairement l'accent sur le manque d'optimisation des outils de reconnaissance de base [ENN 04, KAN 02a].

9. Système de Khorsheed 2003

Dans le système proposé par Khorsheed [KHO 03], le mot manuscrit passe par une étape de squelettisation, puis le graphe du squelette est décomposé en une suite de segments après une étape d'approximation linéaire.

Chaque segment est caractérisé par sa longueur (distance euclidienne entre ses extrémités) et son angle d'orientation (entre le segment et l'axe horizontal). Une quantification vectorielle transforme la séquence de segments linéaires en une séquence de symboles discrets qui sont présentés à un classifieur markovien (MMC) pour produire en sortie une séquence de lettres. La modélisation markovienne proposée contient 296 états répartis de manière inégale entre les 32 modèles MMC gauche-droite de caractères, selon le nombre de segments linéaires que chaque lettre possède. Le modèle de la lettre "ض", par exemple, comprend 17 états alors que celui de la lettre "ة" en contient seulement 5.

Les échantillons de mots utilisés ont été extraits d'un manuscrit ancien, écrit pendant le règne des Khalifes Abbassides et intitulé "الكلبي لابن النسب جمهرة". La taille moyenne d'un caractère dans ce manuscrit est de 35 pixels en hauteur et 24 pixels en largeur. Après un total de 12960 tests, le taux de reconnaissance a atteint 72% sans intégrer aucune procédure de

10. Système de Pechwitz et al. 2003

Pechwitz et al. [PEC 03] proposent un système de reconnaissance de mots manuscrits basé sur une segmentation implicite et une modélisation markovienne des caractères. L'image du mot manuscrit passe d'abord par une étape de squelettisation, puis une normalisation horizontale pour obtenir une largeur moyenne constante de caractères et une normalisation verticale pour uniformiser la hauteur des ascendants et descendants. L'épaisseur du trait étant normalisée pendant la génération du squelette, une étape d'épaississement de trait est ensuite effectuée en utilisant un filtre Gaussien, produisant ainsi une image en niveaux de gris.

Une méthode par fenêtrage est utilisée pour effectuer une segmentation implicite du mot. Une fenêtre glissante de trois colonnes est appliquée sur l'image, de droite à gauche, pour générer un vecteur de caractéristiques correspondant à chaque bande (frame). Une transformation est ensuite effectuée sur les valeurs de niveaux de gris de chaque frame pour réduire le nombre de caractéristiques.

Pechwitz et al. génèrent un modèle MMC pour chaque forme de caractère (selon sa position dans un mot) et chaque ligature. La modélisation est effectuée par 160 MMC semi-continus à raison de 7 états par caractère (ou ligature). L'algorithme de Viterbi est utilisé pour l'apprentissage et la reconnaissance. Les modèles MMC de caractères sont concaténés pour constituer les modèles correspondants aux mots du lexique considéré.

Les expérimentations ont été effectuées en utilisant la base IFN/ENIT et les taux de reconnaissance de mots atteignent 89%.

11. Système de Al-Ma'adeed et al. 2004

Dans le travail décrit dans [ALM 04b], les auteurs proposent un système de reconnaissance de mots manuscrits en utilisant deux classifieurs combinés de manière séquentielle, le premier étant basé sur des règles et le second sur des MMC. Le premier classifieur sert à constituer des groupes de classes de mots en fonction de leurs caractéristiques perceptuelles globales (les auteurs définissent 8 groupes), alors que le second effectue la classification des mots au sein du groupe concerné.

Pour effectuer la reconnaissance d'un mot en entrée, le classifieur basé règles utilise des caractéristiques perceptuelles globales de ce mot, telles que le nombre de points hauts, de points bas et de segments, pour réduire le nombre de choix possibles (réduction du lexique) et orienter la recherche de la solution vers le groupe adéquat de modèles MMC. Ces derniers sont divisés en 8 groupes et se basent sur les caractéristiques locales utilisées dans [ALM 02b]. La construction des modèles dans un groupe, ainsi que le processus d'apprentissage, se passe d'une manière similaire à celle présentée dans la section précédente.

Près de 4200 images de mots arabes issues de la base AHDB et appartenant au vocabulaire des montants littéraux sont utilisées pendant les expérimentations. Les deux tiers ont servi à l'apprentissage et le reste a été réservé à la phase de test pendant laquelle le taux de reconnaissance avoisine les 60%.

12. Système de Kammoun et al. 2004

L'approche initiée dans [KAN 02a, KAN 02b] a permis de valider la démarche globale de recours aux règles linguistiques pour guider la reconnaissance. Le système proposé initialement permettait de piloter, à toutes les phases de traitement, l'espace d'hypothèses par filtrage linguistique, mais ne permettait de traiter que les mots décomposables du vocabulaire.

Dans [KAM 04], Kammoun et al. proposent une nouvelle architecture, plus simple dans sa conception puisque le filtrage linguistique n'est opéré qu'à la fin du processus de reconnaissance, mais également plus générale, puisqu'elle permet également de traiter le vocabulaire de mots non décomposables, de manière analytique uniquement, en cas d'échec de la reconnaissance affixale. Dans cette nouvelle architecture, l'accent est également mis sur les interactions avec l'utilisateur afin d'alimenter en ligne et de manière évolutive les dictionnaires et lexiques utilisés pour la reconnaissance permettant ainsi de fournir un moyen semi-automatique pour la constitution de grands lexiques.

Dans un objectif de validation plus significative de la démarche, une nouvelle base d'images de textes de près de 1400 mots a été constituée et les résultats obtenus sont très prometteurs puisqu'ils atteignent des taux allant de 81% de bonne reconnaissance des mots (Top-1) à plus de 99% de présence du bon mot dans la liste des 4 premières propositions (Top-4) [ENN 04, KAM 04].

13. Système de Farah et al. 2005

Cette proposition s'intègre dans le cadre des travaux de recherche effectués au LRI de l'université d'Annaba. Dans [FAR 05], les auteurs proposent de combiner diverses sources d'informations pour reconnaître les 48 mots manuscrits appartenant au vocabulaire des montants littéraux. L'étape de reconnaissance est effectuée par une combinaison parallèle de trois types de classifieurs (réseau neuronal de type perceptron multicouches, K plus proches voisins, K plus proches voisins flou) utilisant les caractéristiques globales perceptuelles des mots (nombre de sous-mots, ascendants, descendants boucles et points diacritiques). Le contexte grammatical des montants littéraux est utilisé pour prendre une décision finale sur les mots candidats obtenus.

Après l'extraction des caractéristiques à partir d'une image de mot présentée en entrée du système, chaque classifieur fournit en sortie une liste triée des trois meilleurs candidats avec les valeurs de confiance accordées, par le classifieur, à chacune des propositions. Pour chaque candidat, les valeurs de confiance sont normalisées, pour correspondre à des probabilités a posteriori des classes, puis sommées de sorte à établir une liste finale triée par ordre décroissant de valeurs de confiance. Le candidat en tête de liste sera la réponse du combineur.

Une base de 4800 images de mots, représentant les 48 mots du lexique écrits par 100 scripteurs différents, a été utilisée. 1200 ont servi à l'apprentissage des trois classifieurs et le reste a été réservé aux tests. Les taux de reconnaissance obtenus par la combinaison des classifieurs sont de l'ordre de 94%. Une étape de post-traitement syntaxique, basée sur la grammaire des montants littéraux, a permis d'améliorer les résultats de près de 2% par rapport à la combinaison et de 4% par rapport au taux moyen de reconnaissance individuelle des classifieurs.

14. Système de Sari et al. 2005

Cette proposition s'intègre dans le cadre des travaux de recherche effectués au LRI de l'université d'Annaba.

Dans ses travaux antérieurs, Sari propose un algorithme de segmentation explicite de mots manuscrits arabes en caractères [SAR 00, SAR 02] basé sur l'exploitation des caractéristiques topologiques de l'écriture arabe pour la construction d'une base de règles

morphologiques destinées à l'identification des points de segmentation idéaux. Pour cela, il procède à une extraction du contour de l'image du mot, puis applique des filtres topologiques pour rejeter ou valider les minima locaux du contour comme points de segmentation.

Les expérimentations récentes [SAR 05] relatives à cet algorithme de segmentation (ACSA) ont été menées sur une base de 10670 images de mots (près de 42617 caractères) écrits par différents scripteurs, 8200 de ces mots ont été extraits de la base IFN/ENIT et 1740 ont été collectées à partir d'enveloppes postales. La moitié de cette base a été utilisée pour raffiner les règles de segmentation et trouver le meilleur ordre possible pour leur application. L'autre moitié de la base a été utilisée pour tester ces règles et le taux de bonne segmentation atteint-les 86%.

Dans [SAR 05], un système de reconnaissance de mots manuscrits basé sur une segmentation en caractères est proposé en intégrant l'algorithme de segmentation ACSA avec un classifieur neuronal de caractères arabes manuscrits (RECAM. RECAM est un réseau neuronal de type perceptron à trois couches dont la couche d'entrée reçoit 139 caractéristiques extraites de l'image d'un caractère, ces caractéristiques représentent les projections (horizontale et verticale), les profils (gauche et droit) et les densités de pixels par zone (après découpage de l'image du caractère en zones). RECAM est divisé en 4 sous-réseaux, spécialisés, chacun dans la reconnaissance d'un sous-ensemble de caractères correspondant à une des 4 positions possibles (isolé, début, fin, milieu de sous-mot). RECAM a été entraîné avec 3080 caractères issus du processus de segmentation de mots avec ACSA, testé sur 3600 caractères, le taux de reconnaissance avoisine 84% [SAR 05].

Lorsqu'une image de mot manuscrit est présentée au système, elle est d'abord segmentée en caractères par ACSA, chaque caractère est reconnu en utilisant RECAM, puis le mot est reconstitué par concaténation des caractères reconnus.

15. Système de Benzenache 2007

Dans [BEN 07], Benzenache propose une approche globale pour la reconnaissance des mots arabes manuscrits dans un vocabulaire limité basé sur les modèles de Markov cachés de différents types : discrets (HMMs + k-means) et continus. Dans cette approche, chaque mot du lexique est modélisé par un HMM discret ou continu.

Le système effectue une série de prétraitement sur l'image acquise, tel que le filtrage, la binarisation, le lissage, la correction de l'inclinaison, squelettisation en plus de la localisation de la ligne de référence et l'extraction de la zone médiane. Ensuite, une segmentation en trames est appliquée à l'image. Benzenache a utilisé deux types de segmentation : la première est une segmentation uniforme avec une taille empirique de la trame fixée à 12 colonnes et une segmentation non uniforme basée sur l'analyse de l'histogramme de la projection verticale du mot.

Dans le but de maximiser le taux de reconnaissance, le système utilise une architecture hybride de caractéristiques. Les caractéristiques statistiques calculées à partir de l'histogramme et qui permettent une étude de la dispersion des points noirs sur l'image et les caractéristiques structurales basées sur l'extraction du nombre de boucles, les hampes, les jambes et le nombre de points.

Les vecteurs de caractéristiques sont soumis à deux classifieurs de type HMM pour identifier le mot. La discrimination de ce dernier est effectuée à base d'un critère de maximum de vraisemblance en utilisant l'algorithme de Viterbi.

Les tests ont été effectués sur une base de mots écrits par 100 scripteurs avec 3 occurrences pour les 48 wilayas c'est-à-dire une base de 14400 échantillons.

Avec un taux de réussite de l'ordre de 94.48 % et 92.91 % pour le mode multi-scripteurs et 88.12 % pour le mode indépendant du scripteur avec une segmentation non uniforme, obtenu par les HMMs continus. L'auteur estime que ces derniers sont nettement meilleurs par rapport aux HMMs discrets. Cela peut être interprété d'une part par la perte de décision qui représente un inconvénient majeur pour les HMMs discrets, et d'autre part, à la robustesse des distributions à variation continue par rapport à la variation discrète.

16. Système de Zaghdoudi 2008

Dans [ZAG 08], Zaghdoudi propose une approche globale pour la reconnaissance de mots manuscrits arabes basés sur les réseaux de neurones qui présentent d'après l'auteur une solution intéressante dans les problèmes de classification.

Le système effectue une série de prétraitement sur l'image du mot reçu en entrée du système en commençant par une binarisation, ensuite, lissage et bouchage, une opération de

normalisation, correction de l'inclinaison, amincissement et finalement la détermination de la zone médiane du mot.

Après une étape de segmentation basée sur les parties connexes de l'écriture, un module d'extraction de primitives transforme l'image en un vecteur de 28 primitives (16 caractéristiques structurelles et 12 caractéristiques statistiques) pour chaque composante connexe. Le vecteur final est le résultat d'une concaténation des vecteurs créés précédemment. L'auteur a utilisé les caractéristiques structurelles suivantes : nombre de boucles, nombre de hampes, nombre de jambes, nombre de points diacritiques en haut et en bas, nombre de points extrêmes et nombre de points de jonction. Les caractéristiques statistiques sont calculées à partir des histogrammes de projection horizontale, verticale et diagonale, en plus de la moyenne et la variance.

Pour étudier l'effet de la combinaison des caractéristiques, l'auteur a utilisé trois classifieurs de type PMC pour identifier le mot. Le premier reçoit en entrée un vecteur de 72 primitives qui présentent les 12 caractéristiques statistiques possibles pour chacune des six composantes connexes. La couche cachée est composée de 30 neurones. Le deuxième reçoit 96 primitives qui représentent les 16 caractéristiques structurelles possibles pour chacune des six composantes connexes. La couche cachée comporte 40 neurones. Le troisième PMC reçoit 168 primitives qui représentent la concaténation des deux types de caractéristiques utilisées.

Pour tester la fiabilité du système, l'auteur a utilisé une base de 24000 images de 50 scripteurs représentant les 48 wilayas de l'Algérie. Le système a atteint un taux de réussite de 79,65 % pour le premier PMC, 87,27 % pour le deuxième PMC et 91,23 % pour la combinaison des caractéristiques.

17. Conclusion

Nous avons donné dans ce chapitre un état de l'art sur les différents systèmes de reconnaissance de l'écriture. Chaque système a été créé dans une conception unique pour permettre à cette discipline de voir le jour dans des produits commercialisés. Malgré la variété des solutions proposées, le domaine reste dans un grand retard par rapport à d'autres langues. Ce retard est justifiée par l'absence et/ou de la non-disponibilité des bases de mots qui permettent de faire une bonne comparaison des systèmes proposés et ainsi localisés les vrais problèmes qui empêchent de réaliser des taux de réussites équivalentes aux autres langues.

CHAPITRE II TRAVAUX EXISTANTS

1. Introduction	34
2. Système de Miled et al. 1998	34
3. Système d'Essoukhri Ben Amara 1999	36
4. Système d'Amin 2000	37
5. Système de Benouareth 2000	37
6. Système d'Azizi et al. 2002	39
7. Système de Al Ohali 2002	39
8. Système de Kanoun et al., 2002	41
9. Système de Khorsheed 2003	42
10. Système de Pechwitz et al. 2003	43
11. Système de Al-Ma'adeed et al. 2004	43
12. Système de Kammoun et al. 2004	44
13. Système de Farah et al. 2005	45
14. Système de Sari et al. 2005	45
15. Système de Benzenache 2007	46
16. Système de Zaghdoudi 2008	47
17. Conclusion	48

Chapitre III

Les Réseaux de Neurones

1. Introduction

Depuis l'apparition de l'informatique les chercheurs ont tenté de reproduire un mécanisme qui est propre à l'homme, le raisonnement et au sens le plus large du terme, la pensée. Ainsi est née l'intelligence artificielle, dont les tendances de recherche se sont orientées rapidement vers l'approche symbolique. Cette représentation des connaissances est une formalisation du raisonnement, sous forme de règles, inspirées de la logique mathématique [SEB 96].

La théorie des réseaux de neurones est une approche s'appliquant principalement au monde de l'intelligence artificielle, venue renforcer l'impéritie de celle-ci dans certains domaines particuliers tels que l'analyse des images et de la parole, la classification, et la compréhension du langage naturel.

Les techniques neuronales sont par nature des méthodes de classification paramétriques puisque la phase d'apprentissage consiste à estimer les paramètres qui permettront de choisir la fonction de discrimination dans un ensemble de fonctions possibles.

Dans le courant de cette partie, nous aborderons la structure, le fonctionnement et l'apprentissage des réseaux neuronaux, pour cela, un éclaircissement de quelques notions biologiques relatives aux neurones sera essentiel.

2. Définition

Les réseaux de neurones artificiels sont un ensemble de neurones distribués dans une architecture parallèle et fortement connectés ou chaque neurone calcule une sortie en fonction de l'information qu'il a reçue en entrée [CLA 92].

3. Historique

Le fait d'utiliser des outils mathématiques ou de la physique pour modéliser le fonctionnement des systèmes nerveux ne date pas d'aujourd'hui.

- Dès les années 1940, une première école de pensée (fondée par Von Newman, Wiener, Turing et Mc Culloch) a essayé d'établir les fondements d'une science des systèmes capable d'auto- organisation.
- En 1943, Mc Culloch et Pitts ont modélisé la cellule nerveuse ou neurone par un automate à seuil et lui ont donné le nom de neurone formel [MAC et al 43].
- En 1961, Rosenblatt a décrit le premier modèle opérationnel de réseaux de neurones mettant en œuvre les idées de Hebb, Mc Culloch et Pitts. Il lui donna le nom de Perceptron. Ce modèle pouvait déjà reconnaître des formes simples.
- En 1969, deux mathématiciens, Minsky et Pappert ont démontré les sévères limitations du Perceptron. En effet, le Perceptron aidait à résoudre seulement les problèmes dits linéairement séparables. Les chercheurs et les investisseurs se sont donc désintéressés des réseaux de neurones pour rejoindre une nouvelle école de pensée animée par Simon, Minsky, Chomsky, et Mac Carthy. Ce mouvement est basé sur le calcul symbolique faisant l'hypothèse que le raisonnement modélisant la pensée est une combinaison de symboles soumise à des règles logiques. Cette approche symbolique utilisant l'ordinateur comme instrument eut un énorme succès et entraîna le développement de l'intelligence artificielle ce qui permet de dégager des concepts importants et de mettre au point les systèmes experts.

Néanmoins, cette approche symbolique comporte de nombreuses faiblesses, qui sont :

- La lenteur des méthodes séquentielles utilisées, parallélisme difficile ce qui peut donner lieu à une explosion combinatoire.
- Le problème de la représentation des connaissances.
- La faiblesse liée à l'apprentissage.

Les réseaux de neurones ont été délaissés pendant presque 20 ans, jusqu'au début des années 1980, avec l'approche proposée par Hopfield et l'algorithme de rétro-propagation dans le cadre des Perceptrons multicouches (PMC) proposés pour la première fois par Werbos, réinventé par la suite plusieurs fois avant d'être popularisé par Rumelhart et al en 1986.

Les différents problèmes liés à l'approche symbolique sont absents dans les réseaux de neurones. En effet, les réseaux de neurones travaillent essentiellement en parallèle et la représentation des connaissances est distribuée (résistance à la détérioration). Les réseaux,

renforcement des connexions entre les neurones mémorisant un fait, et le réseau s'auto-organise à partir des exemples qui lui sont présentés.

Les avantages de l'approche connexionniste, opposés à l'approche symbolique peuvent se synthétiser de la manière suivante :

- Activité parallèle,
- Représentation distribuée des connaissances,
- Apprentissage par modification des connexions.

4. Le neurone biologique

Le cerveau est composé de milliards de cellules nerveuses appelées neurones, le neurone est l'unité fonctionnelle de base du système nerveux [BOS 00]. Les neurones sont caractérisés par cinq fonctions de base qui sont :

- Recevoir des signaux en prévenance des neurones voisins
- Intégrer ces signaux
- Engendrer un influx nerveux
- Le conduire
- Le transmettre à un autre neurone capable de le recevoir

Structure des neurones :

Il y a quelques années, l'étude de la structure interne des cellules a révélé que tous les neurones étaient constitués indépendamment de leurs tailles et de leurs formes, d'un corps cellulaire (soma), des dendrites, d'un axone et d'une synapse [BOS 00].

