

# وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR - ANNABA UNIVERSITY  
UNIVERSITE BADJI MOKHTAR - ANNABA



جامعة باجي مختار - عنابة

Faculté: Sciences de l'Ingénieur  
Département: Electronique

Année : 2007

## THESE

Présentée en vue de l'obtention du diplôme  
Du Doctorat

*Reconnaissance de mots manuscrits  
Arabe par les modèles de Markov cachés  
et les réseaux de neurones*

Option : Traitement du signal

Par : ZERMI Narima

Devant le jury composé de

|              |      |            |              |
|--------------|------|------------|--------------|
| H.TEBBIKH    | Prof | President  | Univ. GUELMA |
| BEDDA Mouldi | Prof | Rapporteur | Univ. ANNABA |
| N.DOGHMANE   | Prof | Examineur  | Univ. ANNABA |
| DJ.CHIKOUCHE | Prof | Examineur  | Univ.SETIF   |
| L.ZOUAOUI    | MC   | Examineur  | Univ.SETIF   |
| M.RAMDANI    | CC   | Invité     | Univ.ANNABA  |

## *Remerciements*

Je tiens à exprimer toute ma reconnaissance à Monsieur MOULDI BEDDA, Professeur à l'université d'Annaba, trouve ici l'expression de ma profonde reconnaissance, pour m'avoir proposé cette étude. Son expérience et son recul scientifique ont éclairé mon chemin tout le long de ce travail. Je tiens à le remercier particulièrement pour la confiance qu'il m'a accordée et le soutien qu'il m'a apporté pour la réalisation de cette thèse. Je le remercie sincèrement pour la pertinence de ses remarques c'est grâce à son soutien et ses encouragements que ce travail a vu le jour.

Je tiens à remercier Monsieur HICHEM TEBIKH, Professeur à l'institut d'électronique de l'université de Guelma, de l'honneur qu'il me fait en présidant ce jury.

Que Monsieur NOUR-EDDINE DOGHMANE, Professeur à l'institut d'électronique de l'université d'Annaba, trouve ici ma profonde gratitude d'avoir accepté d'être examinateur de cette thèse.

Je remercie Messieurs CHIKOUCHE Professeur & Mr ZOUAOUI Maître de conférence à l'institut d'électronique de l'université de Setif, d'avoir accepté d'examiner mon travail.

J'exprime toute ma gratitude à Monsieur MESSAOUD RAMDANI, chargé de cours à l'institut d'électronique de l'université de Annaba, pour l'intérêt qu'il a manifesté tout au long de la réalisation de ce travail et pour les conseils précieux qu'il n'a cessé de me prodiguer. Je tiens à le remercier pour l'aide précieuse qu'il m'a apportée et pour le travail considérable qu'il a consacré au suivi et à l'élaboration de cette thèse.

J'exprime ma sincère gratitude à Mr NAILI MOHAMED, pour le soutien moral et l'aide qu'il m'a apportée tout le long de ce travail pour mener à bien cette thèse. Enfin, je tiens à remercier toutes les personnes qui m'ont facilité la réalisation de ce travail.

# Table des matières

|  |     |
|--|-----|
| ملخص.....  | I   |
| Abstract.....  | II  |
| Résumé.....  | III |
| Liste des tableaux.....  | V   |
| Liste des figures.....   | VI  |
| Introduction Générale.....   | X   |
| <b>Chapitre 1 : LA RECONNAISSANCE DE L'ECRITURE :</b>                              |     |
| <b>PROBLEMES ET SOLUTIONS EXISTANTES</b>   |     |
| 1.1 Vue d'ensemble.....  | 01  |
| 1.1.1 En Ligne, Hors Ligne.....  | 01  |
| 1.1.2 Styles d'écriture.....   | 01  |
| 1.1.3 De l'image Au Résultat.....  | 02  |
| 1.1.4 Constat et Limites.....  | 02  |
| 1.2 Les prétraitements D'image.....  | 03  |
| 1.2.1 Graphèmes .....  | 03  |
| 1.2.2 Segmentation explicite_ implicite.....                                       | 05  |
| 1.2.3 Caractéristiques.....  | 05  |
| 1.3 Reconnaissance statistique.....  | 06  |
| 1.3.1 Classification en mots.....  | 06  |
| 1.3.2 Séquence et forme.....   | 06  |
| 1.3.3 Choix d'une modélisation.....  | 07  |
| 1.4 Modélisation.....  | 07  |
| 1.4.1 Modèles de Markov cachés hybrides et classifieur quelconque.....             | 07  |
| 1.4.2 Modèles de Markov cachés hybrides et lois gaussiennes.....                   | 08  |
| 1.4.3 Modèles de Markov cachés hybrides et réseau de neurones.....                 | 09  |
| 1.4.4 Réseau de neurones incluant des prétraitements d'images.....                 | 10  |
| <b>CHAPITRE 2 : RECONNAISSANCE AUTOMATIQUE DE L'ECRITURE</b>                       |     |
| <b>ARABE</b>   |     |
| 2.1 Introduction.....  | 11  |
| 2.2 Caractéristiques de l'écriture Arabe.....                                      | 12  |
| 2.3. Un modèle général pour la reconnaissance optique de textes Arabes (AOTR)..... | 15  |
| 2.3.1. Acquisition.....  | 16  |
| 2.3.2. Prétraitement.....  | 17  |
| 2.3.3. Segmentation .....  | 19  |
| 2.3.4. Extraction de caractéristiques.....   | 20  |
| 2.3.5. Classification.....   | 23  |
| 2.3.6. Les approches hybrides .....  | 23  |
| 2.3.7. Apprentissage .....   | 24  |
| 2.3.8. Post-traitement.....  | 25  |
| 2.4 Les Perspectives .....   | 26  |
| 2.5 Conclusion.....  | 28  |
| <b>CHAPITRE 3 : Methodes et outils actuels de la reconnaissance</b>                |     |
| <b>des textes manuscrits</b>   |     |
| 3. Introduction.....   | 29  |
| 3.1. Les primitives, outils de la reconnaissance.....                              | 29  |
| 3.1.1. Les objectifs de l'extraction des primitives.....                           | 29  |
| 3.1.2. La problématique de l'extraction de l'information.....                      | 30  |
| 3. 2. L'extraction des primitives.....   | 32  |
| 3.2.1. Les approches de l'extraction des primitives.....                           | 32  |

|   |     |
|---|-----|
| 3. 2.2. Les catégories de primitives.....                                 | 32  |
| 3.2.3. Les étapes de l'extraction des primitives.....                     | 33  |
| 3.3 Conclusion.....   | 35  |
| <b>Chapitre 4 :</b>   |     |
| <b>Extraction des paramètres et méthodes de reconnaissance</b>            |     |
| 4.I.1-Introduction.....   | 36  |
| 4.I.2-Prétraitement :.....  | 36  |
| 4.I.3- Segmentation en Graphèmes : .....                                  | 39  |
| 4.I.4- Les Différents Groupes de Segments .....                           | 41  |
| 4.I.5-Les caractéristiques extraites des images binaires.....             | 41  |
| 4.I.6-Les Caractéristiques Extraites des Contours.....                    | 46  |
| 4.I.7- Représentation par le squelette.....                               | 47  |
| 4.I.8. Conclusion.....  | 49  |
| <b>Chapitre 5 : Application :</b>   |     |
| <b>Reconnaissance par modèles de Markov cachés et réseaux de neurones</b> |     |
| 5.1 Introduction.....   | 50  |
| 5.2. Reconnaissance des graphèmes par des réseaux de neurones.....        | 51  |
| 5.3- Modèles de Markov cachés.....  | 54  |
| 5.4- Reconnaissance de mots .....   | 55  |
| 5.5- Résultats expérimentaux.....   | 57  |
| Partie II : Reconnaissance de chiffres manuscrits                         | 62  |
| 5.II.1. Conception et réalisation.....                                    | 62  |
| 5.II.2. Reconnaissance de chèque.....                                     | 62  |
| 5.II.3. Conception et réalisation.....                                    | 62  |
| 5.II.4 Module du prétraitement local.....                                 | 64  |
| 5.II.5 Module d'extraction des caractéristiques .....                     | 66  |
| 5.II.6 Module d'apprentissage .....                                       | 67  |
| 5.II.7 Conception du réseau de neurones.....                              | 68  |
| 5.II.8. Description générale des expériences.....                         | 68  |
| 5.II.9. Test et résultat.....   | 69  |
| 5.II.10. Analyse des résultats.....                                       | 72  |
| Conclusion générale.....  | 75  |
| <b>Annexe A :Réseaux de neurones</b>                                      |     |
| A.1 Définition des réseaux de neurones multicouches.....                  | 77  |
| A.1.1 Un neurone.....   | 77  |
| A.1.2 Une couche de neurones.....   | 78  |
| A.1.3 Un réseau de neurones : le perceptron.....                          | 78  |
| A.1.4 La régression.....  | 80  |
| A.2 Régression par un réseau de neurones multi-couches.....               | 84  |
| A.3 Méthode d'optimisation de Newton.....                                 | 86  |
| A.4 Apprentissage d'un réseau de neurones.....                            | 89  |
| <b>Annexe B : Modèles de Markov cachés</b>                                |     |
| B.1 Chaîne de Markov.....   | 93  |
| B.1.1 Définition.....   | 93  |
| B.1.2 Exemple : pièce de monnaie truquée.....                             | 94  |
| B.2 Chaîne de Markov cachée.....  | 95  |
| B.2.1 Exemple : pièce de monnaie truquée.....                             | 95  |
| B.2.2 Définition d'une chaîne de Markov cachée.....                       | 96  |
| B.2.3 Calcul factorisé de la probabilité d'une séquence.....              | 98  |
| B.2.4 Autres résultats intéressants.....                                  | 100 |
| B.2.5 Retour à l'exemple.....   | 101 |
| B.2.6 Introduction d'un état d'entrée et d'un état de sortie.....         | 101 |

|   |     |
|---|-----|
| B.2.7 Représentation d'une chaîne de Markov sous forme de graphe..... | 106 |
| B.3 Algorithme du meilleur chemin : algorithme de Viterbi.....        | 107 |
| B.4 Apprentissage d'une chaîne de Markov cachée.....                  | 110 |
| <b>Bibliographie</b>  |     |

## ملخص

اللغة العربية لغة غنية جدا من حيث عدد الحروف و تغير شكل الحرف حسب موضعه في الكلمة. العمل المقدم في هذه المذكرة يدخل ضمن النظام العام للمعرفة الاتوماتيكية للكتابة العربية المكتوبة باليد. هذا الميدان صعب جدا من جهة بسبب نقص قاعدة المعطيات للحروف و من جهة أخرى الحروف المتداخلة في وسط الكلمة خصائصها تتغير حسب نوع الخط و الكاتب مما يؤدي اتوماتيكيا إلى تعقيد في عدة أنظمة التي تؤدي إلى المعرفة. معرفة الحروف و الكلمات مشكلة معقد من جهة الكتابة الغير الثابتة و من جهة الشكل و نوع الخط. عدة عوامل تسمح بالحكم على تعقيد المشكل. نعرض في هذه المذكرة تجسيد جهاز مختلط يعتمد على نظام ماركوف المخفي و نظام عمل شبكات العصبونات لقراءة و معرفة الكلمات العربية.

## **RESUME**

Ces travaux s'inscrivent dans le cadre de la reconnaissance de l'écriture manuscrite dite hors ligne. Celle-ci consiste à déchiffrer des mots cursifs présents dans une image. Il n'existe pas encore de solution satisfaisante à ce problème si sa résolution est envisagée de telle manière qu'elle soit valable pour toute sorte de documents. Néanmoins, lorsque ceux-ci sont d'un type précis comme un chèque ou un formulaire, le contexte permet de façonner des contraintes ou des restrictions qui limitent la recherche et autorisent des solutions performantes. Il est plus facile de reconnaître un mot si celui-ci est censé être un nombre dans le cas d'un chèque ou un prénom dans le cas d'un formulaire. Plus généralement, le problème abordé par ces travaux est la reconnaissance de mots cursifs appartenant à un vocabulaire donné.