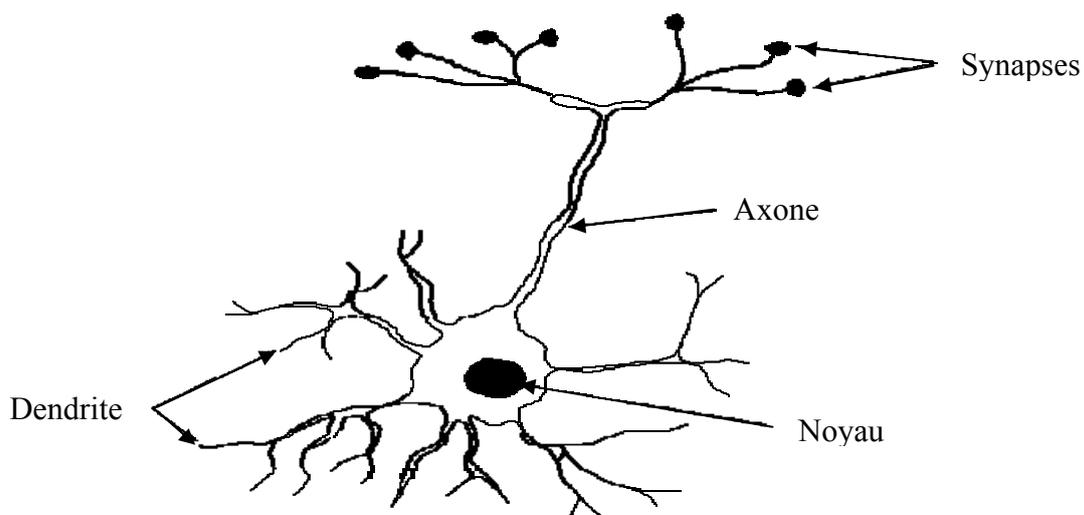


Figure III.1 : Le neurone biologique

5. Les neurones formels

En 1943, Warren McCulloch et Walter Pitts ont proposé un des premiers modèles de neurone artificiel, qui deviendra la base des réseaux de neurones artificiels [LIP 87].

S'inspirant de leurs travaux sur les neurones biologiques, ils ont proposé le modèle suivant :

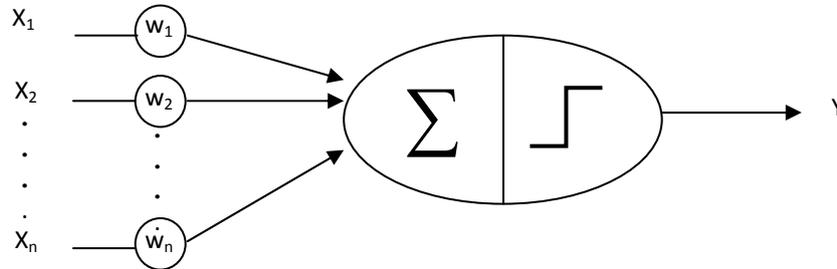


Figure III.2 : Le neurone formel

Un neurone formel fait une somme pondérée des potentiels d'actions qui lui parviennent (chacun de ces potentiels est une valeur numérique qui représente l'état du neurone qu'il a émis), puis s'active suivant la valeur de cette sommation pondérée. Si cette somme dépasse un certain seuil, le neurone est activé et transmet une réponse (sous forme de potentiel d'action) dont la valeur est celle de son activation. Si le neurone n'est pas activé, il ne transmet rien.

Notons :

- (X_i) les entrées des neurones formels ; avec : $i=1 \dots n$
- Y la sortie.
- β son seuil
- W_i les paramètres de pondération.
- f la fonction de seuillage.
- $\sum W_i X_i$ l'entrée nette.

$$f(x) = \begin{cases} 1 & \text{si } x > \beta \\ 0 & \text{si } x \leq \beta \end{cases} \quad (\text{III. 1})$$

Le fonctionnement du neurone formel est alors :

$$Y = f(\sum W_i X_i)$$

La fonction de seuillage peut prendre différentes formes, elle peut être soit binaire (Heaviside) soit linéaire à seuil ou multi seuil (sature), soit sigmoïde ou stochastique.

a) Une fonction binaire à seuil

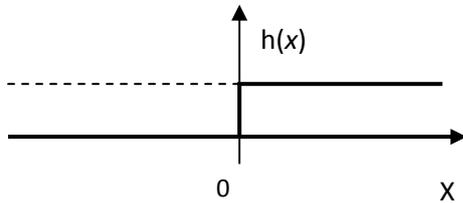


Figure III.3 : Fonction de Heaviside

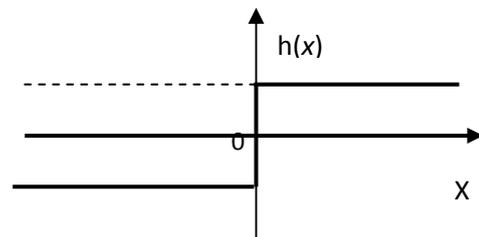


Figure III.4 : Fonction Signe

b) Fonction Linéaire à seuil ou à multi seuils:

$$Sature(X) = \begin{cases} X & \text{si } X \in [U, V] \\ U & \text{si } X \leq U \\ V & \text{si } X > V \end{cases}$$

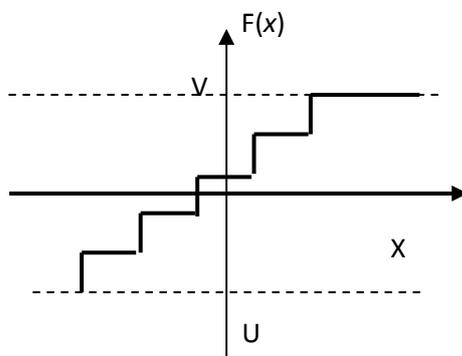


Figure III.5 : Fonction Linéaire multi seuils

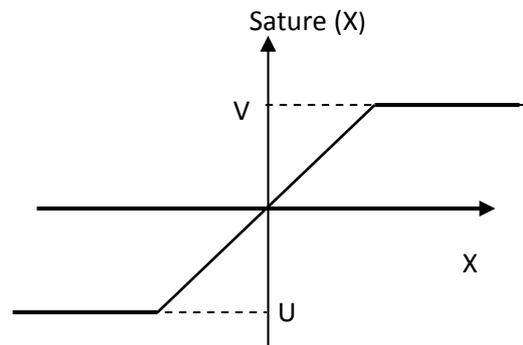
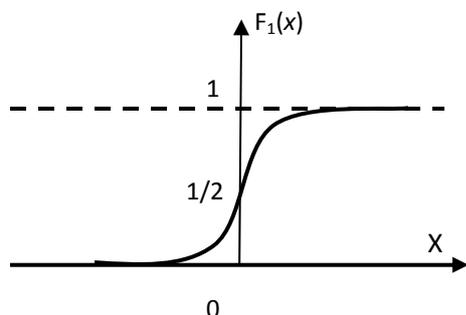


Figure III.6 : Fonction à Seuils

c) La Fonction sigmoïde :

$$F_1(X) = a(e^{kx} - 1) / e^{kx} + 1$$



$$F_2(X) = 1 / (e^{-x} + 1)$$

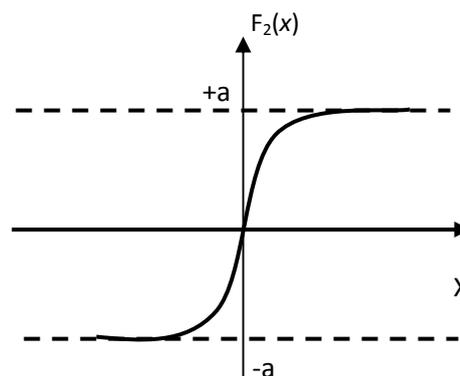


Figure III.7 : Fonctions sigmoïdes

Les neurones ont en effet deux fonctions d'activation, la première étant la production de l'entrée nette ($\sum W_i X_i$), la seconde étant la fonction d'activation. Dans le cas du neurone de McCulloch-Pitts, ainsi que dans la plupart des cas, les neurones sont des neurones "produit scalaire", car l'entrée nette est produite de la même manière que le produit scalaire entre le vecteur d'entrée et le vecteur de poids associé à ce neurone. Dans d'autres cas, on peut aussi utiliser des neurones "distance". Comme le nom l'indique, la production de l'entrée nette dans ce cas est produite par le calcul d'une distance entre le vecteur d'entrée et le vecteur de poids. Habituellement, une autre manière de calculer l'entrée nette amène à faire des choix différents par rapport à la fonction d'activation utilisée. Dans le cas des neurones "distances", les neurones sont utilisés d'une manière plus spécifique, particulièrement dans le cas des réseaux de Kohonen. Il est important de noter que les poids d'un réseau doivent être interprétés de manière différente selon le type de neurone utilisé.

La modélisation proposée offre une très grande analogie avec la réalité biologique, les interconnexions pour les axones et les dendrites, les poids pour les synapses, la fonction seuil pour approximer l'activité du soma. Cependant, ce modèle contient plusieurs hypothèses simplificatrices qui ne sont pas en accord avec la réalité biologique.

6. Architecture des réseaux de neurones

Les réseaux de neurones peuvent être vus comme des graphes orientés dans lesquels les neurones formels sont les sommets et les arcs pondérés sont les connexions entre sortie de neurones et entrée de neurones.

Un réseau de neurones est, comme son nom l'indique, un ensemble de neurones en réseau, de sorte que les signaux sortant (*outputs*) des neurones deviennent des signaux entrant (*inputs*) dans d'autres neurones.

D'une manière générale, l'architecture des réseaux de neurones formels peut aller d'une connectivité totale (tous les neurones sont reliés les uns aux autres), à une connectivité locale où les neurones sont reliés qu'à leurs plus proches voisins.

Il existe deux modèles classiques de réseaux : les réseaux à couches et les réseaux entièrement connectés.

6.1. Les réseaux à couches

Ce sont des réseaux structurés en couches (*layers*) successives telles que les neurones qui appartiennent à une même couche ne sont pas connectés entre eux, la première représentant la couche d'entrée (*input layer*), la dernière étant la couche de sortie (*output layer*), les couches intermédiaires étant les couches cachées (*hidden layers*) du réseau. Ces couches sont dites cachées, car de l'extérieur du réseau, on ne peut analyser clairement leur fonctionnement. On ne connaît vraiment que les signaux entrants et sortants du réseau.

Les neurones de la couche d'entrée ne sont pas vraiment des neurones traitants (*computing neurons*), mais ont pour seule utilité de normaliser l'entrée des signaux ainsi que la distribution des signaux entrants.

Dans cette architecture normalisée, les couches de neurones sont totalement interconnectées, c'est-à-dire que les neurones d'une couche sont tous reliés à tous les neurones des couches adjacentes, chacune des couches recevant des signaux de la couche précédente et transmettant le résultat de ses traitements à la couche suivante.

Cette architecture normalisée peut sembler rigide, mais elle permet une représentation correcte de la plupart des réseaux de neurones, tout en permettant l'utilisation d'algorithmes d'entraînement plus généraux.

Les données entrées peuvent être de n'importe quel type représentable de manière binaire ou numérique. Ces données peuvent également être vues comme des vecteurs, et le réseau de neurones une application vectorielle.

Les réseaux de neurones peuvent être soit statiques, soit dynamiques. Les réseaux dynamiques ont la capacité de s'actualiser (modifier les poids des liaisons) en tout temps. Les réseaux statiques, leur poids sont choisis (et non pas modifiés par un entraînement).

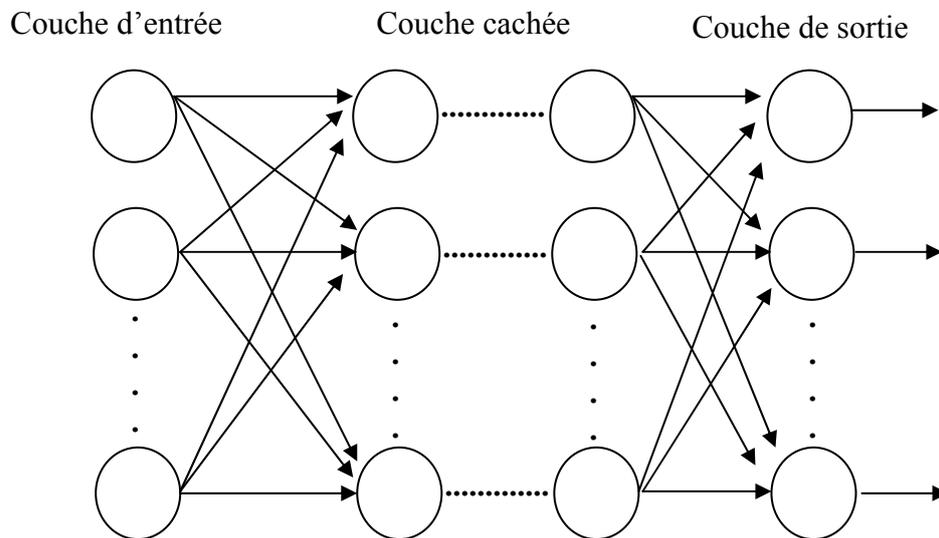


Figure III.8 : Réseau à couches

6.2. Les réseaux entièrement connectés

Dans ce type de réseaux, chaque neurone est totalement connecté aux autres neurones quelque soit la couche à la quel il appartient. Un neurone reçoit des impulsions de tous ses compagnons et même de lui-même. Puisque tous les éléments sont connectés entre eux, il n'y a en général qu'une seule couche vue et au mieux deux ; une d'entrée et une de sortie [SAE 96].

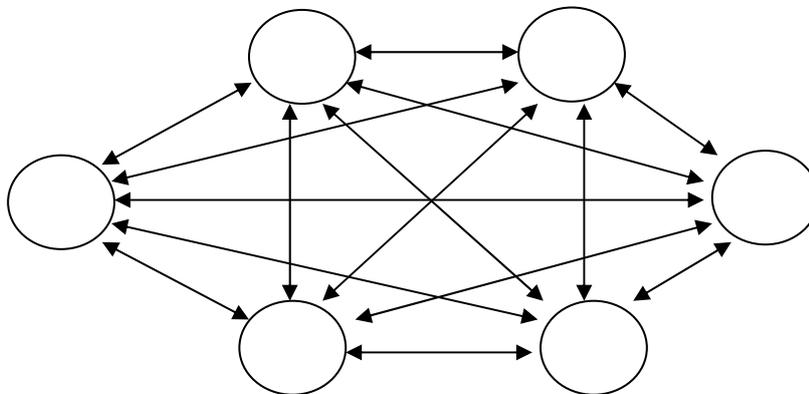


Figure III.9 : Réseaux entièrement connectés

7. L'apprentissage

L'apprentissage (selon Larousse) : "ensemble de méthodes permettant d'établir chez les êtres vivants des connexions entre certains stimulus et certaines réponses".

Les systèmes neuronaux biologiques ne sont pas nés programmés avec toutes les connaissances et les capacités qu'ils vont éventuellement avoir.

Un processus d'apprentissage qui y prend place modifie, avec le temps, le réseau pour y incorporer de nouvelles informations. L'une des plus intéressantes caractéristiques des réseaux de neurones est leur habilité d'apprendre [DRE et al 02].

Un réseau de neurones artificiel est entraîné à réaliser une application donnée à partir d'une base d'exemple, selon une règle d'apprentissage. Si après un certain niveau d'apprentissage, la fonction réalisée par le réseau est proche de l'application souhaitée, on dit alors que celui-ci a une bonne faculté de généralisation. L'idéal est que les exemples appris soient reconnus à 100 %.

Les algorithmes d'apprentissages sont divisés en deux catégories : apprentissage supervisé et apprentissage non supervisé.

7.1. Apprentissage supervisé

Également connu par l'appellation "**Apprentissage avec professeur**" : les sorties désirées sont fournies par le professeur pour tous les exemples que l'on veut faire apprendre au réseau. Il exige une partie de chaque vecteur d'entrée dont l'un est le vecteur de sortie désirée. L'ensemble est appelé paire d'apprentissages (vecteur d'entrée, vecteur de sortie désirée).

La procédure d'apprentissage consiste à appliquer le vecteur d'entrée, calculer la sortie et la comparer au vecteur de sortie désirée. La différence (l'erreur) est réappliquée au réseau en modifiant les poids suivant un algorithme qui tend à minimiser l'erreur. Généralement un bon apprentissage nécessite plusieurs passages sur le jeu complet d'exemples.

À la n ème itération, un ajustement $W_{ij}(n)$ est appliqué au poids synaptique $W_{ij}(n)$ produisant la nouvelle valeur $W_{ij}(n+1)$. On peut donc écrire :

$$W_{ij}(n+1) = W_{ij}(n) + \Delta W_{ij}(n) \quad \text{(III.2)}$$

Les poids synaptiques $W_{ij}(n)$ et $W_{ij}(n+1)$ représentent respectivement l'ancienne et la nouvelle valeur du poids synaptique W_{ij} .

Ces poids évoluent alors jusqu'à ce que la sortie Y devienne en moyenne égale à la sortie désirée fournie par le professeur.

7.2. Apprentissage non supervisé

Également connu par l'appellation "**Apprentissage sans professeur**". Dans ce type d'apprentissage, aucune information extérieure n'est disponible. L'absence de jugement extérieure entraîne un traitement particulier des données. Suivant l'algorithme, les exemples traités sont utilisés comme modèle vers lequel le réseau devra converger pendant l'apprentissage. L'algorithme d'apprentissage modifie les poids du réseau pour produire un vecteur de sortie désirée.

8. Les règles d'apprentissage

Dans les mécanismes d'apprentissage, on trouve les règles relatives aux modèles développés par des chercheurs. Historiquement, la première règle d'apprentissage a été formulée de façon qualitative par **Donald HEBB** en 1949. Grossièrement, il s'agissait de renforcer la connexion reliant deux neurones à chaque fois qu'ils étaient actifs simultanément.

L'autre modèle de règle d'apprentissage est basé sur l'idée de réduire progressivement la différence entre la sortie obtenue et la sortie désirée. C'est la règle de Perceptron, ou la règle de **Widrow-Hoff**, on appelle également règle **Delta généralisée**.

8.1. La règle de HEBB

Cette règle est le premier mécanisme d'évolution proposé pour les synapses. Son interprétation pour les réseaux de neurones formels est la suivante :

Si deux neurones sont connectés entre eux et activés en même temps, la connexion qui les relie doit être renforcée, dans le cas contraire, elle n'est pas modifiée.

En d'autres termes, si à l'instant $(t+dt)$ on soumet le réseau à un stimulus (excitation) pour provoquer l'activation A_i et A_j du neurone i et j respectivement et supposant de plus que ces activations sont booléennes, il y'aura un ajustement de poids qui les relie. L'ajustement des poids sera calculé en appliquant la formule suivante :

$$\Delta W_{ij}(n) = b A_i A_j \quad (\text{III.3})$$

Où b ($b > 0$) est un paramètre de l'intensité de l'apprentissage.

En suite en appliquant la formule suivante :

$$W_{ij}(n+1) = W_{ij}(n) + \Delta W_{ij}(n) \quad (\text{III.4})$$

Avec:

$$W_{ij}(n) = 1/N \sum A_i A_j$$

Où N : représente le nombre total des neurones

A_j : représente l'état du neurone j lorsque l'échantillon k est présenté pour l'apprentissage.

8.2. Règle de Windrow-Hoff

La règle d'apprentissage à correction d'erreur (Windrow-Hoff) appelée aussi « la règle delta généralisée » a été établie en 1960 par Windrow et Hoff. Cette règle indique que la modification des poids synaptiques est proportionnelle à l'erreur entre le résultat souhaité et le résultat réel et les valeurs des entrées [MEN 01].

9. Les modèles des réseaux de neurones

Lorsqu'on aborde le domaine des réseaux de neurones, on regroupe un certain nombre de modèles pour décrire un modèle, il convient d'indiquer ses caractéristiques :

- Une architecture : soit monocouche ou multicouches.
- Une règle de propagation et d'activation.
- Une règle d'apprentissage.

9.1. Adaline de Windrow–Hoff

Inventé en 1960 par Bernard Windrow et Ted Hoff. Ce modèle possède deux couches, une couche d'entrée et une couche de sortie.