Le schéma principal des systèmes de reconnaissance de mots cursifs s'articule autour de deux étapes. Il est tout d'abord nécessaire de prétraiter l'image originale du document de manière à en extraire le mot à reconnaître puis à le segmenter en lettres ou morceaux de lettres appelés graphèmes. Ce résultat est ensuite traité à l'aide de modèles mathématiques qui analysent la forme de chaque graphème et leur séquence. Cette étape probabiliste intègre le plus souvent des modèles de Markov cachés et constitue la reconnaissance proprement dite.

**Mots Clé:** reconnaissance, écriture manuscrite, Modèles de Markov Cachés, traitements d'images, graphèmes.

# **ABSTRACT**

This work deals with offline handwriting recognition. This task consists in automatically reading cursive words from an image. The existing solutions are not yet good enough if they try to solve any kind of such problems. Nevertheless, when the document obeys a given format such as checks or formularies, the context allows us to build an application which gets satisfactory results. Obviously, it is easier to recognize a word if it is known to be a number written on a check, or a first name written on a formulary. The studied problem in this document is the recognition of words which are known to belong to a given vocabulary.

Most of recognition systems include two main steps. First of all, the image source needs to be treated in order to extract the word which has to be recognized and to segment it into letters or pieces of letter which are called graphemes. Then, this result is processed by mathematic models which analyse the grapheme shapes and their sequence. This probabilistic step often includes hidden Markov models and represents the recognition in a strict way.

This thesis describes a hidden Markov model using grapheme neural networks approach designed to recognize off-line unconstrained Arabic handwritten words. After pre-processing, a word image is segmented into characters or pseudo-characters called graphemes and represented by a sequence of observations. Each observation consists of a set of global and local features that reflect the geometrical and topological properties of a grapheme accompanied with information concerning its affiliation to one of five predefined groups. Within its group, the classification of a grapheme is done by a neural network using fuzzy class membership rather than crisp class memberships as desired outputs because it results in more useful grapheme recognition modules for handwritten word recognition. The experimental results on a test database are presented to demonstrate the reliability of this approach.

**Key Words:** Arabic handwritten word recognition, Hidden Markov Models, image processing, grapheme.

## Abréviations

|               |  |
|---------------|--|
| <b>AOCR</b>   | Arabic Optical Character Recognition               |
| <b>ASCII</b>  | American Standard Code for Information Interchange |
| <b>ASMO</b>   | Arabic Standard Metrology Organization             |
| <b>BIC</b>    | Bayesian Information Criteria                      |
| <b>CPI</b>    | Character Per Inch                                 |
| <b>CXX</b>    | Composantes connexes                               |
| <b>DIP</b>    | Densité d'Information en Pixels                    |
| <b>DPI</b>    | Dot Per Inch                                       |
| <b>DPW</b>    | Dynamic Plane Warping                              |
| <b>DTW</b>    | Dynamic Time Warping                               |
| <b>FDP</b>    | Fonction de Densité de Probabilité                 |
| <b>HMM</b>    | Hidden Markov Model                                |
| <b>IM</b>     | Information Mutuelle                               |
| <b>KNN</b>    | K Nearest Neighbors                                |
| <b>MAP</b>    | Maximum A posteriori                               |
| <b>MD-HMM</b> | Model Discriminant HMM                             |
| <b>MDI</b>    | Minimum Discrimination Information                 |
| <b>MLE</b>    | Maximum Likelihood Estimation                      |
| <b>MMI</b>    | Maximum Mutuel Information                         |
| <b>MV</b>     | Maximum de vraisemblance                           |
| <b>N-PAWs</b> | Une suite de "N" PAWs                              |
| <b>OCR</b>    | Optical Character Recognition                      |
| <b>PAW</b>    | Peace of Arabic Word                               |
| <b>PD-HMM</b> | Model Path HMM                                     |
| <b>PHMM</b>   | Planar ou Pseudo 2D Hidden Markov Model            |
| <b>PEL</b>    | Probabilité d'Emission Ligne                       |
| <b>SWS</b>    | Sliding Window Segmentation                        |

## Liste des Figures

|             |  |    |
|-------------|--|----|
| Figure 1.1  | Trois styles d'écriture, imprimé bâton, cursif.....  | 01 |
| Figure 1.2  | schéma de l'ensemble du processus de reconnaissance découpé en trois parties : prétraitement de l'image, reconnaissance statistique, décision .....              | 02 |
| Figure 1.3  | Image extraite de [knerr 01] illustrant le découpage de l'image d'un mot par des fenêtres glissantes.....  | 03 |
| Figure 1.4  | Image d'un paragraphe manuscrit.....   | 04 |
| Figure 1.5  | Image des lignes d'un paragraphe figure 1.4 segmentées à partir d'histogramme (partie gauche de l'image).....  | 04 |
| Figure 1.6  | Graphèmes de la troisième ligne de la figure 1.5.....  | 04 |
| Figure 1.7  | Plusieurs manières d'écrire le même mot.....   | 06 |
| Figure 1.8  | Réponse des modèles mathématiques associés aux mots du dictionnaire.....   | 06 |
| Figure 1.9  | Modèle de Markov Caché + classifieur.....  | 07 |
| Figure 1.10 | Image extraite de [Knerr01], elle représente un graphe de segmentation du mot "et", parmi toutes les possibilités, celle en trait gras est la plus probable..... | 09 |
| Figure 1.11 | Modèle de Markov caché + réseau de neurones.....   | 09 |
| Figure 1.12 | Figure extraite de [LeCun98] représentant le réseau baptisé LeNet-5.....   | 10 |
| Figure 2.1  | Caractères dépendant des points diacritiques.....  | 12 |
| Figure 2.2  | Exemple de mot arabe avec trois composantes connexes.....  | 12 |
| Figure 2.3  | Un échantillon de l'écriture arabe illustrant certaines de ses caractéristiques.....   | 14 |
| Figure 2.4  | Différents styles graphiques de l'écriture arabe [ZAH 90].....   | 14 |
| Figure 2.5  | Un modèle général pour les systèmes AOTR.....  | 16 |
| Figure 2.6  | Un point et ses 8 voisins.....   | 18 |
| Figure 2.7  | Squelette d'un mot binaire :(a) Image binaire du mot ( واحد ), (b)   |    |

|             |   |    |
|-------------|---|----|
|             | Squelette de (a).....   | 19 |
| Figure 3.1  | Les objectifs et les étapes de l'extraction des primitives.....   | 34 |
| Figure 4.1  | Exemple d'un profil des transitions.....  | 37 |
| Figure 4.2  | Principe de l'algorithme de suivi de contour partiel.....   | 37 |
| Figure 4.3  | Le code Freeman.....  | 37 |
| Figure 4.4  | Les points critiques.....   | 39 |
| Figure 4.5  | Principe d'extraction d'un point de départ d'un segment.....  | 40 |
| Figure 4.6  | Exemple des segments générés à partir d'un mot arabe composé de deux caractères.....  | 41 |
| Figure 4.7  | des profils gauches et droits du caractère 3.....   | 43 |
| Figure 4.8  | Exemple des caractéristiques extraites (a) le caractère normalisé(b) la zone des codes chaîne (c) la direction d'histogramme.....           | 48 |
| Figure 4.9  | Projections du caractère THA (a) verticale (b) horizontale.....   | 48 |
| Figure 5.1  | Les différents groupes de graphèmes.....  | 52 |
| Figure 5.2  | Exemple d'ambiguïté entre des mots manuscrits.....  | 52 |
| Figure 5.3  | Quelques graphèmes du groupe GR3 et leurs degrés d'appartenance. Les vraies classes de ces graphème sont respectivement « و » et « م »..... | 54 |
| Figure 5.4  | Des échantillons typiques de l'ensemble d'apprentissage des graphèmes.....  | 54 |
| Figure 5.5  | Principe de calcul du score de reconnaissance par programmation dynamique.....  | 57 |
| Figure 5.6  | (a) Exemple de mots manuscrits correctement reconnues. (b) Exemple de mots manuscrits non reconnus.....                                     | 60 |
| Figure 5.7  | Les différentes étapes de traitement .....  | 60 |
| Figure 5.8  | <b>(b)</b> Evolution du logarithme de la vraisemblance lors de l'apprentissage du modèle HMM du mot (a).....                                | 61 |
| Figure 5.9  | Schéma synoptique d'un système de reconnaissance des chiffres manuscrits.....   | 63 |
| Figure 5.10 | le différent pré traitements Binarisation, filtrage, normalisation, contour et squelettisation.....   | 65 |
| Figure 5.11 | Profil gauche et droit du chiffre 8.....  | 66 |
| Figure 5.12 | profil haut et bas du chiffre 5.....  | 67 |
| Figure 5.13 | apprentissage du réseau de neurone.....   | 68 |

|             |   |     |
|-------------|---|-----|
| Figure 5.14 | L'effet du nombre de neurones dans la couche cachée sur le taux de reconnaissance.....                                    | 70  |
| Figure 5.15 | Variation du taux de reconnaissance selon les valeurs de l'erreur..   | 72  |
| Figure 5.16 | La différence morphologique entre les chiffres de même classe.....  | 73  |
| Figure A.1  | Un neurone.....   | 77  |
| Figure A.2  | Modèle du perceptron multi-couche.....  | 79  |
| Figure A.3  | Régression linéaire de $Y = \frac{3}{2}X^2 - X + \frac{1}{4} + \varepsilon$ .....   | 81  |
| Figure A.4  | Régression non linéaire (1 neurone sur la couche cachée) de $Y = \frac{3}{2}X^2 - X + \frac{1}{4} + \varepsilon$ .....    | 81  |
| Figure A.5  | Régression non linéaire (2 neurones sur la couche cachée) de $Y = \frac{3}{2}X^2 - X + \frac{1}{4} + \varepsilon$ .....   | 82  |
| Figure A.6  | Régression non linéaire (100 neurones sur la couche cachée) de $Y = \frac{3}{2}X^2 - X + \frac{1}{4} + \varepsilon$ ..... | 82  |
| Figure A.7  | Classification d'un plan en deux demi-plans.....  | 84  |
| Figure A.8  | Classification rendue par un réseau de 30 neurones.....   | 84  |
| Figure A.9  | Minimisation par descente de gradient.....  | 87  |
| Figure A.10 | Gradient et gradient conjugué sur une ligne de niveau de la fonction $G(x, y) = 3x^2 + y^2$ .....                         | 91  |
| Figure A.11 | Exemple de minima locaux.....   | 92  |
| Figure B.1  | Représentation d'une chaîne de Markov sous forme de graphe.....   | 94  |
| Figure B.2  | Représentation d'une chaîne de Markov sous forme de graphe.....   | 96  |
| Figure B.3  | Modèles optimaux pour la lettre "M" avec et sans état de sortie.....  | 102 |
| Figure B.4  | Equivalence entre matrice de transition et graphe de transition pour une chaîne de Markov.....                            | 106 |
| Figure B.5  | Equivalence entre matrice de transition et graphe de transition pour une chaîne de Markov ES.....                         | 107 |
| Figure B.6  | Un mot segmenté en graphèmes.....   | 110 |