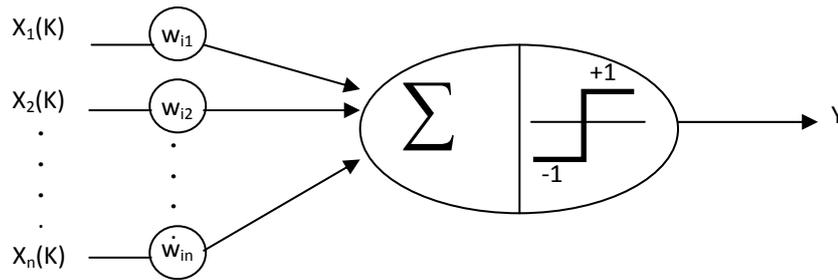


Figure III.10 : Le neurone d'Adaline

Les neurones de la première couche envoient directement leurs entrées X_j vers la couche de sortie sans modifier le signal [JOD 94 b].

Chaque neurone de la couche de sortie effectue la somme pondérée de ses entrées et compare le résultat Y avec un seuil θ . On aura alors :

$$Y_i(k) = \text{SEUIL}(\sum W_{ij}X_j) \quad (\text{III.5})$$

Les poids doivent être ajustés de telle façon que les sorties calculées par le réseau soient proches des sorties désirées. L'ajustement des poids sera effectué par la loi suivante :

$$\Delta W_{ij} = \alpha (D_i - Y_i) X_j \quad (\text{III.6})$$

Où X_j la $j^{\text{ième}}$ entrée du neurone Adaline d'indice i dans le réseau.

W_{ij} poids de la connexion (neurone j vers neurone i).

D_i sortie désirée du neurone i .

α coefficient d'apprentissage.

Y_i sortie calculée par le réseau du neurone i .

9.2. Le Perceptron de ROSENBLATT

Introduit par Frank Rosenblatt en 1958, le Perceptron est organisé en couches

- La première couche appelée « *rétine* » : elle est constituée de cellules d'entrée, recueille les signaux d'entrée.
- La deuxième couche appelée « *couche d'association* » : les connexions entre cette couche et la couche précédente sont pondérées par des poids qui sont fixés et qui n'évoluent pas pendant l'apprentissage .

- La troisième couche appelée « *couche de décision* » : contiens les neurones généralement logiques et à fonction d'activation à seuil

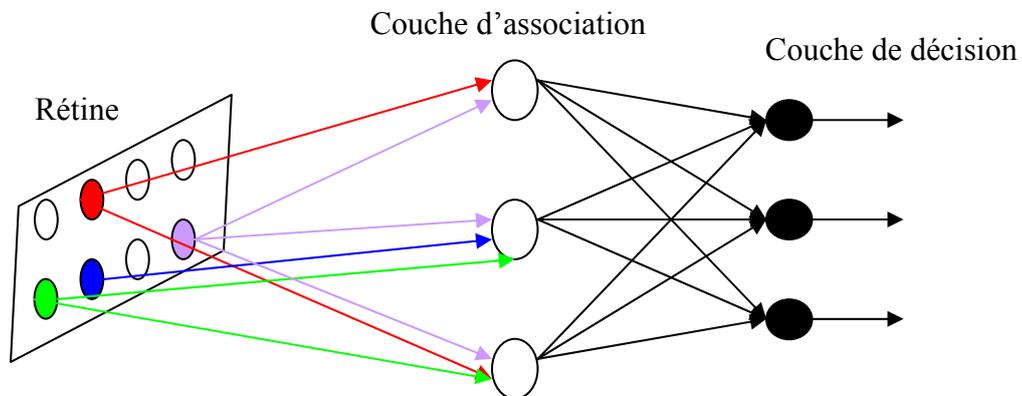


Figure III.11 : Représentation du perceptron

9.3. Le Perceptron multicouche

Un réseau multicouche est une extension du modèle monocouche issu du Perceptron, avec une ou plusieurs couches cachées successives entre la couche d'entrée et la couche de sortie. La connectivité la plus courante consiste à définir les connexions uniquement d'une couche vers la couche suivante.

Il possède les caractéristiques suivantes :

- Chaque neurone est connecté en entrée à tous les neurones de la couche précédente.
- La sortie d'un neurone est connectée à tous les neurones de la couche suivante.
- Il n'y a pas de boucle.
- Le nombre de couches cachées ainsi que le nombre de neurones par couche peut être choisi en fonction de l'application.

L'architecture précédemment décrite est l'une des plus utilisées, sinon la plus utilisée actuellement et cela grâce à l'apparition d'un algorithme d'apprentissage bien approprié, il s'agit du rétro propagation du gradient.

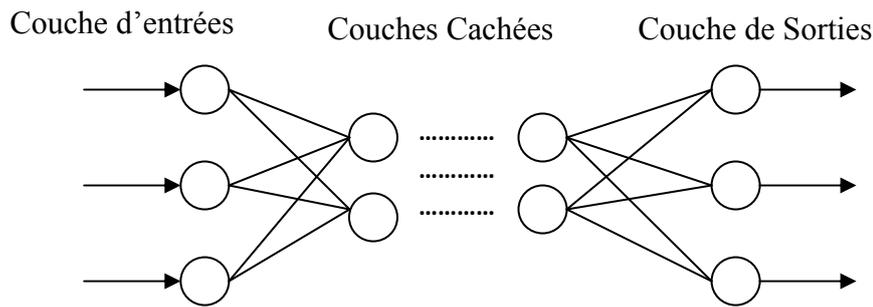


Figure III.12 : Perceptron Multicouches

9.4. Le modèle de HOPFIELD

Ce modèle est un réseau totalement connecté dont les valeurs d'entrée et de sorties appartiennent à l'intervalle $(-1,1)$. Les poids sont symétriques ($W_{ij} = W_{ji}$). La symétrie des connexions assure la convergence du réseau.

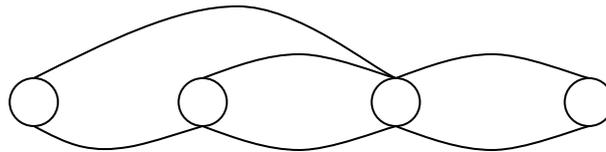


Figure III.13 : Le réseau de HOPFIELD

L'apprentissage dans un réseau de HOPFIELD est parfois qualifié d'apprentissage par cœur, par opposition avec les algorithmes d'apprentissage par essai et erreur.

Au début de l'application, les poids des connexions sont initialisés à des valeurs qui dépendent de l'application et chaque neurone est affecté d'un état.

Durant chaque itération une seule unité de traitement (neurone) est mise à jour.

Son état pourra ainsi basculer éventuellement d'une valeur à une autre. Ce processus de traitement sera répété jusqu'à ce qu'aucun neurone ne subisse de modification d'état.

9.5. Le modèle de KOHONEN

Également connu sous l'appellation de " Carte Auto – organisatrice de KOHONEN ". Ce type de réseau de neurones consiste en une situation où plusieurs neurones entrent en compétition pour être activés. Dans cette situation, un seul neurone est activé (*winner-takes-all neuron*) suite à la compétition.

Le réseau combine une couche d'entrée avec une couche compétitive de neurones entraînés de façon non supervisée (la sortie désirée n'est pas définie).

La carte de KOHONEN illustrée est un réseau à deux couches , la première couche du réseau représente la couche d'entrée , la seconde couche – compétitive – se présente comme une grille à deux dimensions. Toutes les interconnexions vont de la couche d'entrée vers la couche de sortie. Chaque neurone de la couche d'entrée est connecté à tous les neurones de la couche compétitive.

Il est caractérisé par le fait qu'il opère par voisinage durant son apprentissage ,c'est-à-dire que les connexions du neurone le plus proche de la forme présentée sont renforcées. La sortie Y_k d'un neurone d'indice k est définie comme étant la distance euclidienne entre la forme X présentée et le vecteur des poids W_k .

Ces réseaux ont été étudiés par Grossberg, von der Malsburg, Fukushima, mais c'est Tuevo Kohonen qui a vraiment lancé ce domaine en introduisant une nouvelle classe de réseau de neurones, soit les « cartes topologiques d'auto-organisation » (*self-organizing feature map*).

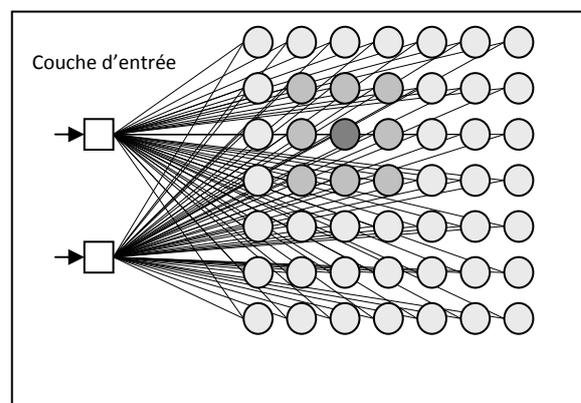


Figure III.14 : Carte de Kohonen avec deux entrées

10. Propriétés des réseaux de neurones

La notion de parallélisme se situe à la base de l'architecture des réseaux de neurones considérés comme un ensemble d'entités élémentaires qui travaillent simultanément. L'intérêt d'une telle conception du traitement des données a été mis en évidence par l'échec des méthodes séquentielles pour traiter des problèmes qui nécessitent des quantités énormes de données [BOU 07]. Le parallélisme permet une rapidité de calcul supérieur, mais exige de penser et de poser différemment les problèmes à résoudre.

La capacité d'adaptation se manifeste tout d'abord dans les réseaux de neurones par la capacité d'apprentissage qui permet au réseau de tenir compte de nouvelles contraintes ou de nouvelles données du monde extérieur. De plus, elle se caractérise dans certains réseaux par leur capacité d'auto organisation qui assure leur stabilité en tant que systèmes dynamiques.

Dans les réseaux de neurones, la mémoire d'un fait correspond à une carte d'activation des neurones. Cette carte est en quelque sorte un codage du fait mémorisé. L'intérêt de cette distribution de la mémoire sur plusieurs entités est la résistance au bruit.

La capacité de généralisation se manifeste dans le fait que les réseaux de neurones peuvent à partir d'ensembles d'exemples, apprendre à retrouver les règles sous-jacentes, ou à mimer les comportements qui permettent de résoudre le problème.

11. Conclusion

Nous avons essayé à travers ce chapitre de donner un aperçu général sur les réseaux de neurones artificiels et les différentes règles d'apprentissage sans rentrer trop dans l'origine des formules mathématiques et les preuves de ces derniers. En effet nous abordons les réseaux de neurones dans la seule perspective de faire de la reconnaissance des formes et de la classification. Notre objectif n'est donc pas celui des « neurosciences » consistant à chercher à utiliser la puissance des ordinateurs pour simuler l'intelligence humaine ou animale.

Les paramètres du réseau sont déterminés statistiquement lors d'une phase d'apprentissage à partir d'une base de données d'exemples, appelée base d'apprentissage. En pratique, l'utilisation de la méthode neuronale pose certaines difficultés telles que l'optimisation de la phase d'apprentissage, le choix de l'architecture adéquate et la détermination du « pas d'apprentissage ».

Dans les chapitres suivants, nous allons utiliser le Perceptron multicouches vu sa simplicité d'implémentation, sa robustesse ainsi que son pouvoir de généralisation à partir un ensemble réduit d'échantillons d'apprentissages.

CHAPITRE III LES RÉSEAUX DE NEURONES

1. Introduction	49
2. Définition	49
3. Historique	49
4. Le neurone biologique	51
5. Les neurones formels	52
6. Architecture des réseaux de neurones	54
6.1. Les réseaux à couches	55
6.2. Les réseaux entièrement connectés	56
7. L'apprentissage	57
7.1. Apprentissage supervisé	57
7.2. Apprentissage non supervisé	58
8. Les règles d'apprentissage	58
8.1. La règle de HEBB	58
8.2. Règle de Windrow-Hoff	59
9. Les modèles des réseaux de neurones	59
9.1. Adaline de Windrow-Hoff	59
9.2. Le perceptron de ROSENBLATT	60
9.3. Le perceptron multicouche	61
9.4. Le modèle de HOPFIELD	62
9.5. Le modèle de KOHONEN	63
10. Propriétés des réseaux de neurones	64
11. Conclusion	64

Chapitre IV

Les modèles de Markov cachés

1. Introduction

Les Modèles de Markov Caché connaissent actuellement un essor important, grâce à leur capacité d'intégration du contexte et d'absorption du bruit [AVI 96] [BEL 97]. Les différents travaux réalisés reposent dans une grande part sur l'expérience accumulée dans le domaine de la reconnaissance de la parole où les HMMs sont fréquemment utilisés [AGA 93]. Tout comme la parole, l'écriture se prête à une modélisation stochastique à tous les niveaux de reconnaissance. En effet, les lettres sont régies par un contexte lexical des mots de la langue, se traduisant par des probabilités d'apparition et de succession dans des mots.

Dans ce chapitre, nous allons d'abord voir la théorie de base des HMMs en tant qu'outils statistiques. Ensuite, expliquer les HMMs à temps discrets, qui sont l'objet de notre étude. Nous allons donner quelques topologies en détaillant chacune d'elles et nous discutons les différentes formules mathématiques qui auront besoin d'être implémentées.

2. Définitions préliminaires

Une variable aléatoire réelle est une fonction mesurable à valeur dans \mathbb{R} , définie sur un espace de probabilités Ω :

$$X : \Omega \rightarrow \mathbb{R}$$

$$W \rightarrow X(w)$$

Une variable aléatoire discrète est une variable aléatoire à valeurs dans un ensemble fini ou dénombrable de réalisations (x_1, x_2, \dots, x_k) . On notera $P(X=x_k)$ la probabilité, au sens générique, de la réalisation x_k de la variable aléatoire X [ELY 96].

Une variable aléatoire continue X est une variable à valeurs dans \mathbb{R} . la fonction de densité de probabilité (FDP) notée $f_x(x)$ associée à X est définie comme étant la probabilité que X prend la valeur $x \in \mathbb{R}$ [FEH 99].

C'est une fonction partout positive qui satisfait la condition : $\int_R f_x(X) dX = 1$.

Soient X, Y, Z trois variables aléatoires, X et Y sont indépendantes relativement (indépendance conditionnelle) à Z , si et seulement si on a :

$$P(X.Y / Z) = P(X / Z) P(Y / Z) \quad (\text{IV.1a})$$

Ce qui est équivalent à avoir :

$$P(X / Y, Z) = P(X / Z) \text{ et } P(Y / X, Z) = P(Y / Z) \quad (\text{IV.1b})$$

Un processus stochastique $\{X(t); t \in T\}$ est une suite de variables aléatoires où t est un paramètre réel [FEH 99]. L'espace des paramètres où espace du temps T peut être discret : $T = \{0, 1, 2, \dots\}$ ou continu : $T = [0, \infty[$. Nous parlons alors de processus stochastiques à temps discret ou continu. L'espace des états S est l'ensemble dénombrable des valeurs prises par l'ensemble des variables aléatoires du processus stochastique. Dans le cas où l'espace des états est discret : $S = \{s_0, s_1, \dots, s_{N-1}\}$, nous parlons de processus stochastique à état discret.

L'évolution d'un processus stochastique est une suite de transitions d'états : $s_0 s_1 \dots s_T$, où dans cette notation s_0 correspond à l'état du processus à l'instant 0. Sa loi d'évolution est obtenue à l'aide de la probabilité $\Pr(s_0 \dots s_T)$ [LAH et al 97]. Cette dernière est définie de proche de la manière suivante :

$$\Pr(s_0 \dots s_T) = \Pr(s_0 \dots s_{T-1}) \times \Pr(s_T / s_0 \dots s_{T-1}) \quad (\text{IV.1c})$$

$$\Pr(s_0) \times \Pr(s_1 / s_0) \times \Pr(s_2 / s_0 s_1) \times \dots \times \Pr(s_T / s_0 \dots s_{T-1}) \quad (\text{IV.1d})$$

La caractérisation du processus se résume donc à l'obtention des probabilités initiales $\Pr(s_0)$ et des probabilités des états conditionnés par les évolutions antérieures. La situation générale est celle où la loi de probabilité des états, à certains instants, dépend de la totalité de l'histoire du processus. On dira que celui-ci garde la mémoire de son passé.

3. Le modèle de Markov à temps discret

Un processus de Markov discret est un automate stochastique à nombre d'états finis N . C'est un processus aléatoire qui se déplace d'état en état à chaque instant t , et il génère une séquence d'observations aléatoire $q_1, q_2, q_3, \dots, q_t$ où q_t (l'indice t représente le temps)

prennent leurs valeurs parmi l'ensemble des N états du modèle $\{s_0, s_1, \dots, s_N\}$ [MIL 97 et al] [CHE et al 94].

On simplifie le modèle pour des raisons pratiques en supposant que le processus markovien est de premier ordre et stationnaire [RAB 89].

3.1. Propriétés de Markov du premier ordre

Un processus stochastique vérifie la propriété de Markov du premier ordre, si la probabilité de passer de l'état j à l'instant t ne dépend que de l'état précédent i à l'instant $t-1$:

$$\Pr(q_t = s_j | q_{t-1} = s_i, q_{t-2} = s_k, \dots) = \Pr(q_t = s_j | q_{t-1} = s_i) \quad (\text{IV.2a})$$

L'équation précédente peut alors être réécrite de la manière suivante :

$$\Pr(s_0 \dots s_T) = \Pr(s_0) \times \Pr(s_1 / s_0) \times \Pr(s_2 / s_1) \times \dots \times \Pr(s_T / s_{T-1}) \quad (\text{IV.2b})$$

Un processus ainsi défini est appelé processus de Markov du premier ordre. On dira parfois aussi qu'un tel processus stochastique est sans mémoire. Il est dit stationnaire si pour tout instant t et tout décalage temporel k :

$$\Pr(q_t = s_j / q_{t-1} = s_i) = \Pr(q_{t+k} = s_j / q_{t+k-1} = s_i) \quad (\text{IV.2c})$$

3.2. Paramètres d'un modèle de Markov discret

Un modèle de Markov discret est caractérisé par :

- N : nombre d'états dans le modèle.
- $S = \{s_i\}$: ensemble d'états du modèle. q_t : Représentant l'état du système à l'instant t .
- A : matrice de probabilité de transition entre les états de la chaîne.

$A = [a_{ij}]$ matrice des probabilités de transition $N \times N$ où a_{ij} désigne la probabilité que le modèle passe de l'état s_i vers l'état s_j elle est décrite dans le cas général par :

$$a_{ij} = p(q_t = s_j | q_{t-1} = s_i) \quad 0 \leq i, j \leq N-1, \quad \forall t \in [2 \dots T] \quad (\text{IV.3})$$

De plus : $a_{ij} \geq 0 \quad \forall i, j, \quad \sum_{j=1}^N a_{ij} = 1$

- $\pi = \{\pi_i\}$: vecteur de densités de probabilités initiales.

$$\pi_i = p(q_1 = s_i), \forall i, j \in [1 \dots N], 0 \leq \pi_i \leq 1 \text{ et } \sum_{i=1}^N \pi_i = 1 \quad (\text{IV.4})$$

π_i : représente la probabilité que l'état de départ du modèle est l'état i .

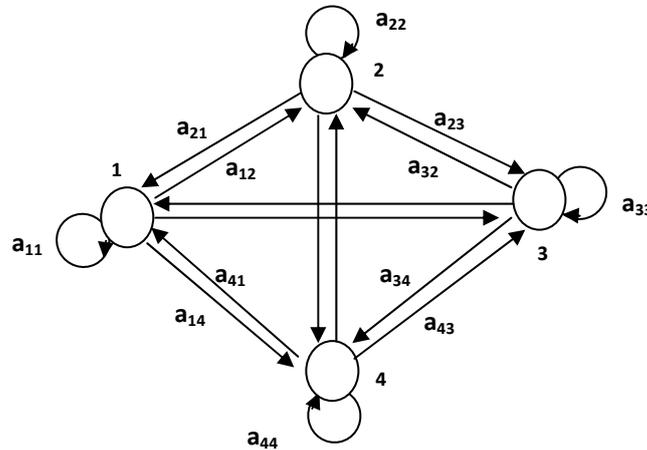


Figure IV. 1 : Diagramme d'états d'un processus de Markov discret

Ce processus stochastique peut être appelé *modèle de Markov observable*, puisque chaque état correspond à un état physique *observable* du système à chaque instant t .