## Liste des Tableaux

|             |   |     |
|-------------|---|-----|
| Tableau 2.1 | Caractères de l'alphabet arabe. Le triangle montre les caractères qui ne peuvent pas être attachés à leur successeur dans le mot.....   | 13  |
| Tableau 2.2 | Caractéristiques et performance de certains systèmes OCR arabe (TC : Taux de reconnaissance de caractères; TM : Taux de reconnaissance de mot; CPM : Caractères par minute; MPM : Mots par minute)..... | 26  |
| Tableau 5.1 | Résultats d'apprentissage pour les réseaux " CRISP" et " Flous " aux niveaux des graphèmes .....  | 61  |
| Tableau 5.2 | Les paramètres du réseau de neurones.....   | 72  |
| Tableau 5.3 | Les résultats du test par classe.....   | 73  |
| Tableau 5.4 | Les différents cas de confusion entre classes.....  | 74  |
| Tableau A.1 | Fonctions de transfert usuelles.....  | 78  |
| Tableau B.1 | Connexions non nulles dans les modèles de reconnaissance de lettres.....  | 108 |

## **Introduction Générale**

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

Quelques articles plus loin, la reconnaissance de l'écriture manuscrite apparaît comme un sujet de recherche toujours vivace et suffisamment vaste pour que très peu d'articles envisagent de décrire un tel système dans sa globalité. De l'image au résultat, deux grandes étapes se succèdent dont la première est une transcription de l'image dans une forme faisant intervenir des espaces vectoriels. Celle-ci fait intervenir de nombreux traitements d'images selon la qualité du document initial vient ensuite une modélisation plus formelle qui constitue la reconnaissance proprement dite, on y croise le terme apprentissage cher à l'intelligence artificielle qui équivaut souvent à une optimisation de fonctions aux nombreux coefficients.

La reconnaissance de l'écriture manuscrite est le vieux rêve de tous ceux qui ont eu besoin d'entrer 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.

Ces applications montrent clairement les spécificités du domaine de la reconnaissance de l'écriture manuscrite par rapport à celui de la reconnaissance optique des caractères (OCR : Optical Character Recognition) qui concerne les caractères imprimés ou dactylographiés. Il est nécessaire de distinguer également la reconnaissance en ligne (on-line) de l'écriture manuscrite, qui relève plutôt de l'interfaçage entre l'homme et l'ordinateur (un stylo spécial est connecté à la machine et ne fonctionne que sur une tablette sensible), de la reconnaissance hors ligne (off-line). Seule la reconnaissance hors ligne sera considérée dans ce travail.

Quelques systèmes de reconnaissance de l'écriture manuscrite ont été réalisés et sont opérationnels à ce jour. Cependant, ils sont spécifiques à un domaine précis et sont encore limités. Par exemple, en ce qui concerne la poste, la reconnaissance de l'écriture manuscrite, contrairement à celle des caractères imprimés, se limite au code postal en chiffres ainsi qu'à la ville en caractères majuscules.

La reconnaissance universelle de l'écriture manuscrite par l'ordinateur est du domaine de la fiction pour quelques années encore. Tous les chercheurs sont confrontés à un problème difficile et incontournable, celui de la segmentation. La segmentation fait partie du processus de prétraitement et d'extraction de l'information, qui est un préalable à toute reconnaissance. Nous y avons consacré près de la moitié de notre travail.

Les apports spécifiques de ce travail peuvent être synthétisés en évoquant la situation en début et en fin de thèse :

Nous présenterons au chapitre 1 la reconnaissance de l'écriture : problèmes et solutions existantes. Le chapitre 2 décrira, la reconnaissance automatique de l'écriture Arabe. Au chapitre 3 nous détaillerons les méthodes et outils actuels de la reconnaissance des textes manuscrits. Le 4<sup>ème</sup> chapitre extraction des paramètres et méthode de reconnaissance et Le dernier chapitre est consacré aux résultats ; et vous trouvez en annexe les réseaux de neurones et les méthodes de modèle de Markov cachée.

# La reconnaissance de l'écriture: problèmes et solutions existantes

## 1.3 Vue d'ensemble

### 1.3.1 En Ligne, Hors Ligne

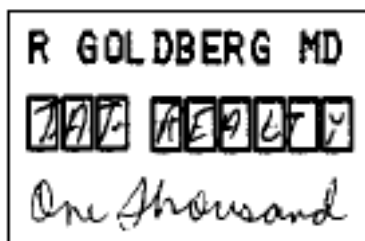
On a coutume de distinguer deux parties dans le domaine de la reconnaissance de l'écriture manuscrite, les reconnaissances en-ligne et hors-ligne. La reconnaissance dite en-ligne s'effectue en même temps que les mots sont écrits, elle concerne les nombreux objets électroniques de poche permettant de saisir du texte sans clavier. La reconnaissance dite hors-ligne concerne tout document déjà écrit comme des formulaires, des livres, des chèques.

La reconnaissance en-ligne commence à apparaître au travers des annuaires portatifs où la saisie s'effectue en majuscule et lettre par lettre. Elle utilise un stylo et un mécanisme de repérage qui mémorise le tracé. Privée de ces informations, la reconnaissance hors-ligne est plus difficile, elle se retrouve souvent cantonnée, à des problématiques très précises telles que la lecture d'adresses postales, de montants littéraux de chèques. La gamme de ces problèmes s'étend au fur et à mesure que la puissance des ordinateurs s'accroît.

### 1.3.2 Styles d'écriture

La difficulté de la reconnaissance est en partie liée au style d'écriture, plus l'écriture est lisible et régulière, plus la résolution est facile. Cette constatation paraît évidente mais si un lecteur humain s'en soucie rarement, les performances obtenues par des logiciels de reconnaissance varient beaucoup avec la clarté des images fournies (lisibilité, bonne résolution de l'image, lignes d'un paragraphe bien espacées,...). On distingue trois styles d'écriture classés par ordre croissant de la difficulté (figure 1.1) bien qu'ils soient parfois emmêlés :

- ✚ L'écriture imprimée.
- ✚ L'écriture manuscrite mais en capitales d'imprimerie ou caractères bâtons.
- ✚ L'écriture manuscrite cursive.