4. Le modèle de Markov caché discret (DHMM)

Dans la partie précédente, on considérait les modèles de Markov dont chaque état correspond à un événement observable (physique). Ceci est très restrictif si l'on veut modéliser des problèmes plus complexes [BEN 99]. Le concept de modèle de Markov est donc étendu pour que l'observation soit une fonction de probabilité d'état [RAB 89]. Le modèle est un processus aléatoire qui n'est pas observable, mais qui peut être étudié à travers un autre ensemble de processus aléatoires qui produisent une série d'observations [LEM et al 96]. Ces deux niveaux donnent à l'approche markovienne une flexibilité qui est séduisante pour modéliser des phénomènes aussi complexes que la production de la parole, le traitement d'image, le traitement des séries temporelles non stationnaires, le contrôle de qualité, etc.

4.1. Définitions

Un modèle de Markov est un automate probabiliste à nombre d'états finis constitué de N états. Un processus aléatoire se déplace d'état en état à chaque instant, et on note q_t le

numéro de l'état atteint par le processus à l'instant t . L'état réel q_t du processus n'est pas directement observable, il est caché, mais peut être observé par un autre processus aléatoire qui émet après chaque changement d'état un symbole o_t . Dans le cas d'un processus markovien d'ordre 1, la probabilité de passer de l'état i à l'état j à l'instant t et d'émettre o_t ne dépend ni du temps, ni des états aux instants précédents.

Donc, un modèle de Markov caché (HMM) est représenté de la même façon qu'une chaîne de Markov, il s'agit d'un processus doublement stochastique, dans lequel les séquences d'observations dont l'état de chaque observation n'est pas observé, mais associées à une fonction de densité de probabilité [BEN 99] [RAB 89].

4.2. Paramètres d'un modèle de Markov caché discret

Un modèle de Markov caché (HMM) : $\lambda = (A, B, \pi)$ est alors défini par :

- N : nombre d'états dans le modèle.
- $S = \{s_i\} \ 1 \leq i \leq N$ l'ensemble des N états, en sachant que le processus part de l'état s_1 à l'instant $t=0$ et arrive à l'état final s_N à l'instant $t=T$.
- M : nombre des différents symboles observables par état (taille de l'alphabet).
- $V = \{v_k\}$ l'ensemble des M symboles observables dans un état ($N \gg M$), $k = 1, 2, \dots, M$, étant l'alphabet de symboles observables.
- $A = \{a_{ij}\} \ 1 \leq i \leq N \text{ et } 1 \leq j \leq N$ la matrice des probabilités de transition entre les états de la chaîne :

$$a_{ij} = p(q_t = s_j | q_{t-1} = s_i) \text{ avec } a_{ij} \geq 0 \ \forall i, j \text{ et } \sum_{j=1}^N a_{ij} = 1 \quad (\text{IV.5})$$

a_{ij} : représente la probabilité que le modèle évolue de l'état i vers l'état j .

- $B = \{b_j(k)\} \ 1 \leq j \leq N \text{ et } 1 \leq k \leq M$ l'ensemble des probabilités d'émission des symboles v_k dans chaque état j du modèle avec :

$$b_j(k) = p(o_t = v_k | q_t = s_j) \ 1 \leq j \leq N, 1 \leq k \leq M \quad (\text{IV.6})$$

$$b_j(k) = 0 \ \forall j, k \text{ et } \sum_{k=1}^M b_j(k) = 1$$

Le terme $b_j(k)$ représente la probabilité que l'on observe le symbole v_k à l'instant t , sachant que le modèle se trouve dans l'état j .

- $\pi = [\pi_i] \ 1 \leq i \leq N$ la distribution d'état initial :

$$\pi_i = p[q_1 = s_i], \ 1 \leq i \leq N \tag{IV.7}$$

$$\text{et } \forall i \ \pi_i \geq 0, \ \sum_{i=1}^N \pi_i = 1$$

π_i représente la probabilité que l'état de départ du modèle est l'état i .

5. Topologie d'HMMs

Le type d'architecture du modèle est l'un des éléments qui influencent sur la qualité de reconnaissance utilisant un modèle de Markov caché [RAB 89]. Il existe deux types principaux de modèles de Markov cachés, le modèle ergodique et le modèle gauche-droite [MIL 98a][BEN 07].

5.1. Modèle ergodique

C'est un modèle sans contrainte où toutes les transitions d'un état vers un autre sont possibles, c'est-à-dire, que tous les états peuvent être atteints de n'importe où en un seul pas, on peut remarquer que la connaissance de π est primordial pour le choix de l'état de départ.

5.2. Modèle gauche-droite

C'est un modèle contenant des contraintes sur des transitions, c'est-à-dire où il peut y avoir interdiction de certaines transitions :

$$a_{ij} = 0 \ \text{pour } j > i + \Delta$$

Cela veut dire que les transitions de l'état dont l'indice est inférieur à celui de l'état actuel sont interdites :

$$a_{ij} = 0 \ \text{pour } j < i$$

Il existe deux sous types de ce modèle qui sont le modèle parallèle et le modèle séquentiel.

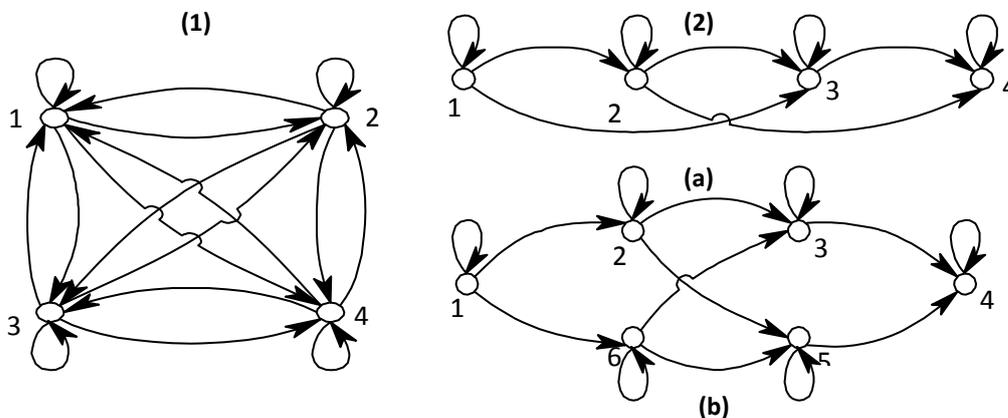


Figure IV.2 : (1) Modèle ergodique à 4 états, (2) Modèle gauche-droite : (a) le modèle séquentiel, (b) le modèle parallèle [RAB 89]

6. Les trois problèmes associés aux modèles de Markov Cachés

La modélisation de processus réels par des modèles de Markov cachés nécessite la résolution des trois problèmes de base suivants :

a) Problème de l'évaluation

Étant donné un modèle de Markov caché $\lambda = (A, B, \pi)$ et une séquence d'observations $O = \{o_1, o_2, \dots, o_T\}$, quelle est la probabilité conditionnée $p(O|\lambda)$ par ce modèle pour produire la séquence respective? La solution la plus efficace est représentée par l'algorithme Forward-Backward [MIL 98a] [MAH 99] [BEN 07]. Elle se montre donc particulièrement utile pour choisir, parmi plusieurs modèles, celui qui correspond le mieux à cette séquence.

b) Problème d'entraînement

Étant donné un modèle de Markov caché et une séquence d'observations $O = \{o_1, o_2, \dots, o_T\}$, comment peut-on ajuster les paramètres du modèle $\lambda = (A, B, \pi)$ pour maximiser la probabilité de production de la séquence $p(O|\lambda)$? C'est le point crucial de l'application des HMMs. Une très bonne solution pour ce problème peut être donnée par l'algorithme de Baum-Welch [MIL 98a] [MAH 99].

c) Problème de décodage

Étant donné un modèle de Markov caché $\lambda = (A, B, \pi)$ et une séquence d'observations, quelle est la meilleure séquence d'états $Q = \{q_1, q_2, \dots, q_T\}$ qui a donné naissance à la séquence d'observations $O = \{o_1, o_2, \dots, o_T\}$? La solution de ce problème nous permet d'estimer de façon optimale, la composante cachée du processus [MIL 98a] [MAH 99]. L'algorithme de Viterbi est l'un des algorithmes les plus utilisés afin de trouver la séquence d'états optimale. Dans les sections suivantes, on essaiera de donner une brève description pour chaque problème évoqué ci-dessus.

6.1. Solution du problème d'évaluation

Connaissant une suite d'observations, la règle de décision Bayésienne désigne le modèle qui les a émis. Si chaque mot est modélisé par un modèle λ [RAB 89].

$$p(O/M) = p(O/\lambda) \quad (\text{IV.8})$$

Il est nécessaire pour cela de calculer la probabilité d'émission de la suite des observations par chaque modèle. L'évaluation de cette probabilité est compliquée par le fait que le chemin parcouru n'est pas connu, mais un algorithme récursif efficace peut être utilisé [BEN 07].

6.1.1. Calcul de la probabilité d'émission d'une séquence d'observations

Il s'agit de calculer $p(O/\lambda)$ la probabilité de la séquence d'observations $O = \{o_1, o_2, \dots, o_T\}$ étant donné le modèle $\lambda = (A, B, \pi)$ et la séquence d'états. Avec T observations, donc N^T est le nombre de chaînes d'états possible. Considérons une chaîne d'états $Q = (q_1, q_2, \dots, q_T)$ dont q_1 est l'état initial. La probabilité de la série d'observations O étant donnée la chaîne d'états Q sera estimée comme suit :

$$p(O/Q, \lambda) = \prod_{t=1}^T p(o_t / q_t, \lambda) \quad (\text{IV.9a})$$

Si les observations sont statistiquement indépendantes alors :

$$p(O/Q, \lambda) = b_{q_1}(o_1) \cdot b_{q_2}(o_2) \cdot \dots \cdot b_{q_T}(o_T) \quad (\text{IV.9b})$$

Et la probabilité pour la chaîne d'états Q est :

$$p(Q/\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} \quad (\text{IV.10})$$

Dès lors

$$p(O/\lambda) = p(O/Q, \lambda) \cdot p(Q/\lambda) \quad (\text{IV.11})$$

La probabilité de la séquence d'observations étant donné le modèle λ est la somme de toutes les probabilités des chaînes Q :

$$p(O/\lambda) = \sum_Q p(O/Q, \lambda) \cdot p(Q/\lambda)$$

$$p(O/\lambda) = \sum_{q_1, \dots, q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \dots a_{q_{T-1} q_T} b_{q_T}(o_T) \quad (\text{IV.12})$$

Avec une machine à N états, le nombre de chemins possibles est de l'ordre de N^T , et l'ensemble des chemins devient très rapidement impossible à décrire. Il existe, heureusement un algorithme, dit « Forward-pass » qui permet de calculer cette quantité [MIL 98a] [BAU 72].

6.1.2. Procédure récursive directe (l'algorithme Forward)

L'algorithme « Forward » introduit une nouvelle variable $\alpha_t(i)$, appelée variable directe pour le calcul de la probabilité d'émission, et définie comme suit :

$$\alpha_t(i) = p(o_1, o_2, \dots, o_t, q_t = s_i | \lambda) \quad (\text{IV.13})$$

C'est la probabilité d'une séquence d'observations partielle O_t^i , aboutissant à l'état s_i au temps t , étant donné le modèle λ . Ces paramètres peuvent se calculer de manière inductive, en trois étapes :

Initialisation

$$\alpha_1(i) = \pi_i \cdot b_i(o_1) \quad 1 \leq i \leq N \quad (\text{IV.14a})$$

Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \right] \cdot b_j(o_{t+1}) \quad 1 \leq t \leq T-1, 1 \leq j \leq N \quad (\text{IV.14b})$$

Cette formule, permet le calcul des variables directes d'indice $t+1$ à partir de celles d'indice immédiatement inférieur. Elle explicite simplement le fait que l'état s_j peut être atteint à l'instant $t+1$ à partir de chacun des N états possibles, compte tenu de leurs probabilités respectives de transition vers cet état. Elle prend enfin en compte la probabilité de l'observation supplémentaire o_{t+1} pour l'état s_j .

Terminaison

Par conséquent, la probabilité $p(O/\lambda)$ sera donnée par :

$$p(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (\text{IV.14c})$$

La probabilité cherchée est obtenue en sommant, sur l'ensemble des N états finaux, les probabilités de la séquence complète d'observations aboutissant à l'état s_i .

Avec cette procédure la quantité d'opérations rasante n'est qu'environ de (N^2T) . Ce nombre est déjà beaucoup plus petit que $(2TN^T)$.

Cette amélioration tient simplement au fait qu'à chaque itération, ce sont les N mêmes variables $\alpha_t(i)$ de l'étape précédente, calculées une fois pour toutes, qui sont recyclées dans le calcul des $\alpha_{t+1}(i)$. La méthode exploite ainsi le fait que, quelle que soit la taille de la séquence d'observations, il n'y a, à chaque instant, que N états possibles.

6.1.3. Procédure récursive inverse (algorithme Backward)

L'algorithme " Backward " introduit une variable, dite variable inverse, définie comme suit :

$$\beta_t(i) = p(O_{t+1}^T | q_t = s_i, \lambda) \quad (\text{IV.15})$$

Il s'agit de la probabilité de la séquence partielle d'observations $O_{t+1}^T = \{o_{t+1}, o_{t+2}, \dots, o_T\}$ étant donné que l'état à l'instant t est s_i , pour le modèle $\lambda = (A, B, \pi)$.

Ces paramètres peuvent se calculer de manière inductive, en trois étapes :

Initialisation

$$\beta_T(i) = 1 \quad 1 \leq i \leq N \quad (\text{IV.16a})$$

Ce paramètre est arbitrairement fixé à 1 pour tout i .

Induction

$$\beta_t(i) = \sum_{j=1}^N a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j) \quad 1 \leq t \leq T-1, 1 \leq i \leq N \quad (\text{IV.16b})$$

Cette formule est l'analogie de celle utilisée lors du calcul de la variable directe, si ce n'est qu'elle procède à reculons.

Terminaison

Donc la probabilité $p(O/\lambda)$ sera donnée par :

$$p(O|\lambda) = \sum_{i=1}^N \pi_i \cdot b_i(o_1) \cdot \beta_1(i) \quad (\text{IV.16c})$$

Les variables directes et inverses sont toutes deux d'un grand intérêt pour résoudre les deux autres problèmes fondamentaux des modèles de Markov cachés.

6.2. Solution du problème d'apprentissage (estimation des paramètres)

La reconnaissance d'un caractère est rendue possible par l'évaluation de la probabilité d'émission des observations (algorithme Forward) par tous les modèles des mots. Cela suppose l'existence d'un modèle au moins pour chaque mot et l'apprentissage des paramètres de ces modèles. Ces paramètres sont les probabilités de transition entre états et les probabilités d'émission associées aux états, car la topologie du modèle, à savoir le nombre d'états des modèles, les transitions autorisées entre ces états l'alphabet des symboles émis est supposé fixé à priori. Ainsi, connaissant une suite d'observations émises par un modèle, il est possible de modifier les paramètres du modèle de manière à rendre l'émission des observations par le modèle. Il s'agit d'une estimation avec le critère du maximum de vraisemblance qui est réalisé par l'algorithme de Baum-Welch.

6.2.1. Maximum de vraisemblance

L'estimation par maximum de vraisemblance ou MLE consiste à choisir les paramètres du modèle λ afin de rendre maximale la probabilité d'émission des observations O par le modèle.

$$\tilde{\lambda} = \arg \max_{\lambda} p(O/\lambda) \quad (\text{IV.17})$$

Les formules de Baum-Welch permettent une ré-estimation itérative des paramètres a_{ij} et $b_j(k)$ du modèle en appliquant ce critère. À la suite du ré-estimation des paramètres du modèle $\tilde{\lambda}_i$, le nouveau modèle $\tilde{\lambda}_{i+1}$ vérifié :

$$p(O/\tilde{\lambda}_{i+1}) \geq p(O/\tilde{\lambda}_i) \quad (\text{IV.18})$$

La convergence vers un optimum local est démontrée, mais les valeurs initiales des paramètres A et B sont cruciales pour assurer une convergence correcte et rapide le plus près possible du maximum global. L'algorithme de Viterbi réalisant le décodage peut servir à l'initialisation des modèles.

6.2.2. Ré-estimation du modèle

La ré-estimation des paramètres du modèle λ est basée sur le comptage du nombre moyen des transitions observées entre les états i et j . Soit $\varepsilon_t(i, j)$ la probabilité d'être dans l'état s_i à l'instant t et dans l'état s_j à l'instant $t+1$ ayant le modèle λ et les séquences d'observations O :

$$\varepsilon_t(i, j) = p(q_t = s_i, q_{t+1} = s_j | O, \lambda) \quad (\text{IV.19})$$

En fonction des variables directes et inverses, on aura :

$$\varepsilon_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{p(O/\lambda)} = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)} \quad (\text{IV.20})$$

Dans laquelle, le numérateur est $p(q_t = s_i, q_{t+1} = s_j / O, \lambda)$ et la division par $p(O/\lambda)$ donne une mesure de probabilité.

En sommant ces probabilités sur le compte de j pour les N états, on obtient la probabilité d'occuper l'état s_i à l'instant t , étant donnés le modèle et la séquence d'observations, soit la variable $\gamma_t(i)$, définie précédemment :

$$\gamma_t(i) = \sum_{j=1}^N \varepsilon_t(i, j) \quad (\text{IV.21})$$

En sommant ces paramètres sur le compte de l'indice de temps t , on obtient une quantité qui peut être interprétée comme le nombre escompté de fois que l'état s_i est visité au cours de la séquence, ou encore, comme le nombre de transitions issues de l'état s_i au cours de cette séquence telle que:

$$\sum_{t=1}^{T-1} \gamma_t(i) : \text{nombre escompté de transitions depuis } s_i \quad (\text{IV.22a})$$

$$\sum_{t=1}^{T-1} \varepsilon_t(i, j) : \text{nombre escompté de transitions } s_i \text{ vers } s_j \quad (\text{IV.22b})$$

En utilisant les formules ci-dessus, il a été démontré qu'on peut ré-estimer les paramètres du modèle.

❖ Formules de ré-estimation Baum-Welch

Donc, la probabilité de transition entre les états i et j sera donnée par la relation suivante :

$$\begin{aligned} \bar{a}_{ij} &= \frac{\text{nombre escompté d'occurrences de l'état } s_i \text{ à l'état } s_j}{\text{nombre escompté de transitions depuis l'état } s_j} \\ &= \frac{\sum_{t=1}^{T-1} \varepsilon_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)} \end{aligned} \quad (\text{IV.23})$$

Tandis que la probabilité d'émission aura la formule ci-dessous :

$$\begin{aligned} \bar{b}_j(k) &= \frac{\text{nombre d'occurrences du symbole } v_k \text{ dans l'état } s_j}{\text{nombre d'occurrences de l'état } s_j} \\ &= \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} = \frac{\sum_{t=1}^T \alpha_t(j) \beta_t(j)}{\sum_{t=1}^T \alpha_t(j) \beta_t(j)} \end{aligned} \quad (\text{IV.24})$$

$\bar{\pi}_i$ = Nombre de fois d'être dans l'état s_i à l'instant $t=1$

Et :

$$\bar{\pi}_i = y_1(i) = \frac{\alpha_1(i) \beta_1(i)}{\sum_{j=1}^N \alpha_1(j) \beta_1(j)} \quad 1 \leq i \leq N \quad (\text{IV.25})$$

Partant de valeurs initiales des paramètres $\lambda = (A, B, \pi)$ du modèle, on peut calculer les paramètres des membres de droite, et déduire ainsi de ces trois relations les nouvelles estimations des paramètres $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$. La procédure peut alors être réitérée, Baum et al ont démontré qu'à chaque itération, la valeur de $p(O/\lambda)$ était accrue et l'algorithme converge donc vers un maximum de $p(O/\lambda)$. Il faut cependant noter que cette fonction est la plupart du temps très complexe, comportant un grand nombre de maxima locaux, et qu'il faut dès lors veiller à choisir de bonnes estimations initiales des paramètres pour être assuré de la convergence de l'algorithme vers le maximum global.

6.2.3. Autres critères d'apprentissage (MAP et MMI)

Le critère par maximum de vraisemblance MLE n'est pas nécessairement le meilleur critère d'apprentissage des modèles, car sa convergence nécessite une bonne initialisation des paramètres et un nombre élevé de données d'apprentissage. Pour surmonter ces problèmes il

existe d'autres critères d'apprentissage comme les critères MAP (Maximum A Posteriori) et MMI (Maximum Mutual Information).

Le critère Bayésien ou MAP (Maximum A Posteriori) maximise la probabilité a posteriori :

$$\tilde{\lambda} = \arg \max_{\lambda} p(O/\lambda) = \arg \max_{\lambda} p(O/\lambda)p(\lambda) \quad (\text{IV.26})$$

Le modèle λ est considéré comme une variable aléatoire, ce qui permet une estimation de la probabilité $p(\lambda)$. Les problèmes qui se posent sont le choix d'une densité de probabilité pour les paramètres du modèle λ , et l'estimation a posteriori du modèle. Lorsque les modèles sont a priori équiprobables, le critère MAP devient identique au MLE.

Le critère d'apprentissage par MMI tient compte de tous les modèles lors du ré-estimation, ce qui autorise à décomposer la probabilité à priori des observations $p(O)$ comme une somme des probabilités conjointes sur tous les modèles :

$$\tilde{\lambda} = \arg \max_{\lambda} p(\lambda/O) = \arg \max_{\lambda} \frac{p(O/\lambda)p(\lambda)}{\sum_m p(O/\lambda_m)p(\lambda_m)} \quad (\text{IV.27})$$

Les modèles sont modifiés simultanément de manière à maximiser la probabilité d'émission par le bon modèle tout en la diminuant pour les autres. Le MMI ne conduit pas à des formules de ré-estimation analytiques comme le MLE, sa mise en œuvre repose sur des techniques de type descente du gradient, il est aussi exigeant en temps de calcul, car l'apprentissage par MMI peut s'avérer plus coûteux que le MLE [BEN 07], de plus la convergence des modèles n'est pas toujours assurée. Une alternative au MMI est la stratégie d'apprentissage fondée sur la minimisation des erreurs, l'apprentissage correctif fut la première expérience utilisant ce critère [BEN 07], le principe est de corriger l'apprentissage MLE lorsque celui-ci ne classe pas correctement les données et de renforcer les modèles, son problème est que les paramètres des modèles sont corrigés de façon heuristique.

6.3. Solution du problème de décodage (reconnaissance)

Les algorithmes d'estimation directe et rétrograde fournissent la probabilité d'émission des observations cumulée sur toutes les séquences d'états possibles, sans choisir un chemin particulier. Il est parfois utile de connaître la séquence d'états cachés la plus probable et calcule la probabilité d'émission le long de ce chemin. La probabilité ainsi estimée néglige les chemins moins probables, et une reconnaissance à partir de cette

probabilité est sous-optimale par rapport à l'estimation directe et rétrograde. Mais la procédure fournit une segmentation du signal qui est exploitée pour l'apprentissage initial des modèles ainsi que pour le décodage.

La reconnaissance revient à trouver le meilleur chemin correspondant à l'observation dans un graphe. L'algorithme de Viterbi est particulièrement bien adapté à ce type de problème, il est basé sur les techniques de la programmation dynamique [FOR 73].

❖ Algorithme de Viterbi

Pour trouver le meilleur chemin $Q = \{q_1, q_2, \dots, q_T\}$ pour une suite d'observations donnée $O = \{o_1, o_2, \dots, o_T\}$, nous définissons la quantité $\delta_t(i)$ comme étant la probabilité du meilleur chemin partiel amenant à l'état s_i à l'instant t guidé par les t premières observations :

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} p[q_1, q_2, \dots, q_{t-1}, q_t = s_i, O_1^t | \lambda] \quad (\text{IV.28})$$

L'algorithme de calcul est donné par les étapes suivantes :

Initialisation

$$\delta_1(t) = \pi_i \cdot b_i(o_1) \quad 1 \leq i \leq N \quad (\text{IV.29a})$$

La valeur initiale est identique à $\alpha_1(i)$, et se calcule dès lors de la même façon.

Induction

$$\delta_t(j) = (\max_{1 \leq i \leq N} (\delta_{t-1}(i) \cdot a_{ij})) \cdot b_j(o_t) \quad 2 \leq t \leq T, 1 \leq j \leq N \quad (\text{IV.29b})$$

Afin de pouvoir reconstituer ultérieurement la suite des états parcourus, une trace de l'indice i qui conduit, à l'instant t , à la valeur optimale de l'expression (IV.29b), doit être conservée dans une fonction spécifique $\Psi_t(j)$.

$$\Psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) \cdot a_{ij}] \quad 2 \leq t \leq T, 1 \leq j \leq N \quad (\text{IV.29c})$$

Terminaison

$$p^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (\text{IV.29d})$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)] \quad (\text{IV.29e})$$

Meilleure chemin

Les indices des états qui constituent la séquence optimale peuvent alors être extraits des vecteurs $\Psi_t(j)$. Ceci s'effectue de manière récursive, en partant de l'instant final T :

$$q_t^* = \Psi_{t+1}(q_{t+1}^*) \quad t = T-1, T-2, \dots, 1 \quad (\text{IV.29f})$$

L'algorithme de Viterbi cherche la séquence d'états cachés la plus probable et calcule la probabilité d'émission le long de ce chemin. Au lieu de prendre en compte tous les chemins autorisés, seul le plus probable est gardé, ainsi il suffit de remplacer dans les équations précédentes des algorithmes Forward-Backward l'opérateur de somme (Σ) par MAX. Ce critère est largement utilisé en reconnaissance de l'écriture du fait du faible coût qui lui est associé.

7. Conclusion

Nous avons présenté au cours du présent chapitre une étude théorique générale sur l'un des meilleurs classifieurs vus jusqu'à nos jours. Les MMC sont des puissants outils dans le traitement du signal. Malgré leurs limites ils demeurent la technique la plus utilisée dans les systèmes de reconnaissance de la parole. La similarité qui existe entre la parole et l'écriture manuscrite a aidé les chercheurs pour faire une extension et ainsi migrer cette méthode dans le domaine de l'écriture pour résoudre les problèmes de similarité et variété entre les symboles.

CHAPITRE IV LES MODÈLES DE MARKOV CACHÉS

1. Introduction	65
2. Définitions préliminaires	65
3. Le modèle de Markov à temps discret	66
3.1. Propriétés de Markov du premier ordre	67
3.2. Paramètres d'un modèle de Markov discret	67
4. Le modèle de Markov caché discret (DHMM)	68
4.1. Définitions	68
4.2. Paramètres d'un modèle de Markov caché discret	69
5. Topologie d'HMMs	70
5.1. Modèle ergodique	70
5.2. Modèle gauche-droite	70
6. Les trois problèmes associés aux modèles de Markov Cachés	71
6.1. Solution du problème d'évaluation	71
6.1.1. Calcul de la probabilité d'émission d'une séquence d'observations	72
6.1.2. Procédure récursive directe (l'algorithme Forward)	73
6.1.3. Procédure récursive inverse (algorithme Backward)	74
6.2. Solution du problème d'apprentissage (estimation des paramètres)	74
6.2.1. Maximum de vraisemblance	75
6.2.2. Ré-estimation du modèle	75
6.2.3. Autres critères d'apprentissage (MAP et MMI)	77
6.3. Solution du problème de décodage (reconnaissance)	78
7. Conclusion	80

Chapitre V

Système hybride RN/MMC pour la reconnaissance de mots manuscrits arabe

1. Introduction

Dans les dix dernières années, des progrès considérables ont été réalisés dans le domaine de la reconnaissance de l'écriture manuscrite. Ce progrès est dû d'une part, aux nombreux travaux effectués dans ce domaine et d'autre part à la disponibilité de bases de données internationales standards relatives à l'écriture manuscrite qui permettait aux chercheurs de rapporter de façon crédible les performances de leurs approches dans ce domaine, avec la possibilité de les comparer avec d'autres approches, vu qu'ils utilisent les mêmes bases. L'industrie des téléphones portables intelligents et des assistants numériques personnels a bénéficié largement de ces progrès, ce qui a ramené les chercheurs à travailler d'avantage dans le domaine et d'essayer d'améliorer la qualité de la reconnaissance.

La langue arabe n'a pas eu cette chance, contrairement au latin, elle reste encore au niveau de la recherche et de l'expérimentation, c'est-à-dire que le problème reste encore un pari ouvert pour les chercheurs. L'écriture arabe étant par nature cursive, elle pose de nombreux problèmes aux systèmes de reconnaissance automatique. Le problème le plus difficile lors de la conception d'un système de reconnaissance de l'écriture manuscrite est la segmentation des mots manuscrits en vue de leur reconnaissance, qui n'est pas toujours triviale et demande beaucoup de temps et de calcul. D'autre part les systèmes se basant sur une analyse globale négligeant les informations locales ce qui peut diminuer considérablement la performance du système. Pour remédier à ces problèmes, des approches hybrides ont été proposées pour la reconnaissance des mots arabes manuscrits dans un vocabulaire limité. Dans un tel système, il doit nécessairement intégrer la prise en compte d'un nombre important de variabilités.

Notre travail consiste à la conception d'un système de reconnaissance d'écriture manuscrite arabe dans un vocabulaire limité, nous proposons une approche hybride basée sur les réseaux de neurones et les modèles de Markov cachés comme outil de classification.

2. La langue arabe

L'Arabe est une langue parlée par les Arabes dans plus de 20 pays. Elle est associée avec la région géographique du Moyen-Orient et le nord de l'Afrique. C'est une langue qui est aussi parlée comme une deuxième langue par plusieurs pays asiatique dans lesquels l'islam est la principale religion telle que l'Indonésie. Cependant, les langues non sémitiques telles que le Perse et quelques langues sud-africaines ont adopté l'alphabet arabe pour écrire.

À cause de la nature cursive du script arabe, il y a plusieurs caractéristiques qui rendent la reconnaissance de l'écriture distincte du script latin ou chinois.

Alphabet arabe

La langue arabe contient 28 lettres, elle est basée sur 15 formes distinctes qui varient selon leurs connexions avec les caractères précédents et/ou suivants. En utilisant une combinaison de points diacritiques et de symboles au-dessous et/ou au-dessus de ces formes en obtient la totalité des 28 caractères de l'arabe.



Figure V. 1 : Les classes de reconnaissance

La langue arabe est une langue cursive, il n'y a pas de lettre majuscule et quelques lettres ne se connectent pas avec les caractères qui les suivent.

La langue arabe s'écrit de droite à gauche et pour chaque lettre il y a 4 formes d'écriture dépendant de la position de ce dernier dans le mot (Tableau V. 1). Il y a la forme initiale, finale, médiane et la forme isolée. Dans quelques systèmes les classes sont séparées en 60(= 15 * 4) formes au lieu de 15 pour s'adapter à toutes les formes de chaque lettre.

Caractère	Isolé	Initial	Médian	Final
Alif	أ	أ	ا	ى / ا
Ba	ب	ب	ب	ب
Ta	ة / ت	ت	ت	ة / ت
Tha	ث	ث	ث	ث
Jim	ج	ج	ج	ج
Ha	ح	ح	ح	ح
Kha	خ	خ	خ	خ
Dal	د	د	د	د
Thal	ذ	ذ	ذ	ذ
Ra	ر	ر	ر	ر
Zin	ز	ز	ز	ز
Sin	س	س	س	س
Chin	ش	ش	ش	ش
Sad	ص	ص	ص	ص
Dad	ض	ض	ض	ض
Ta	ط	ط	ط	ط
Da	ظ	ظ	ظ	ظ
Ain	ع	ع	ع	ع
Gain	غ	غ	غ	غ
Fa	ف	ف	ف	ف
Kaf	ق	ق	ق	ق
Kaf	ك	ك	ك	ك
Lam	ل	ل	ل	ل
Mim	م	م	م	م
Noun	ن	ن	ن	ن
Ha	ه	ه	ه	ه
Waw	و	و	و	و
Ya	ي	ي	ي	ي

Tableau V. 1 : Les formes des caractères arabes dans des différentes positions

Le mot arabe n'a pas de longueur fixe, il peut comprendre un ou plusieurs pseudo-mots incluant chacun un nombre souvent différent de caractères. L'étude de la morphologie des pseudo-mots montre que l'écriture arabe présente des variations dans des bandes horizontales plus ou moins complexes en fonction de la calligraphie des caractères contenus dans le pseudo-mot [BEN et al 98]. La bande centrale est généralement la plus chargée au point de vue densité d'informations en pixels. Elle correspond aux lieux des ligatures horizontales, aux caractères centrés (sans extensions), aux boucles... La Figure V.2 donne des exemples d'écritures arabes imprimées et manuscrites.

Par ailleurs, les différents travaux élaborés en AOCR, montrent que la cursivité de l'écriture arabe, la complexité de la morphologie des caractères, les elongations des ligatures horizontales ainsi que les combinaisons verticales de certains caractères constituent les problèmes majeurs liés au traitement de cette écriture. En effet, ces problèmes engendrent une forte inertie à différents niveaux notamment dans :

- Le choix de primitives pertinentes décrivant la variabilité de la morphologie des caractères, sachant que certaines caractéristiques topologiques sont sensibles à la dégradation, notamment les points diacritiques et les boucles.
- La méthode de segmentation en caractères ou même en pseudo-mots (qui peuvent se chevaucher surtout dans le cas du manuscrit).

Tous ces problèmes et bien d'autres se trouvent accentués dans le cas du manuscrit où d'autres facteurs interviennent (variabilité intra et inter-scripteurs, conditions de l'écriture, fusion de points diacritiques, chevauchement de pseudo-mots, graphismes inégalement proportionnés...).

Face à ces problèmes, la nécessité d'une modélisation robuste s'impose, les méthodes classiques étant peu efficaces pour prendre en considération toutes les variations morphologiques de l'arabe.

3. Architecture du Système proposé

Notre système (figure V. 3) est constitué essentiellement de deux sous-systèmes de reconnaissance avec une interaction à deux niveaux, la première dans la segmentation et la deuxième dans la prise de décision. Nous allons dans la suite essayer de détailler chaque sous-système.

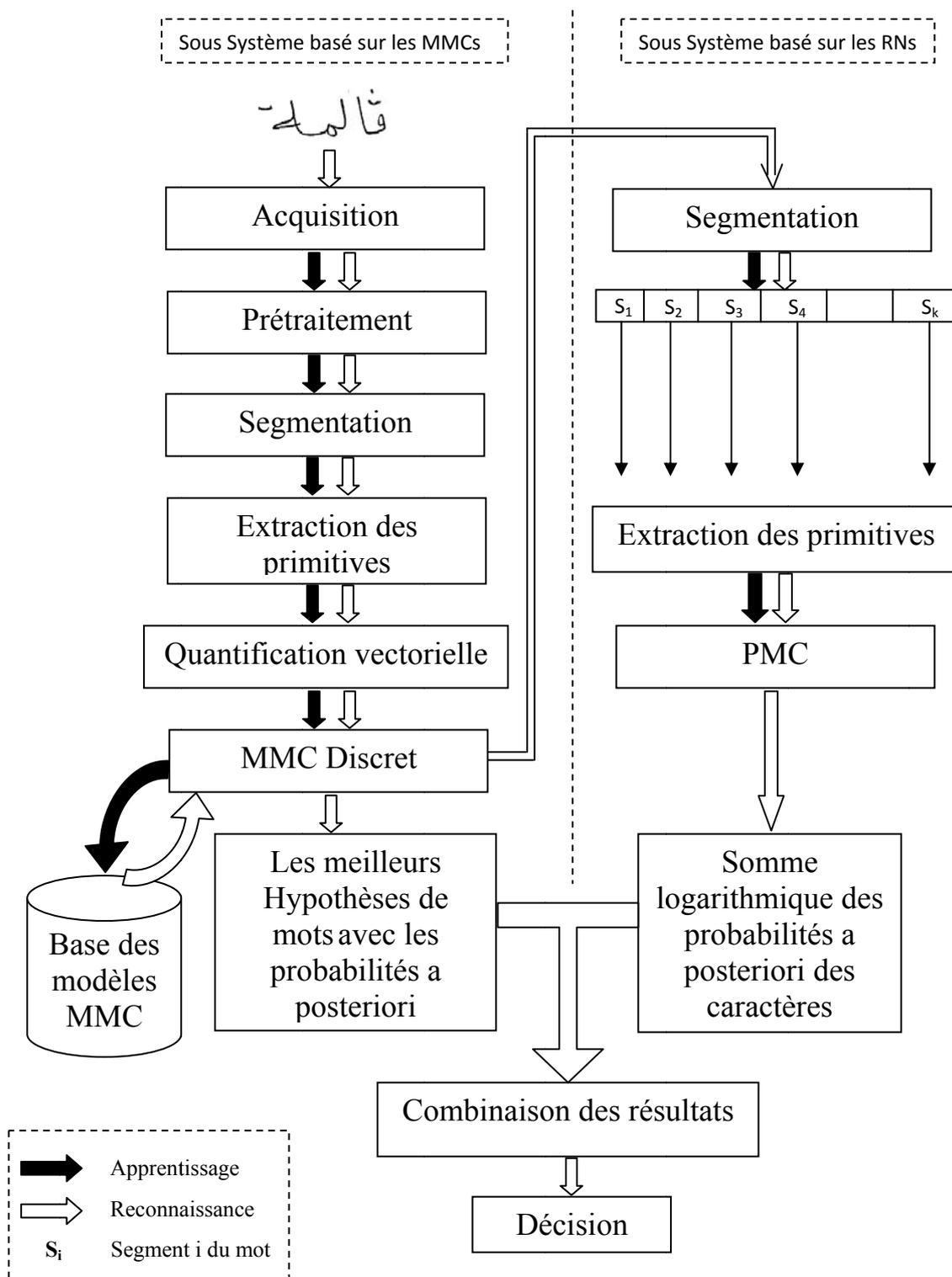


Figure V. 3 : Architecture du Système

3.1. Sous Système basé sur les Modèles de Markov cachés

Le sous-système proposé est composé de cinq modules : Acquisition, prétraitement, segmentation, extraction des primitives et le classifieur MMC que nous allons les détailler par la suite.

3.1.1. Acquisition

Le système reçoit en entrée l'image d'un mot arabe manuscrit écrit sans contrainte représentant une des wilayas algériennes stockées dans un fichier BMP.

3.1.2. Prétraitement

Les prétraitements appliqués sur l'image du mot permettent, d'une part, d'éliminer ou de réduire le bruit dans l'image, et d'autre part, de simplifier les traitements ultérieurs. Dans notre système nous avons utilisé la binarisation, le lissage, la normalisation, le cadrage, la squelettisation, la correction de la déformation des caractères et l'estimation de la ligne de base.

3.1.2.a. Binarisation

Le but de la binarisation est de faire surgir l'information utile par rapport à l'arrière-plan, malheureusement à cause de la mauvaise qualité de l'image reçue en entrée (Niveau du gris de l'arrière-plan très élevé) la méthode expliquée dans le chapitre *I.4.2.1* n'a pas donné de bons résultats ce qui nous a obligés à tester plusieurs seuils afin de trouver un compromis entre les différentes images utilisées. *La figure V.4* montre quelques résultats de l'opération de binarisation.

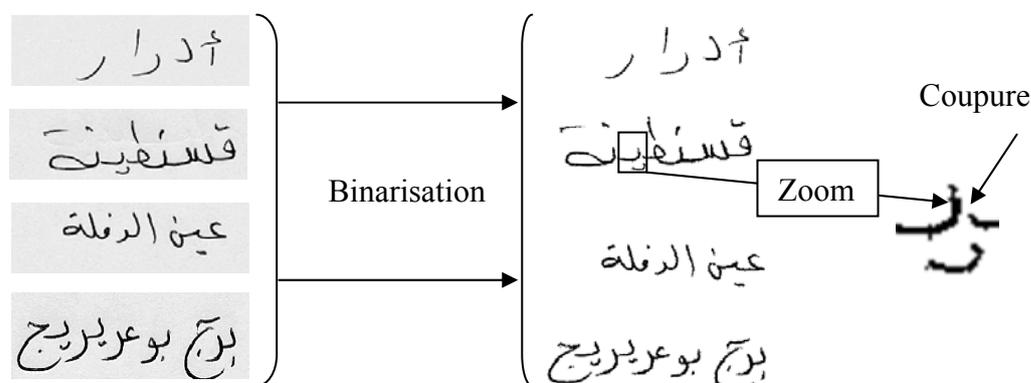


Figure V. 4 : L'opération de Binarisation et ses effets sur l'image

3.1.2.b. Lissage

L'opération de lissage est appliquée afin d'éliminer les bruits introduits dans l'image à cause des systèmes d'acquisition, les effets de temps ou tout simplement à cause de la qualité du papier et du stylo utilisés, en vue de la décrire par une séquence de vecteurs de caractéristiques plus au moins stables. Nous avons utilisé les opérations de nettoyage et de bouchage cité dans le chapitre *I.4.2.5*. La figure V.5 montre quelques résultats de l'opération de lissage réalisé dans notre système.