**Fig. 1.1** : Trois styles d'écriture, imprimé bâton, cursif.

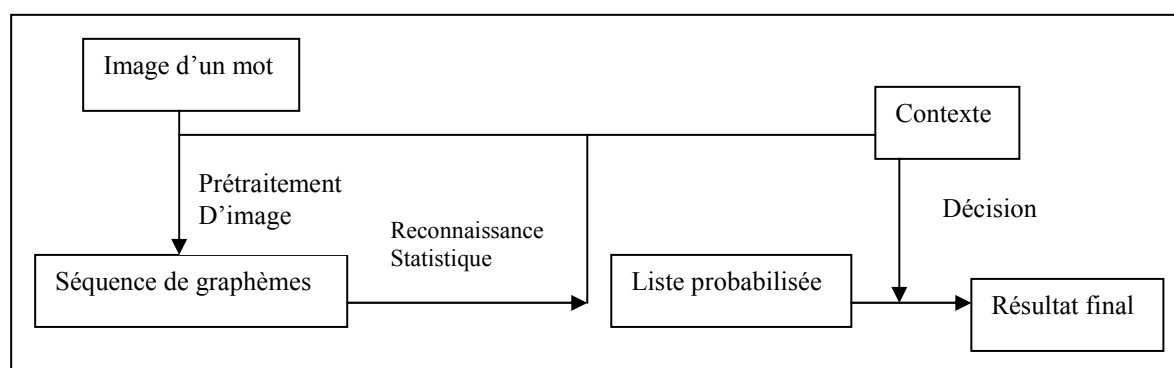
L'écriture imprimée est un problème pour lequel les solutions existantes sont satisfaisantes. Elles commencent à être accessibles aux particuliers puisqu'elles accompagnent fréquemment les logiciels fournis avec les scanners. Il existe par exemple des logiciels de lecture automatique de carte de visite. L'écriture bâton concerne principalement les formulaires renseignés manuellement comme les

feuilles de maladie, les questionnaires à choix multiples. Contrairement à l'écriture imprimée, l'objectif n'est pas de tout décrypter mais au moins d'en traiter automatiquement une bonne partie avec un faible taux d'erreur tandis que les documents mal écrits seront toujours au soin d'un opérateur de saisie. En revanche, les applications traitant la lecture de l'écriture manuscrite sont peu nombreuses. Il s'agit souvent de problèmes précis et réduits comme la reconnaissance du montant littéral d'un chèque, d'une date, quelques champs d'un formulaire.

### 1.3.3 De l'image Au Résultat

Le processus de reconnaissance part d'une image et aboutit à une liste de propositions accompagnées d'une probabilité, ce long processus peut être découpé en trois parties illustré par la figure 1.2 :

- ✚ Prétraitement de l'image.
- ✚ Reconnaissance statistique.
- ✚ Décision.



**Fig. 1.2 :** schéma de l'ensemble du processus de reconnaissance découpé en trois parties : prétraitement de l'image, reconnaissance statistique, décision.

Le terme *contexte* regroupe toutes les informations inhérentes au problème à résoudre comme la langue, le type de document à traiter, le type d'information à reconnaître (nom, prénom, montant,...).

La chaîne des *prétraitements de l'image* comprend essentiellement une segmentation en graphèmes, celle-ci consiste à scinder un problème complexe en plusieurs petits problèmes plus simples.

La *reconnaissance statistique* est centrée autour d'un type de modèle probabiliste adapté à la séquence de graphèmes obtenue à l'étape précédente. Les mieux adaptés sont les modèles de Markov cachés car ils associent séquence et forme des graphèmes.

Enfin, l'étape de *décision* permet d'affiner les résultats de la reconnaissance statistique en associant plusieurs reconnaisseurs en tenant compte d'un contexte propre à l'expérience comme le code postal pour la lecture d'une adresse, ce qui permet d'avoir un *a priori* sur la ville à décrypter. La reconnaissance statistique est fortement dépendante de la langue alors que les prétraitements sont plutôt liés au type de document exploité, enveloppe, chèques, formulaires.

### 1.3.4 Constat et Limites

Les recherches effectuées dans les années précédentes ont aboutit aux modèles de Markov cachés qui sont encore à l'heure actuelle la modélisation la mieux adaptée. Leurs performances se sont d'ailleurs concrétisées par la

commercialisation de produits offrant de bonnes performances mais dans des domaines précis comme la reconnaissance du montant des chèques ou la lecture d'adresses postales. Ces deux aspects -nombreuses étapes, domaine précis- expliquent pourquoi il est difficile de comparer différents systèmes de reconnaissance. Il est donc impossible de choisir une modélisation plutôt qu'une autre lorsqu'elles n'utilisent pas les mêmes prétraitements. La reconnaissance est aussi améliorée par le contexte de l'expérience qui nuit en même temps à une comparaison fiable de deux systèmes puisqu'ils sont utilisés dans des conditions d'expériences différentes. Dans le cas des chèques, montants littéral et numérique sont mis en correspondances, dans le cas d'une adresse, c'est le cas pour la ville et le code postal.

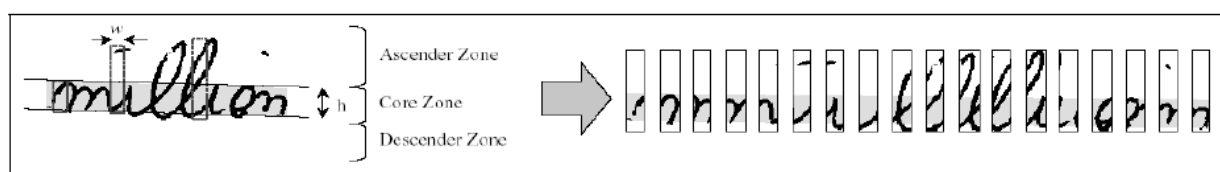
La reconnaissance de l'écriture manuscrite a donc beaucoup suscité l'intérêt des chercheurs dans les années 1990 à 2000. Les différentes solutions proposées ont difficilement pu être comparées puisque apprises sur des données rarement identiques dans des contextes rarement semblables.