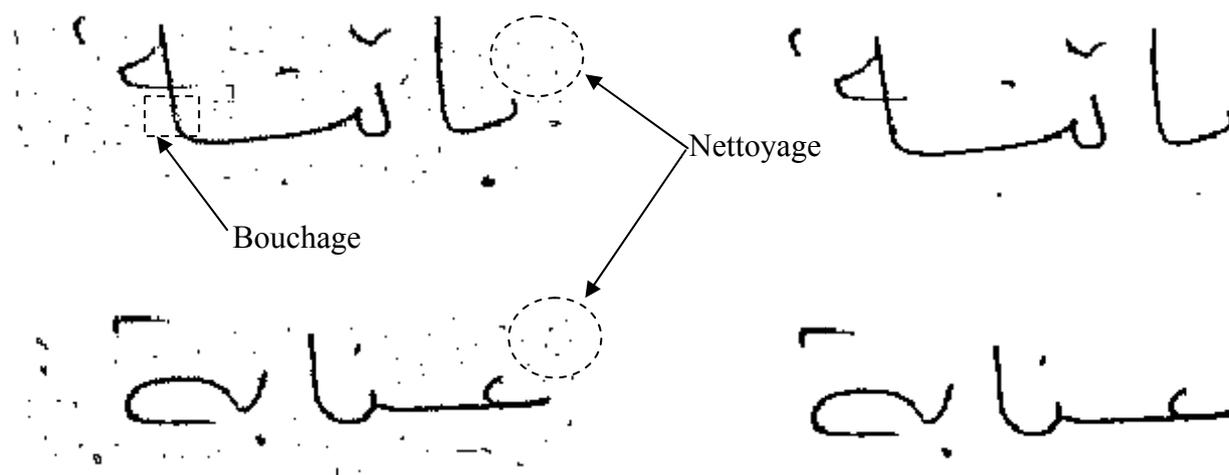


Figure V. 5 : Opération de Lissage

3.1.2.c. Normalisation

La normalisation est une tâche nécessaire lorsque l'acquisition n'est pas réalisée avec un scanner relié au système (image existante). En effet, si l'entrée du système est une image externe par rapport au système (Notre Système par exemple), il faut impérativement ramener les caractères à la même taille, car à cause de la variation des fontes ou des opérations d'agrandissement ou de réductions de la taille des images, les caractères peuvent subir une légère déformation dans la taille ce qui complique les tâches de segmentation et influence sur la stabilité des paramètres. Nous avons utilisé la technique décrite dans le chapitre *I.4.2.7* pour normaliser la taille de l'image.

3.1.2.d. Cadrage

Nous avons remarqué que la surface utilisable de l'image est très inférieure à la surface totale ce qui nous ramène à traiter un nombre important de pixel qui ne participe pas

dans la tâche de reconnaissance. Nous avons utilisé le cadrage pour minimiser la surface utilisable et donc minimiser le temps de traitement.

L'opération de cadrage consiste à chercher la première et la dernière ligne / colonne signifiante (pixel \diamond de l'arrière-plan), ensuite créer une nouvelle image cadrée à partir de l'image mère. La figure V.6 montre un exemple de cadrage.

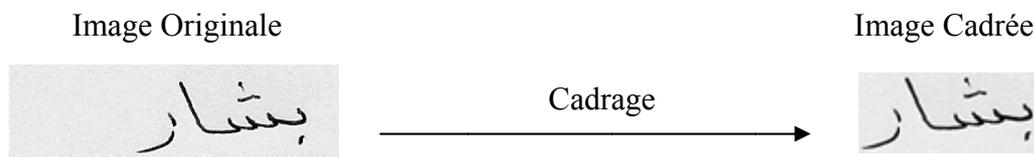


Figure V. 6 : Opération de Cadrage

3.1.2.e. Squelettisation

La squelettisation est l'une des techniques les plus utilisées dans la reconnaissance des formes. Elle permet de diminuer l'information utile en ne gardant que le squelette de la forme. Le principe est de ramener l'image du mot à une écriture linéaire d'une épaisseur égale à un pixel, en préservant la forme, la connexité et la topologie du tracé.

Il existe plusieurs algorithmes de squelettisation dans la littérature avec des degrés de performance très rapprochée dont les plus utilisés sont l'algorithme de Zhang-Suen, Huang et al., homotopy preserving algorithme, squelettisation utilisant la DTSA et squelettisation utilisant la SPTA. Nous avons choisi Zhang-Suen en nous basant sur la comparaison réalisée dans [HAD 05] et qui a montré que Zhang-Suen est l'algorithme le plus adapté pour le script arabe. La figure V.7 montre les résultats de la squelettisation sur quelques images de mots.

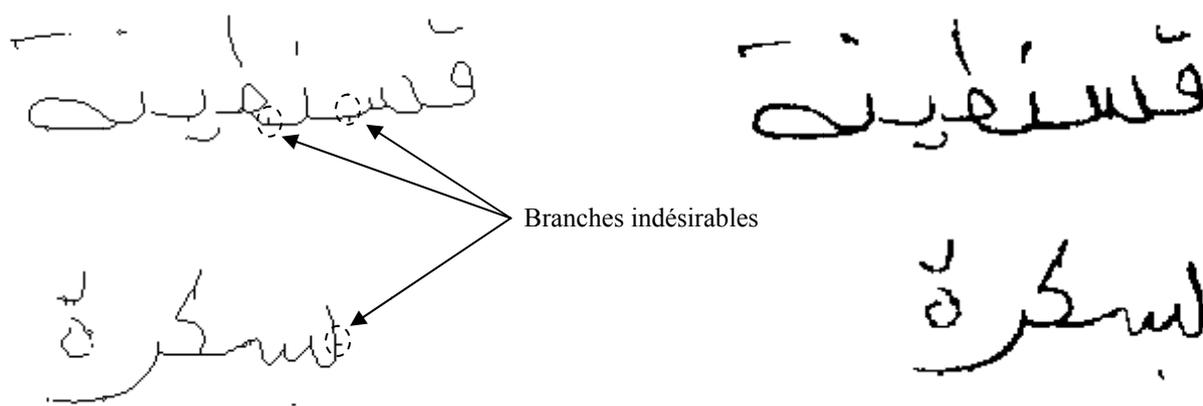


Figure V. 7 : Algorithme de squelettisation de Zhang-Suen appliqué sur des mots manuscrits arabes

Malgré l'efficacité de l'algorithme, nous avons remarqué quelques branches indésirables apparaître dans les squelettes des mots, en plus quelques points extrêmes ne respectent pas la contrainte de l'épaisseur égale 1 pixel. Pour remédier à ce problème, nous avons effectué des modifications sur l'algorithme de Zhang-Suen.

Algorithme de Zhang-Suen

Conditions

- | | | | | | | | | | | | |
|---|--|---|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| <p>1) $2 \leq N(P_1) \leq 6$</p> <p>2) $S(P_1) = 1$</p> <p>3) $P_2 \times P_4 \times P_6 = 0$</p> <p>4) $P_4 \times P_6 \times P_8 = 0$</p> | <p>5) $P_2 \times P_4 \times P_8 = 0$</p> <p>6) $P_2 \times P_6 \times P_8 = 0$</p> <p>$N(P_1) =$ le nombre de voisins non nul de P_1</p> <p>$S(P_1) =$ le nombre de transitions 0-1 dans l'ordre $P_2, P_3, P_4...$</p> | <table border="1" style="border-collapse: collapse; margin: auto;"> <tr> <td style="padding: 2px 5px;">P_9</td> <td style="padding: 2px 5px;">P_2</td> <td style="padding: 2px 5px;">P_3</td> </tr> <tr> <td style="padding: 2px 5px;">P_8</td> <td style="padding: 2px 5px;">P_1</td> <td style="padding: 2px 5px;">P_4</td> </tr> <tr> <td style="padding: 2px 5px;">P_7</td> <td style="padding: 2px 5px;">P_6</td> <td style="padding: 2px 5px;">P_5</td> </tr> </table> | P_9 | P_2 | P_3 | P_8 | P_1 | P_4 | P_7 | P_6 | P_5 |
| P_9 | P_2 | P_3 | | | | | | | | | |
| P_8 | P_1 | P_4 | | | | | | | | | |
| P_7 | P_6 | P_5 | | | | | | | | | |

Suppositions

- 1) tous les pixels de l'arrière-plan sont d'une valeur égale à 0
- 2) tous les pixels du corps de l'écriture sont d'une valeur égale à 1
- 3) le voisinage d'un pixel est une matrice 3x3 de pixel

Répéter pour chaque Pixel

- 1) Étiqueter tous les points de contour qui satisfaisaient les conditions (1), (2), (3) et (4)
- 2) Supprimer les points étiquetés
- 3) Étiqueter tous les points de contour qui satisfaisaient les conditions (1), (2), (5) et (6)
- 4) Supprimer les points étiquetés

Jusqu'à « aucun changement sur l'image »

Nous avons ajouté une étape (étape 5) dans l'algorithme qui permet d'éliminer les branches indésirables et d'accélérer la convergence vers la stabilité. Soit les masques suivants :

0 0 0	0 0 0	1 0 0	1 1 0	1 1 1	0 1 1	0 0 1	0 0 0
0 1 0	1 1 0	1 1 0	1 1 0	0 1 0	0 1 1	0 1 1	0 1 1
1 1 1	1 1 0	1 0 0	0 0 0	0 0 0	0 0 0	0 0 1	0 1 1
M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8

Algorithme de Zhang-Suen modifié

Répéter pour chaque Pixel

- 1) Étiqueter tous les points de contour qui satisfaisaient les conditions (1), (2), (3) et (4)
- 2) Supprimer les points étiquetés
- 3) Étiqueter tous les points de contour qui satisfaisaient les conditions (1), (2), (5) et (6)
- 4) Supprimer les points étiquetés
- 5) Appliquer les masques M_1 à M_8 sur le pixel

Jusqu'à « aucun changement sur l'image »

La figure suivante montre le résultat de la modification apportée à l'algorithme de Zhang-Suen. Vous remarquerez qu'il y a une élimination totale des branches indésirables et correction de la forme du squelette au niveau du caractère « ة » dans le mot « بسكرة »

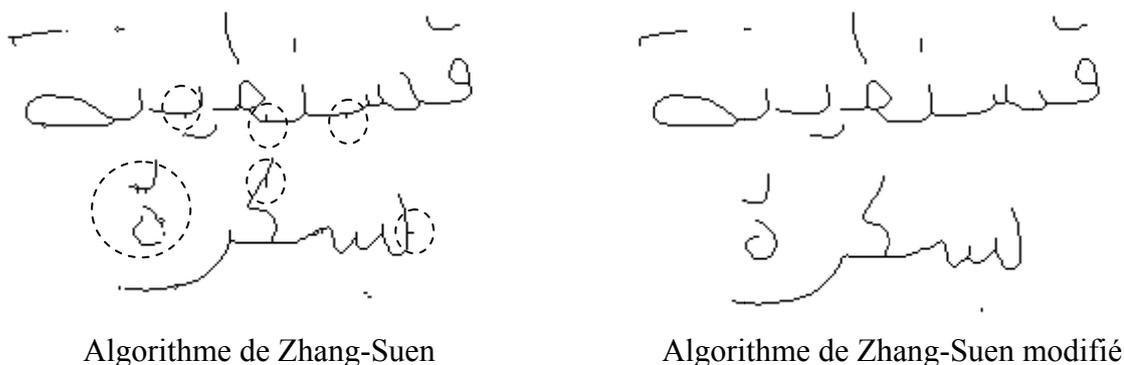


Figure V. 8 : Résultat de la squelettisation des mots manuscrits arabes en utilisant l'algorithme de Zhang-Suen modifié

3.1.2.f. Correction de la déformation des caractères

Nous désirons de cette étape la correction de la déformation des caractères. En effet après la squelettisation du corps des caractères, plusieurs déformations de type concave et convexe ont apparu au niveau des lignes continues (supposées sans concave ni convexe). La *figure V.9* montre quelques déformations détectées après l'étape de squelettisation.

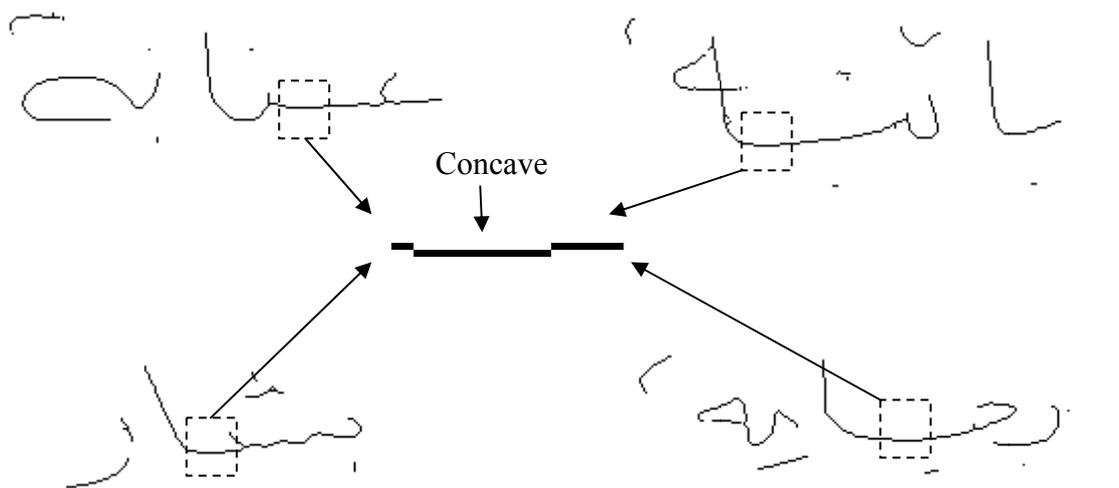


Figure V. 9 : Déformation des mots après la squelettisation (création de concave et convexe non désirées)

Pour remédier à ce problème, nous avons proposé un algorithme qui permet de redresser les lignes. Cet algorithme se base sur le principe de continuité c.-à-d. si deux segments horizontaux ou verticaux qui se trouvent sur la même ligne respectivement colonne, et s'il existe un autre segment horizontal respectivement vertical et ce dernier contient des pixels voisins avec les deux segments, alors c'est une déformation horizontale respectivement verticale. La correction s'effectue par le déplacement du segment (voir *figure V.10*). L'algorithme peut se résumer dans les instructions suivantes :

Algorithme

Répéter

Pour chaque ligne / colonne faire

- 1) Détecter les bords du segment **A**
- 2) Détecter les bords du segment **B**
- 3) S'il existe un troisième segment **C** dont les pixels des bords sont voisins avec le premier et le deuxième segment alors :
 - a. Créer un nouveau segment entre les segments **A** et **B**
 - b. Supprimer le segment **C**

Fsi

Fpour

Jusqu'à « aucune modification n'est possible sur l'image »

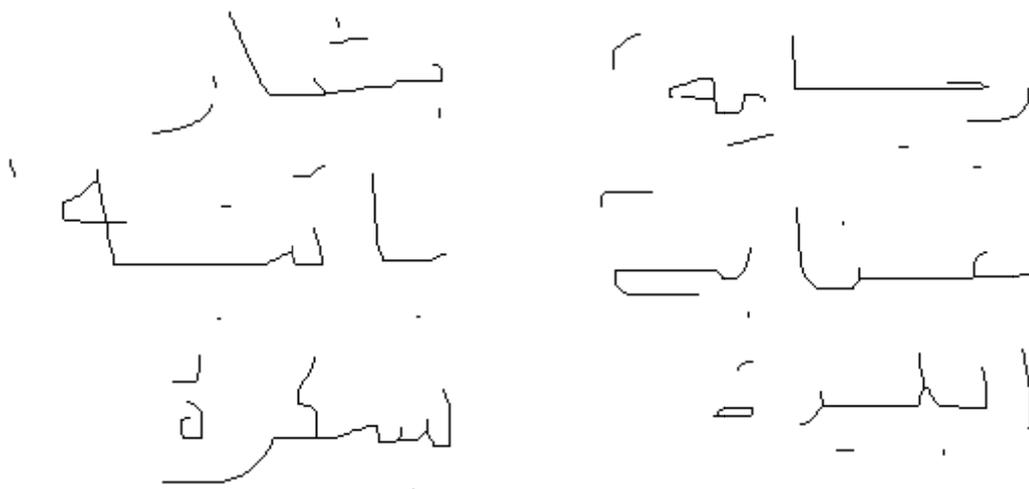


Figure V. 10 : Résultats de l'opération de correction de la déformation sur des mots manuscrits arabes

3.1.2.g. Estimation de la ligne de base

L'extraction de certaines caractéristiques (c.-à-d. points diacritiques) demande l'estimation de la ligne de base d'écriture du mot. La méthode utilisée donne une bonne estimation de la ligne de base. Elle est basée sur l'analyse de l'histogramme de projection horizontale. La figure V.11 montre le traçage de la ligne de base en utilisant l'histogramme de projection horizontale.

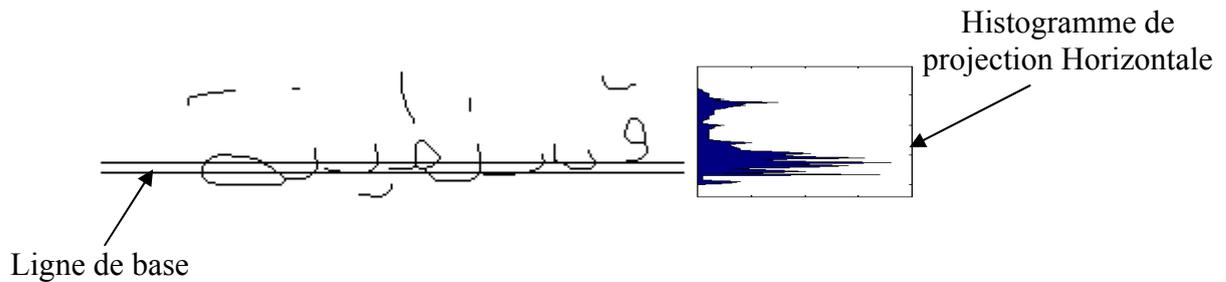


Figure V. 11 : Détection de la ligne de base en utilisant l'histogramme de projection horizontale

3.1.3. Segmentation

Nous avons utilisé dans notre système une segmentation non uniforme basée sur l'analyse de l'histogramme de projection verticale. Le but de cette méthode est de diminuer la dimension de l'information contenue dans l'image. Cette technique se base sur le principe que la liaison entre deux caractères est la partie la plus mince du tracé manuscrit. La figure V.12 montre le résultat de l'opération de segmentation sur le mot « قسنطينة ».

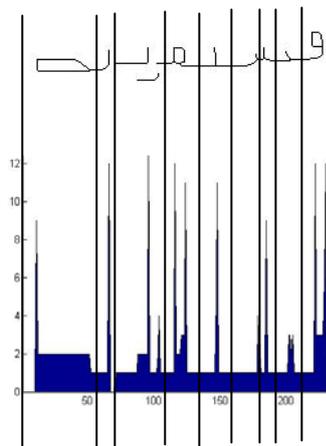


Figure V. 12 : Segmentation non uniforme utilisant l'analyse de l'histogramme de projection verticale

Comme la *figure V.12* le montre, nous avons éliminé les points diacritiques de l'image du mot pour éviter la déformation de l'histogramme de projection verticale. Concernant les points de segmentation, nous avons pris les minimums autour d'un pic. Nous avons utilisé un seuil égal à 1. Ce choix a été déterminé empiriquement après avoir testé plusieurs cas.

3.1.4. Extraction des primitives

L'identification directe du mot à partir de son image (matrice de pixels) est presque impossible à cause de la grande variabilité inhérente au style d'écriture utilisé et au bruit entachant l'image. D'où la nécessité d'extraire, à partir de la représentation en pixels du mot, un ensemble de caractéristiques permettant d'identifier facilement ce dernier. Ces primitives doivent être discriminatives et invariantes vis-à-vis les différentes transformations que peut subir l'image telle que la rotation, variation de la taille...etc.

Nous avons utilisé dans notre système un mélange entre les caractéristiques statistiques et structurelles qui peut d'après la littérature donner des meilleurs résultats. Le choix des caractéristiques n'est pas une tâche simple, malheureusement il n'y a pas de théorie qui permet de choisir tel ou tel caractéristique. Après plusieurs tests, nous avons choisi les caractéristiques suivantes :

- Les sept moments invariants.
- Le nombre de boucles.
- Le nombre et le type de chaque concave (vers la Gauche, vers la Droite, vers le Haut et vers le Bas).
- Les hampes et les Jambes.
- Le nombre, le type et la position des points diacritiques (Hamza, Madda et chapeau inclus).
- Nombre de points extrêmes.
- Nombre de points de branchement.
- Nombre de points de croisement.
- Le nombre de composantes connexes

Le résultat de cette étape est un vecteur de caractéristiques comportant 26 caractéristiques statistiques et structurelles. *Le tableau V.2* illustre le format du vecteur ainsi construit.

	Caractéristique	Code
	Caractéristiques structurelles	Nombre de Boucles
Nombre de Concave Vers la gauche		NCG
Nombre de Concave Vers la droite		NCD
Nombre de Concave Vers le haut		NCH
Nombre de Concave Vers le bas		NCB
Nombre de Hampes		NH
Nombre de Jambes		NJ
Un Point Haut		UPH
Un Point Bas		UPB
Deux Points Hauts		DPH
Deux Points Bas		DPB
Trois Points Hauts		TPH
Hamza		H
Maada		M
Chapeau		C
Nombre de Points Extrêmes		NPE
Nombre de Points de Branchement		NPB
Nombre de Points de Croisement		NPC
Nombre de composantes Connexes		NCC
Caractéristiques statistiques	Les sept moments invariants	M1
		M2
		M3
		M4
		M5
		M6
		M7

Tableau V. 2 : Structure du vecteur de caractéristiques pour le sous système basé sur le classifieur MMC

3.1.5. Quantification vectorielle

Les HMMs utilisés pour la modélisation des mots sont de nature discrète, leurs densités de probabilités d'observations sont discrètes, ce qui nécessite l'utilisation d'un quantifieur vectoriel pour faire correspondre chaque vecteur continu à un indice discret d'un dictionnaire de référence (CodeBook). Une fois le dictionnaire de référence obtenu, cette correspondance entre les vecteurs caractéristiques des trames et les indices du dictionnaire deviennent un simple calcul de type plus proche voisin.

Divers travaux ont été effectués dans ce sens pour développer des procédures itératives pour la conception des dictionnaires en se basant sur des séquences de vecteurs d'apprentissage représentatives. Cette procédure consiste premièrement à partitionner les

vecteurs caractéristiques obtenus (lors de l'étape d'extraction des primitives) à partir de toutes les images d'apprentissage, en M' ensembles disjoints (où M' est la taille du dictionnaire à concevoir). Deuxièmement, à représenter chaque ensemble par un vecteur unique ($v_k (1 \leq k \leq M')$), qui est généralement le centroïde des vecteurs caractéristiques de l'ensemble d'apprentissages affectés à la même classe. Ensuite, elle optimise itérativement la partition du dictionnaire (le centroïde de chaque région).

Nous avons décidé donc d'estimer la densité de probabilité d'émission des états par une quantification vectorielle, en appliquant l'algorithme *k-means*.

Algorithme K-means

Chaque vecteur x est quantifié par le prototype du dictionnaire dont il est le plus proche au sens d'une distance $d(x, y)$ définie dans l'espace.

$$x \xrightarrow{\text{QV}} \tilde{x} = v_k \quad \text{avec} \quad \tilde{k} = \arg \min_{1 \leq k \leq M'} d(x, v_k)$$

Il suffit ensuite de manipuler l'indice k du prototype à la place du vecteur x . Divers algorithmes ont été proposés pour la réalisation du dictionnaire des prototypes, le plus utilisé est le *k-means* dont nous rappelons ci-dessous son principe :

Étape 1 : Initialisation ; choisir l'ensemble des prototypes initial $\{v_k\} 1 \leq k \leq M'$.

Étape 2 : Classification ; classer l'ensemble des vecteurs x de l'espace de représentation dans des classes c_k par règle du plus proche voisin :

$$x \in c_k \quad \text{Si} \quad \text{est} \quad \text{Seulement} \quad \text{Si} :$$

$$d(x, v_k) \leq d(x, v_j) \quad \forall j \neq k \quad (\text{V.1})$$

Étape 3 : Recalcule du prototype de chaque classe

$$v_k = \text{centre}(c_k) \quad (\text{V.2})$$

Étape 4 : Test d'arrêt ; si ε est inférieur à une certaine limite, stop, sinon aller à l'étape 2.

$$\varepsilon = \frac{D(i) - D(i-1)}{D(i)} * 100\% \quad (\text{V.3})$$

Avec $D(i)$: distorsion totale à l'itération i ;

$$D(.) = \sum_k d(x / x \in c_k, v_k) \quad (\text{V.4})$$

3.1.6. Classifieur MMC

La modélisation du classifieur nécessite la définition des observations émises par les états du modèle et l'architecture des modèles de mots.

Pour définir l'architecture du modèle, nous devons tenir compte de la topologie et le nombre d'états du modèle. La topologie adoptée dans notre système est de type droite-gauche conformément à l'écriture arabe avec saut inter-états et intra-état. Ce type de modèles a l'avantage de conserver la notion du temps dans la modélisation, s'approchant ainsi de la nature de l'écriture. En outre, c'est le type le moins gourmand en temps de calcul et en nombre de paramètres à estimer lors de l'apprentissage. Dans notre modélisation chaque état correspond à un caractère c.-à-d. le nombre d'états est différent d'un modèle à un autre, ce qui nécessite la création de 48 modèles représentant les 48 wilayas algériennes.

Nous avons considéré les caractéristiques statistiques et structurelles extraites à partir de l'image de mot comme des observations pour notre modèle. La quantification vectorielle garantit la transformation du vecteur de caractéristiques en une séquence discrète d'observation. La figure V.13 montre la modélisation du mot « قالمة » par un MMC droite-gauche.

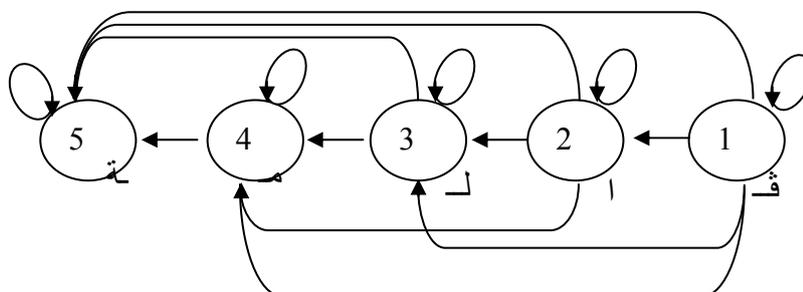


Figure V. 13 : Modèle HMM de type droite-gauche du mot قالمة

Apprentissage et reconnaissance

L'apprentissage des paramètres des modèles HMMs (A , B , Π) correspondant aux classes de mots est réalisé par l'algorithme de Baum-Welch (voir **Chapitre IV.6.6.2**) pour estimer les paramètres des différentes distributions de probabilité d'état. Cet algorithme permet d'aligner les observations sur les états, et d'une façon générale, converge vers un minimum local du fait de manque de données. La reconnaissance est effectuée par la recherche du modèle discriminant, elle peut se faire simplement par le calcul des probabilités

d'émission de la forme par les modèles que l'on suppose a priori équiprobables. La forme à reconnaître est affectée à la classe dont le modèle fournit la probabilité la plus importante :

$$\lambda^* = \operatorname{argmax}_{\lambda \in A} P(O/\lambda) \quad (\text{V.5})$$

Où A désigne l'ensemble des modèles. L'évaluation de la probabilité de chaque modèle est réalisée grâce à une méthode à base de programmation dynamique qui est l'algorithme de Viterbi (voir **Chapitre IV.6.3**).

3.2. Sous Système basé sur les réseaux de neurones

Le sous-système proposé est composé de trois modules : segmentation, extraction des primitives et le classifieur RN que nous allons les détailler par la suite.

3.2.1. Segmentation

La segmentation utilisée dans ce système est de type non uniforme, le nombre et la taille de chaque segment sont des informations extraites non pas de l'analyse de l'histogramme de projection verticale, mais à partir du premier sous-système. En effet, le classifieur MMC permet de donner après avoir déterminé le chemin le plus probable, le nombre de caractères à reconnaître ainsi que la surface occupée par le caractère dans l'image du mot.

3.2.2. Extraction des primitives

Nous avons utilisé dans notre système une combinaison de caractéristiques statistiques et structurelles. L'hybridation des caractéristiques est un peu différente par rapport au premier sous-système. En effet, nous avons privilégié les caractéristiques statistiques en nous basant sur les testes et les travaux effectués par Zeghdoudi dans [ZEG 08]. Voici une liste détaillée des caractéristiques statistiques utilisées :

- La moyenne, la variance et la valeur maximale / minimale calculées pour chaque histogramme de projection (Horizontale, verticale et diagonale 45° et 135°) du mot.
- La moyenne, la variance et la valeur maximale / minimale calculées pour chaque histogramme de transition (horizontale et verticale).
- Les sept moments invariants.

Pour les caractéristiques structurelles, nous avons utilisé le nombre de hampe, de jambe et de boucle. Le résultat de cette étape est un vecteur de 34 caractéristiques composé

de 31 caractéristiques statistiques et de 3 caractéristiques structurelles. *Le tableau V.3 illustre le format du vecteur de caractéristiques utilisé.*

Caractéristiques	Caractéristique	Code	
Caractéristiques structurelles	Le Nombre de Boucles	NB	
	Le Nombre de Hampes	NH	
	Le Nombre de Jambes	NJ	
Caractéristiques statistiques	Moyenne Histogramme de Projection Verticale	MHPV	
	Variance Histogramme de Projection Verticale	VHPV	
	Valeur Max Histogramme de Projection Verticale	MAHPV	
	Valeur Min Histogramme de Projection Verticale	MIHPV	
	Moyenne Histogramme de Projection Horizontale	MHPH	
	Variance Histogramme de Projection Horizontale	VHPH	
	Valeur Max Histogramme de Projection Horizontale	MAHPH	
	Valeur Min Histogramme de Projection Horizontale	MIHPH	
	Moyenne Histogramme de Projection Diagonale 45°	MHPD1	
	Variance Histogramme de Projection Diagonale 45°	VHPD1	
	Valeur Max Histogramme de Projection Diagonale 45°	MAHPD1	
	Valeur Min Histogramme de Projection Diagonale 45°	MIHPD1	
	Moyenne Histogramme de Projection Diagonale 135°	MHPD2	
	Variance Histogramme de Projection Diagonale 135°	VHPD2	
	Valeur Max Histogramme de Projection Diagonale 135°	MAHPD2	
	Valeur Min Histogramme de Projection Diagonale 135°	MIHPD2	
	Moyenne Histogramme de Transition Verticale	MHTV	
	Variance Histogramme de Transition Verticale	VHTV	
	Valeur Max Histogramme de Transition Verticale	MAHTV	
	Valeur Min Histogramme de Transition Verticale	MIHTV	
	Moyenne Histogramme de Transition Horizontale	MHTH	
	Variance Histogramme de Transition Horizontale	VHTH	
	Valeur Max Histogramme de Transition Horizontale	MAH	
	Valeur Min Histogramme de Transition Horizontale	MIHTH	
	Les sept moments invariants		M1
			M2
		M3	
		M4	
		M5	
		M6	
		M7	

Tableau V. 3 : Structure du vecteur de caractéristiques pour le sous système basé sur le classifieur RN

Le choix des caractéristiques n'est pas une tâche simple. En effet, pour atteindre la perfection il faut choisir les caractéristiques qui permettent la discriminabilité, la fiabilité,

l'indépendance et bien sûr qui permettent de minimiser le temps d'apprentissage, or jusqu'à présent il n'y a pas une théorie qui permet le choix de telles caractéristiques.

3.2.3. Classifieur RN

Le classifieur que nous avons utilisé est un PMC à rétro-propagation du gradient d'erreur à une couche cachée (Voir **Chapitre III.9.3**). Les 34 caractéristiques statistiques et structurelles extraites à partir des segments de caractère isolés générés par le module de segmentation sont les entrées du réseau. La couche cachée est composée de 25 neurones. Les classes à discriminer sont les 28 caractères de l'alphabet arabe, d'où le choix de 28 neurones pour la couche de sortie. La fonction d'activation des neurones est la fonction sigmoïde unipolaire (Voir **Chapitre III.5**). Ce choix de classifieur est basé sur les critères suivant : sa rapidité, sa capacité à traiter des données hétérogènes et le plus important, s'il est bien entraîné, un PMC estime des probabilités bayésiennes *a posteriori* [LIP 91]. Ce dernier point est très important dans notre modélisation vu que nous avons utilisé les modèles de Markov cachés et que ce dernier génère aussi des probabilités *a posteriori*. Donc, la mise en œuvre de la combinaison des résultats va être simplifiée.

3.3. Combinaison des résultats des classifieurs MMC/RN

Cette étape est le deuxième raccordement des deux sous-systèmes étant le premier au niveau de la segmentation. Nous voulons de cette étape la correction des inconvenances générées par les modèles de Markov cachés. En effet, malgré le progrès réalisé au niveau de la reconnaissance de la parole et de l'écriture manuscrite par les MMCs, nous les reprochons de négliger un peu les informations locales. De plus, la condition d'indépendance imposée par le modèle de Markov (chaque observation doit être indépendante des observations voisines) rend les MMC incapable de tirer avantage de la corrélation qui existe réellement parmi les observations d'un même caractère.

Pour effectuer la combinaison des résultats, nous avons calculé un nouveau score des mots à partir des probabilités *a posteriori* de chaque classifieur. En effet étant donné que les deux classificateurs estiment des probabilités *a posteriori* en sortie, nous pouvons calculer un score composé P^* par combinaison des sorties des classificateurs :

$$P^* = \log(P_{MMC}) + \log(P_{PMC}) \quad (\text{V.6})$$

La sortie du système (Complet) une liste des N meilleures hypothèses. Le choix du mot candidat est effectué par rapport à la probabilité la plus élevée.

4. Tests et Résultats

Pour construire notre système et évaluer ses performances, nous avons utilisé deux bases de données. La première contient 14400 échantillons écrits sur du papier par 100 scripteurs différents et 3 occurrences pour chaque mot. Les mots de cette base constituent un lexique de 48 mots des wilayas algériennes. Les échantillons de la base de données utilisée ont été collectés au sein du laboratoire de recherche en informatique d'Annaba. La deuxième contient 2800 caractères segmentés manuellement à partir des mots de la première base. Le choix des mots à segmenter est pris au hasard parmi les 14400 mots contenus dans la première base. Nous avons pris de chaque caractère 100 exemplaires qui n'appartiennent pas forcément au même scripteur. Nous avons construit la deuxième base pour éviter l'apprentissage collectif qui fait à lui seul l'objet d'une thèse de recherche. Les tableaux V.4 et V.5 montrent quelques échantillons des deux bases.

أدرار	أدرار	أدرار	أدرار	أدرار	أدرار
عين الدفلة					
عين تموشنت					
الجزائر	الجزائر	الجزائر	الجزائر	الجزائر	الجزائر
عناية	عناية	عناية	عناية	عناية	عناية
باتنة	باتنة	باتنة	باتنة	باتنة	باتنة
بشار	بشار	بشار	بشار	بشار	بشار
بجاية	بجاية	بجاية	بجاية	بجاية	بجاية
بسكرة	بسكرة	بسكرة	بسكرة	بسكرة	بسكرة
البلدية	البلدية	البلدية	البلدية	البلدية	البلدية
البويرة	البويرة	البويرة	البويرة	البويرة	البويرة
بومرداس	بومرداس	بومرداس	بومرداس	بومرداس	بومرداس
قسنطينة	قسنطينة	قسنطينة	قسنطينة	قسنطينة	قسنطينة
قالمة	قالمة	قالمة	قالمة	قالمة	قالمة

Tableau V. 4 : Quelques échantillons de la base des mots



Tableau V. 5 : Quelques échantillons de la base des caractères

La base des caractères a permis simplement de réaliser l'apprentissage du Perceptron multicouche, tandis que la base des mots a servi pour l'apprentissage des modèles de Markov cachés et pour le test des performances du système. Nous avons divisé la base de mots en deux sous-bases, la première contient 80 % (11520 échantillons) des mots pour l'opération d'apprentissage et la deuxième contient 20 % (2880 échantillons) des mots pour les tests.

Nous avons effectué plusieurs tests par rapport au changement de caractéristiques, par rapport au nombre de neurones de la couche cachée, par rapport à la présence et l'absence du module de correction de la déformation générée par la squelettisation et par rapport au seuil du module de binarisation.

Nous allons dans ce qui suit montrer seulement les résultats finaux de la reconnaissance des mots manuscrits arabes de notre système c.-à-d. avec toutes les corrections effectuées au niveau des modules du système (version finale du système). La meilleure configuration de notre système a donné les résultats illustrés dans *le tableau V.6*.

Wilayas	Taux de reconnaissance (en %)	Taux de rejet (en %)
أدرار	94,54	05,46
الشلف	88,37	11,63
الأغواط	89,71	10,29
أم البواقي	88,06	11,94
باتنة	92,32	07,68
بجاية	91,71	08,29
بسكرة	90,24	09,76
بشار	88,43	11,57
البلدية	94,32	05,68
البويرة	91,18	08,82

تبسة	93,42	06,58
تمنراست	88,37	11,63
تلمسان	90,44	09,56
تيارت	93,75	06,25
تيزي وزو	94,25	05,75
الجزائر	93,51	06,49
الجلفة	89,18	10,82
جيجل	92,46	07,54
سطيف	93,72	06,28
سعيدة	94,38	05,62
سكيكدة	92,31	07,69
سيدي بلعباس	90,54	09,46
عنابة	94,10	05,90
قالمة	88,81	11,19
قسنطينة	90,87	09,13
المدية	93,84	06,16
مستغانم	93,03	06,97
مسيلة	92,17	07,83
معسكر	89,36	10,64
ورقلة	89,58	10,42
وهران	91,86	08,14
البيضاء	94,62	05,38
اليزي	93,34	06,66
برج بوعريج	90,64	09,36
الطارف	93,61	06,39
بومرداس	93,62	06,38
تندوف	90,37	09,63
تسمسيلت	91,03	08,97
الوادي	88,29	11,71
خنشلة	91,08	08,92
سوق أهراس	92,74	07,26
تيبازة	92,66	07,34
ميلة	89,12	10,88
عين الدفلة	94,23	05,77
النعامة	92,10	07,90
غليزان	94,32	05,68
غرداية	92,13	07,87
عين تيموشنت	92,50	07,50
Taux moyen de Reconnaissance %	91,77	08,23

Tableau V. 6 : Résultats de la reconnaissance

D'après les résultats du tableau V.6 nous remarquons que le meilleur taux de reconnaissance est de **94,62 %** et le mauvais taux de reconnaissance et de **88,06 %** avec un taux de réussite globale de **91,77 %** et un taux de rejet de **08,23 %**. Nous voulons de cette présentation, montrer que malgré la simplicité du script de quelques mots (الشلف, بشار, قالمة,) (ميلة) le taux de reconnaissance été nettement faible par rapport aux autres mots. En effet, nous avons effectué une recherche minutieuse au niveau de chaque étape du processus de reconnaissance et nous avons remarqué que les images de ces mots ont perdu de l'information au niveau de la binarisation. Donc, c'est la qualité médiocre des images qui a perturbé le processus de reconnaissance. Concernant le taux d'échec dans chaque classe, il est justifié par les erreurs commises au niveau des autres modules de notre système.

Malheureusement à cause de l'architecture de notre système nous n'avons pas pu tester chaque sous-système à part pour montrer l'influence de chaque sous-système sur l'autre. Pour cela nous avons essayé de comparer notre système avec celui de Benzenache dans [BEN 07] basé sur les MMC et celui de Zaghdoudi dans [ZEG 08] basé sur le PMC. Nous estimons que la comparaison été crédible vu que nous avons utilisé la même base de mots.