Elles proposent malgré tout des solutions exploitables industriellement qui permettent d'envisager la reconnaissance d'un mot manuscrit dans un vocabulaire réduit comme un problème quasiment résolu. Toutefois, ces mêmes modèles ont montré leur limite appliqués à des problèmes moins contraints, où le contexte est moins important.

## 1.4 Les prétraitements D'image

Les articles relatifs à cette partie sont moins nombreux. Les algorithmes utilisés sont plus le résultat d'expériences, d'une succession d'heuristiques, plutôt que le corollaire d'une modélisation théorique de l'image.

L'article [Simon 92] s'intéresse à une segmentation d'un mot à partir du squelette de l'image. Le résultat souhaité est une segmentation proche des caractères [lecolinet 96] le résultat peut être simple et aboutit à des morceaux de petites tailles qui sont plus difficiles à traiter par la suite de la reconnaissance statistique. La méthode des fenêtres glissantes (voir [knerr 01]), (figure 1.3) fait partie de cette catégorie. L'image d'un mot est simplement découpée en bandelettes verticales sans relation avec les lettres. La reconnaissance est dévolue à des modèles mathématiques complexes.



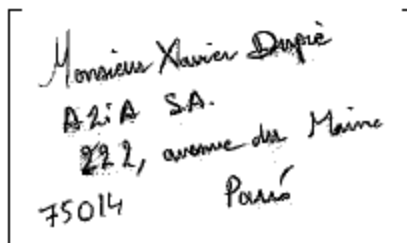
**Fig. 1.3** : Image extraite de [knerr 01] illustrant le découpage de l'image d'un mot par des fenêtres glissantes.

### 1.2.1 Graphèmes :

Le sens de lecture de gauche à droite apparente la reconnaissance de l'écriture à la reconnaissance de la parole. L'analogie n'est valable que si l'image en deux dimensions d'un paragraphe (figure 1.4) est décomposée d'abord en lignes (figure 1.5) puis en caractères ou demi-caractères de manière à retrouver cet ordonnancement de gauche à droite comparable au découpage temporel inhérent à la reconnaissance vocale (voir figure 1.6).

Plutôt d'utiliser des fenêtres glissantes, les premières segmentations élaborées ont tout de suite cherché à isoler les lettres. Ce résultat difficile à obtenir réduit par la suite la taille des modèles de reconnaissance des caractères alors que les fenêtres glissantes aboutissent à des images de caractères décorées de bouts de

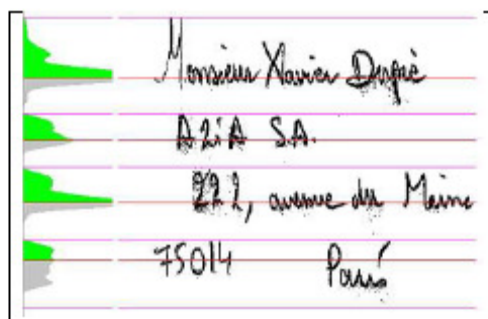
caractères voisins qui s'apparentent à du bruit. Cette plus grande variabilité mène à des modèles de reconnaissance plus consistants. Cette différence n'est plus importante aujourd'hui mais l'était il y a dix ans lorsque les ordinateurs étaient limités en mémoire.



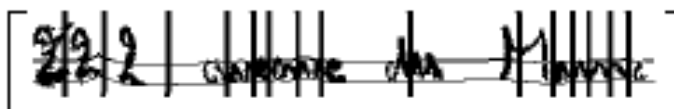
**Fig. 1.4 :** Image d'un paragraphe manuscrit

Les segmentations en lignes et en mots peuvent être réalisées à partir d'histogramme (figure 1.5) sommant les pixels noirs de l'image dans une direction parallèle à celle de segmentation.

Les pics et les creux de l'histogramme permettent de déterminer sans trop d'erreurs le bon découpage. Comme les histogrammes obtenus sont lissés avant leur exploitation, les résultats sont assez stables à moins que l'image ne contienne des traits parallèles à la direction de segmentation comme un mot souligné. C'est pourquoi l'image est d'abord nettoyée, débarrassée de pixels isolés et grands traits horizontaux ou verticaux susceptibles de n'appartenir à aucune lettre. Ce nettoyage est généralement fait à partir d'une squelettisation.



**Fig. 1.5 :** Image des lignes d'un paragraphe figure 1.4 segmentées à partir d'histogramme (partie gauche de l'image).



**Fig. 1.6 :** Graphèmes de la troisième ligne de la figure 1.5.

Malheureusement, cette méthode simple n'est pas applicable à une segmentation en caractères. Comme le montre la figure 1.3, les lettres sont souvent penchées et se prêtent mal à un découpage vertical.

Les graphèmes représentent donc une meilleure segmentation puisqu'elle est plus proche des caractères, elle est aussi plus sensible au bruit car elle est parfois basée sur une squelettisation. Le squelette est une image plus facile à traiter car il

peut être représenté sous forme de graphe. Le découpage est effectué en repérant les ascendants et les descendants sur le squelette.

Ce prétraitement aboutit à une séquence de petites images appelées *graphèmes* qu'il faut décrire d'une manière exploitable par des modèles mathématiques de reconnaissance tels que des réseaux de neurones.

Pourquoi découper ? Pourquoi ne pas conserver l'image du mot dans son intégralité plutôt que de la segmenter avec une méthode instable ? Deux arguments penchent pourtant en sa faveur :

- ✚ Il est plus facile de décrire des morceaux de lettres plutôt que l'image complète du mot.
- ✚ Dans un problème où les mots sont peu nombreux, il est facile de classer telle ou telle image comme étant tel ou tel mot. Lorsque ce nombre de mots est grand (quelques milliers), la classification devient impossible. Il est préférable de classer des objets plus petits comme les lettres ou des morceaux de lettres.