	Notre Système (MMC/RN)	Système de Benzenache (MMC continu)	Système de Zaghdoudi (RN)
Taux de réussite	91,77 %	89,05 %	91,23 %
Taux de rejet	08,23 %	10,95 %	08,77 %

Tableau V. 7 : Comparaison des résultats de notre système avec celui de Benzenache [BEN 07] et Zaghdoudi [ZEG 08]

Le tableau V.7 montre que notre système a donné de meilleurs résultats par rapport aux deux systèmes de référence. Nous pouvons dire que nous avons atteint des résultats encourageants.

5. Conclusion

Nous avons présenté dans ce chapitre notre modélisation pour un système de reconnaissance de l'écriture manuscrite arabe à vocabulaire limité. Notre approche est basée sur l'hybridation des deux classifieurs les plus utilisés dans le domaine de la reconnaissance des formes et en particulier la reconnaissance de l'écriture et de la parole. Nous avons apporté quelques modifications sur l'algorithme de squelettisation de Zhang-Suen et nous avons aussi ajouté un module de correction de la déformation introduite par l'étape de squelettisation.

Malgré un taux de réussite de 91,77 % que nous estimons encourageant, notre système est loin d'atteindre la perfection. Les erreurs de classification sont dues d'une part à l'utilisation des MMCs discrets. En effet l'utilisation de la quantification vectorielle permet de représenter chaque classe par un vecteur unique (centroïde) ce qui influence sur la qualité de l'information véhiculée dans chaque vecteur. D'autre part, la qualité des images de la base est très médiocre ; la numérisation des images est une étape extrêmement importante dont il faut prendre le soin à le faire.

**CHAPITRE V SYSTEME HYBRIDE RN/MMC POUR LA RECONNAISSANCE DE
MOTS MANUSCRITS ARABE**

1. Introduction	81
2. La langue arabe	82
3. Architecture du Système proposé	86
3.1. Sous Système basé sur les Modèles de Markov cachés	87
3.1.1. Acquisition	87
3.1.2. Prétraitement	87
3.1.2.a. Binarisation	87
3.1.2.b. Lissage	88
3.1.2.c. Normalisation	88
3.1.2.d. Cadrage	88
3.1.2.e. Squelettisation	89
3.1.2.f. Correction de la déformation des caractères	91
3.1.2.g. Estimation de la ligne de base	93
3.1.3. Segmentation	93
3.1.4. Extraction des primitives	94
3.1.5. Quantification vectorielle	95
3.1.6. Classifieur MMC	97
3.2. Sous Système basé sur les réseaux de neurones	98
3.2.1. Segmentation	98
3.2.2. Extraction des primitives	98
3.2.3. Classifieur RN	100
3.3. Combinaison des résultats des classifieurs MMC/RN	100
4. Tests et Résultats	101
5. Conclusion	105

Conclusion générale

Malgré que plusieurs travaux aient été réalisés dans le domaine de la reconnaissance de l'écriture manuscrite, aucun système n'a réussi à atteindre la perfection. Donc, c'est un domaine qui est toujours ouvert aux propositions et aux expérimentations des chercheurs dans toutes les langues et spécialement le script arabe.

Dans ce cadre, nous avons proposé une solution au problème de la reconnaissance automatique de l'écriture manuscrite arabe dans un vocabulaire limité. Cette solution est fondée sur le développement d'une approche hybride basée sur les modèles de Markov cachés et les réseaux de neurones. L'idée était d'utiliser les modèles de Markov cachés pour contourner le problème de la segmentation des mots en caractères, mais le faible pouvoir discriminatif de ses derniers à diminuer la capacité décisionnelle du système. En contrepartie, la puissance discriminative des réseaux de neurones est affaiblie par la non-compétence des systèmes de segmentation.

Notre travail était focalisé sur la reconnaissance de mots manuscrits arabe représentant les wilayas algériennes et qui peut être étendu ultérieurement à la lecture des adresses postales et la vérification des chèques. Les tests ont été effectués sur une base de mots écrits par 100 scripteurs avec 3 occurrences pour les 48 wilayas c'est-à-dire une base de 14400 échantillons.

Les résultats obtenus sont tout à fait encourageants, ils ont montré que la modélisation des mots de reconnaissance utilisant une approche hybride (les modèles de Markov cachés et les réseaux de neurones) permet d'améliorer le pouvoir discriminatif.

Cette modeste expérience s'est avérée intéressante et encourageante, mais elle n'a pas la prétention d'être parfaite, car on peut toujours envisager des possibilités d'extensions futures telles que :

- L'utilisation d'autres caractéristiques qu'on juge pertinentes pour la discrimination des mots arabes. Jusqu'à présent, les caractéristiques utilisées dans la plupart des systèmes de reconnaissance de l'écriture manuscrite arabe sont une réplique directe des langues latines, or la langue arabe se distingue des autres langues par des caractéristiques propres à elle.

- Utiliser l'apprentissage collectif des approches hybrides. En effet, lors de la phase d'apprentissage de notre système nous avons utilisé deux bases différentes pour l'apprentissage des classifieurs dans le but de simplifier la mise en œuvre de notre système. Ce type d'apprentissage peut influencer sur la qualité de la reconnaissance.
- L'utilisation d'autres modélisations Markoviennes comme les modèles continus ou planaires.
- Utiliser au lieu d'un simple Perceptron multicouche, un TDNN pour mieux modéliser le temps au niveau de la reconnaissance des caractères.
- Intégration d'un PMC en amont des HMMs pour une meilleure estimation des probabilités d'émission d'observations.
- Développement d'un algorithme de sélection automatique des primitives les plus pertinentes.
- Combiner plusieurs classifieurs dans une architecture appliquant le principe de vote majoritaire
- Essayer de compléter le système avec un module de posttraitement utilisant les TALNs.

Bibliographie

[**AGA 93**] O.E. Agazzi, S. Kuo, E. Levin, R. Pieraccini, "Connected and degraded text recognition using planar hidden Markov models", Proc. IEEE International conference on acoustics, speech and signal processing (ICASSP'93), pp. V-113-V-116, 1993.

[**AKM 98**] H. Akroum, L. Hamami, "Prétraitement et segmentation d'images de documents", Proceedings 1st national workshop on signal & image processing, NWSIP'98, Sidi Bel Abbes-Algérie, pp. 24-27, 1er Décembre 1998.

[**AKR 98**] H. Akroum, L. Hamami, "Prétraitement et segmentation d'images de documents", Proceedings 1st national workshop on signal & image processing, NWSIP'98, Sidi Bel Abbes-Algérie, pp. 24-27, 1er Décembre 1998.

[**ALO 02**] Y. Al-Ohali, "Handwritten Word Recognition - Application to Arabic Cheque Processing", PhD Thesis, Concordia University, Montreal, Quebec, Canada, February 2002.

[**AMI 00a**] A. Amin, "Recognition of printed Arabic text based on global features and decision tree learning techniques", Pattern Recognition, Vol. 33, pp. 1309-1323, 2000.

[**AVI 96**] M. Avila : "Optimisation de modèles markoviens pour la reconnaissance de l'écrit". Thèse, université de Rouen, 1996.

[**AZI 02a**] N. Azizi, T. Sari, L. Souici-Meslati, M. Sellami., "Une architecture de combinaison floue de classifieurs neuronaux pour la reconnaissance de mots arabes manuscrits", CIFED'2002, Colloque International Francophone sur l'Ecrit et le Document, pp. 89-96, Hammamet, Tunisie, Octobre 2002.

[**AZI 02b**] N. Azizi, "Combinaison de classifieurs neuronaux basée sur la logique floue : application à la reconnaissance des mots arabes manuscrits", Mémoire de magister, Laboratoire LRI, Département d'informatique, Université d'Annaba, Algérie, 2002.

[**BAU 72**] L.E. Baum, "An inequality and associated maximization technique for probabilistic functions of Markov processes", Inequalities, Vol. 3 pp. 1-8, 1972.

[**BEL 92**] A. Belaïd et Y. Belaïd, "Reconnaissance des formes Méthodes et application 1992.

[**BEL 97**] A. Belaïd, G. Saon, "Utilisation des processus markoviens en reconnaissance de l'écriture". Traitement de signal, vol. 14 n°2, pp 161-178, 1997.

[**BEL 01**] A. Belaid, "Reconnaissance automatique de l'écriture et du document", Pour la science, 2001.

[**BEN 00**] A. Benouareth, "Reconnaissance de l'écriture arabe manuscrite par une approche hybride", Mémoire de magister, Labo. LRI, Département d'informatique, Université d'Annaba, Algérie, Février 2000.

- [BEN 07]** A. Benzenache, "Reconnaissance hors-ligne des mots arabes manuscrits par les modèles de Markov cachés" Mémoire de magister, Département Génie Electrique, Université 08 mai 45, Guelma, Algérie, 2007.
- [BEN 98]** N. Ben Amara, A. Belaïd, N. Ellouze, "Modélisation pseudo bidimensionnelle pour la reconnaissance de chaînes de caractères arabes imprimés". Proc. 1er Colloque international francophone sur l'écrit et le document (CIFED'98), pp.131-140, Québec, Canada, 1998.
- [BEN 99]** N. Ben Amara, "Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée". Thèse de doctorat, Université des sciences, des techniques et de médecine de Tunis II, Tunisie, 1999.
- [BEN 00]** N. Ben Amara, A. Belaïd, N. Ellouze, "Utilisation des modèles markoviens en reconnaissance de l'écriture arabe : état de l'art ", Colloque international francophone sur l'écrit et le document (CIFED'00), 2000.
- [BOS 00]** B. Bosselin, "Traitement de l'information, classification et Reconnaissance de formes", Copyright © 2000 Faculté polytechniques de Mons.
- [BOU 07]** H. Bouyedda, "Reconnaissance de l'écriture arabe imprimé par réseaux de neurones", mémoire de Magister, Université de Guelma 2007.
- [CHE 98]** M. Cheriet, H. Miled, C. Olivier, "Visual aspect of cursive Arabic handwriting recognition", Proceedings Vision Interface, VI'98, pp. 262-270, 1998.
- [CHE 94]** M.Y. Chen, A. Kundu, "A complement to variable duration hidden Markov in handwritten word recognition". Proc. IEEE. International conference on image processing (ICIP'94), vol.I, pp. 174-178, Austin, Texas, 1994.
- [CLA 92]** T. Claude, "Les réseaux de neurones artificiels", Juillet 1992.
- [COR 02]** A. Cornuéjols, A. Miclet, Y. Kodratoff, "Apprentissage artificiel: concepts et algorithmes", Eyrolles, 2002.
- [DAV 93]** E. Davalo, P. Naim, "Des réseaux de neurones", Eyrolles, 1993.
- [DRE 02]** Dreyfus et co-auteurs, "Réseaux de Neurones, Méthodologie et Applications", Eyrolles, 2002.
- [ELY 96]** A. El Yacoubi, "Modélisation markovienne de l'écriture manuscrite, application à la reconnaissance des adresses postales". Thèse de doctorat, Université de Rennes 1, 1996.
- [ENN 04]** A. Ennaji, "Systèmes d'apprentissage: Stratégies émergentes et connaissances", Habilitation à Diriger les Recherches, Université de Rouen, 2004.
- [ESS 99]** N. Essoukhri Ben Amara, "Utilisation des modèles de markov cachés planaires en reconnaissance de l'écriture arabe imprimée", Thèse de doctorat, Ecole Nationale d'Ingénieur de Tunis, Février 1999.

[FAR 05] N. Farah, L. Souici, M. Sellami, "Arabic Word Recognition by Classifiers and Context", JCST, Journal of Computer Science and Technology, Vol. 20, No. 3, pp. 402-410, May 2005.

[FEH 99] M.C. Fehri, "Reconnaissance de textes arabes multi fonte à l'aide d'une approche hybride neuro-markoviennes". Thèse de doctorat, Université des sciences, des techniques et de médecine de Tunis II, Tunisie, 1999.

[FOR 73] G.D. Forney, "The Viterbi Algorithm", Proc IEEE, Vol. 61, No. 3, pp. 268- 278, March 1973.

[HER 94] J. Hérault, C. Jutten, "Réseaux neuronaux et traitement du signal", Hermès, 1994.

[HEU 94] L. Heutte, "Reconnaissance de caractères manuscrits : Application à la lecture automatique des chèques et enveloppes postales", Thèse de Doctorat, Université de Rouen, 1994.

[HEU 03] L. Heutte, "Analyse et Reconnaissance de l'écriture : de Nouvelles Perspectives en Traitement Automatique de Documents Manuscrits", Habilitation à Diriger les Recherches, Université de Rouen, 2003.

[JOD 94a] J. F. Jodouin, "Les réseaux de neurones : Principes et définitions", Hermès, 1994.

[JOD 94b] J. F. Jodouin, "Les réseaux neuromimétiques : Modèles et applications", Hermès, 1994.

[KAM 04] W. Kammoun, A. Ennaji, "Reconnaissance de Textes Arabes à Vocabulaire Ouvert", CIFED'2004, Colloque International Francophone sur l'Écrit et le Document, pp. 223-228, La Rochelle, France, Juin 2004.

[KAN 02a] S. Kanoun, "Identification et analyse de textes arabes par approche affixale", Thèse de Doctorat, Université de Rouen, France, Octobre 2002.

[KAN 02b] S. Kanoun, A. Alimi, A. Ennaji, Y. Lecourtier, "Reconnaissance de mots arabes par approche affixale", CIFED'2002, Colloque International Francophone sur l'Écrit et le Document, pp. 21- 30, Hammamet, Tunisie, Octobre 2002.

[KHO 03] M.S. Khorsheed, "Recognising handwritten Arabic manuscripts using a single hidden Markov model", Pattern Recognition Letters, Vol. 24, pp. 2235-2242, 2003.

[KNE 01] S. Knerr, Y. Tay, P. Lallican, M. Khalid, C. Viard-Gaudin, "An off-line cursive handwritten word recognition system", In proceedings of IEEE region 10 conference 2001.

[KOE 02] A. Koerich, "Large vocabulary off-line handwritten word recognition" Thèse de Docteur Ecole de Technologie Supérieure, 2002.

[KUN 00] M. Kunt, "Traitement de l'information : reconnaissance des formes et analyse de scènes". Vol 3, 2000.

[LAH 97] S. Lahouar, N. Ben Amara, "Reconnaissance de l'écriture arabe imprimée par modèles de Markov cachés". 17èmes Journées tunisiennes en électrotechnique et automatique (JTEA'97), pp. 363-370, Nabeul, Tunisie, 1997.

[LEM 96] B. Lemarié, M. Gilloux, M. Leroux, "Un modèle neuro-markovien contextuel pour la reconnaissance de l'écriture manuscrite". Actes du congrès AFCET'96 de reconnaissance de forme et intelligence artificielle, Rennes, France, pp. 645-650, 1996.

[LER 97] Leroy, "Reconnaissance d'écriture manuscrite dynamique par approche descendante - caractérisation du style de l'écriture et application". Thèse de Docteur d'Université de Rennes1, avril 1997.

[LIP 87] R. Lippman, "An Introduction To Computing With Neurl Nets" IEEE Assp Magazine, Volume 4, n° 2, pp 4-22, Avril 1987.

[LIP 91] R. Lippmann, M. Richard "Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities ", *Neural Computation*, vol. 3, 1991, p. 461-483.

[MAC 43] W.C. Macculloch et W. Pitts, "A logical calculus of the ideas immament en nervous activity", page 115-133, Bulletin of mathematical biophysics, vol.15, 1943.

[MAH 99] M.A. Mahjoub, "Application des modèles de Markov cachés stationnaires et non stationnaires à la reconnaissance en-ligne des caractères arabes ", Thèse de doctorat, Université des sciences, des techniques et de médecine de Tunis II, Tunisie, 1999".

[MEN 01] A. Menasria et A.H. Boualleg, "Commande d'un processus par Modèle inverse Neural", Mémoire d'ingénieur, Université de Geulma 2001.

[MIL 97] H. Miled, C. Olivier, M. Cheriet, Y. Lecourtier, "Coupling observation/letter for a Markovian modelisation applied to the recognition of arabic handwriting". IEEE Proc. 4th International conference on document analysis and recognition (ICDAR'97), pp. 580-583, Ulm, Germany, 1997.

[MIL 98a] H. Miled, "Stratégies de résolution en reconnaissance de l'écriture semi-cursive : application aux mots manuscrits arabes", Thèse de doctorat, Université de Rouen, 1998.

[MIL 98b] H. Miled, M. Cheriet, C. Olivier, "Multi-level arabic handwritten words recognition", Lecture Notes in Computer Science, LNCS 1451, pp. 944-951, Advances in Pattern Recognition, Proceedings of joint international workshops SSPR'98 and SPR'98, Sydney, Australia, August 1998.

[MIL 98c] H. Miled, M. Cheriet, C. Olivier, Y. Lecourtier, "Modélisation markovienne de l'écriture arabe manuscrite: une approche analytique", Actes CIFED'98, 1er Colloque International Francophone sur l'Ecrit et le Document, pp. 50-59, Québec, Canada, Mai 1998.

[MIL 00] H. Miled, C. Olivier, M. Cheriet, "Modélisation de la notion de pseudo-mot en reconnaissance de mots manuscrits arabes", Actes CIFED'2000, Colloque International Francophone sur l'Ecrit et le Document, pp. 161-170, Lyon, France, 2000.

[OLL 99] D. Ollivier, "Une approche économisant les traitements pour reconnaître l'écriture manuscrite: application à la reconnaissance de montants littéraux de chèques bancaires", Thèse en Informatique, Laboratoire LIX, Paris, 1999.

[PAQ 00] T. Paquet, "Reconnaissance de l'écriture manuscrite: des modèles aux systèmes", Habilitation à diriger les recherches, Université de Rouen, Décembre 2000.

- [PEC 03]** M. Pechwitz, V. Märgner, "HMM based approach for handwritten Arabic word recognition using the IFN/ENIT - database", International Conference on Document Analysis and Recognition, ICDAR, pp. 890-894, Edinburgh, Scotland, 2003.
- [RAB 89]** L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", Proc. IEEE Vol. 77, No. 4, pp. 336-349, August 1989.
- [RAG 03]** N. Ragot, "Reconnaissance de formes par modélisation mixte intrinsèque /discriminante à base de système d'inférence floue hiérarchisés". Thèse de docteur de l'université de Rennes1, octobre 2003.
- [SAE 96]** J. Saelen, O. Platmont, L. Mime, "Les réseaux de neurones", exposé, P4, Janvier 1996.
- [SAR 00]** T. Sari, "Un système de reconnaissance de mots arabes manuscrits basé segmentation", Mémoire de magister, Laboratoire LRI, Département d'informatique, Université Badji Mokhtar, Annaba, Algérie, Février 2000.
- [SAR 02]** T. Sari, L. Souici, M. Sellami, "Off-line Handwritten Arabic Character Segmentation and Recognition System: ACSA-RECAM", IWFHR'2002, 8th International Workshop on Frontiers in Handwriting Recognition, pp. 452-457, Niagara-on-the-Lake, Ontario, Canada, August 2002
- [SAR 05]** T. Sari, M. Sellami, "Cursive Arabic script segmentation and recognition system", International Journal of Computers and Applications, Vol. 27, No. 3, 2005.
- [SEB 96]** M. Sébastien, "Un moteur de type réseau de Neurones pour la révision des connaissances dans un système tueur intelligent", Diplôme d'étude approfondies, 1995-1996.
- [THO 81]** P. Théo, "Algorithms for graphics and image processing". Hurray Hill, New Jrsay, 1981.
- [TRI 96]** O.D. Trier, A.K. Jain, T. Taxt, "Feature extraction methods for character recognition: a survey", Pattern Recognition, Vol. 29, No. 4, pp. 641-662, 1996.
- [ZEG 08]** R. Zaghoudi, "Reconnaissance hors-ligne des mots arabes manuscrits par les réseaux de neurones" Mémoire de magister, Département Génie Electrique, Université 08 mai 45, Guelma, Algérie, 2008.
- [ZER 00]** N. Zermi, "Reconnaissance de mots manuscrites arabes par un modèle neuromarkovien". Thèse de Magister d'université, Badji Mokhtar-Annaba, 2000.