### 1.2.2 Segmentation explicite implicite

La littérature (voir [vinciarelli 02]), distingue parfois deux types de segmentations : implicite et explicite, lorsque la segmentation en graphèmes a pour objectif de découper de l'image d'un mot directement en lettres, celle-ci est explicite. La reconnaissance de ce mot est alors réduite à une reconnaissance de caractères. En revanche, lorsque la segmentation en graphèmes découpe cette image en lettres ou en morceaux de lettres, la reconnaissance statistique doit intégrer le fait qu'une lettre est constituée d'un ou plusieurs morceaux. Par conséquent, la segmentation est dite implicite car les lettres n'apparaissent jamais de manière explicite.

L'article [Sayre 73] souligne le paradoxe de la segmentation implicite qui se résume par cette phrase : une lettre ne peut être segmentée avant d'avoir été reconnue et ne peut être reconnue avant d'avoir été segmentée. Par conséquent, les segmentations explicites sont plutôt un compromis, par assez précises pour définir explicitement des lettres, mais suffisamment pour avoir une association graphème-lettres relativement simple. En règle générale, elles tentent de segmenter l'image d'un mot en morceaux qui sont inclus dans le dessin d'une lettre. Ces segmentations sont souvent regroupées sous le terme *sur segmentation*.

### 1.2.3 Caractéristiques

Etant donné que les graphèmes peuvent avoir des tailles variables, l'étape suivante consiste à décrire ces images par un vecteur de caractéristiques de dimension fixe. Ces nombres sont réels et chacun d'eux est censé décrire un aspect de l'image comme :

- ✚ la taille du graphème (largeur, hauteur),
- ✚ l'aire occupée par les pixels noirs,
- ✚ le barycentre des pixels noirs,
- ✚ l'inclinaison,
- ✚ l'épaisseur moyenne des traits verticaux et horizontaux, ces chiffres peuvent être calculés sur l'image entière ou une des bandes verticales ou horizontales, l'image est découpée en quatre ou cinq, selon la résolution,
- ✚ le nombre moyen de traits verticaux et horizontaux (transition pixel noir - pixel blanc sur une colonne de l'image),
- ✚ la position relative du précédent graphème, du suivant.

En définitive, ce prétraitement transforme l'image d'un mot en une séquence de vecteurs de dimension fixe ou *mot mathématique*. Si cette dimension est notée  $D$ , un mot mathématique est donc :

### Définition 1.2.1 : mot mathématique

On définit un mot mathématique comme une séquence finie de vecteurs de dimension fixe :  $mot \leftrightarrow m \in (R^D)^N$  Où  $D$  est la dimension de l'espace des caractéristiques.

## 1.3 Reconnaissance statistique

### 1.3.1 Classification en mots

Dans le cas du problème de reconnaissance avec dictionnaire, reconnaître un mot écrit sur une image revient à classer le mot mathématique qui lui correspond dans la classe qui regroupe toutes les écritures différentes de ce même mot (figure 1.7). La reconnaissance est un problème de classification.

Pour chaque classe de mots, un modèle mathématique est construit afin de modéliser l'ensemble des mots mathématiques lui correspondant. Ensuite, pour attribuer la bonne classe à un mot mathématique, tous les modèles de mots sont sollicités et celui qui donne la meilleure réponse est considéré comme la bonne réponse (figure 1.8).

L'avantage majeur de cette organisation est le possible ajout de modèles. Par exemple, en ce qui concerne la reconnaissance du montant littéral des chèques, le passage à dinars algériens consiste à ajouter le mot DA à l'ensemble des mots possibles pour écrire un montant.

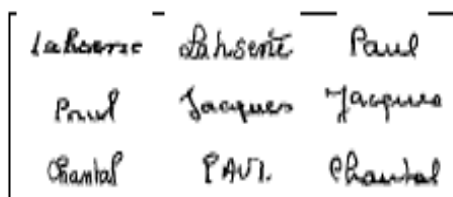


Fig. 1.7: Plusieurs manières d'écrire le même mot.

|       |       |       |       |
|-------|-------|-------|-------|
| MARIO | 0,782 | MARIA | 0,004 |
| MAIRE | 0,108 | FRANC | 0,002 |
| MARIE | 0,076 | FRANS | 0,002 |
| MAUD  | 0,023 | HANS  | 0,001 |
| ...   | ...   | ...   | ...   |

Fig. 1.8: Réponse des modèles mathématiques associés aux mots du dictionnaire.

La construction des modèles de mots dépend du problème à résoudre. Toujours dans le cas de la reconnaissance avec dictionnaire, si le dictionnaire est petit (pas plus d'une centaine de mots), un modèle par mot sera construit, c'est-à-dire qu'il aura été appris pour reconnaître ce mot.

Dans le cas d'un grand dictionnaire (quelques milliers de mots), on préférera obtenir ce modèle de mot par l'assemblage de modèles de lettres, qui auront appris à reconnaître spécifiquement les lettres. C'est dans cette optique qu'une bonne segmentation en graphèmes est importante. Si un graphème représente plus d'une lettre, la reconnaissance aura toutes les chances d'être faussée.

### 1.3.2 Séquence et forme

La définition 1.2.1 définit un mot mathématique comme une séquence de vecteurs, ceci signifie qu'à un mot correspond des mots mathématiques de

longueurs différentes et de vecteurs différents mais de dimension fixe. Le paragraphe 1.2.1 a déjà répondu à propos de l'intérêt de conserver un paramètre variable dans la description de l'image d'un mot. Cette description fait donc intervenir deux aspects :

1. une séquence de longueur variable,
2. la forme des graphèmes car cette séquence est composée de vecteurs de dimension fixe décrivant la forme de chacun des graphèmes qui la composent.

Dans les modèles présentés dans ce document, beaucoup traitent ces deux aspects séparément. La forme, le plus simple des deux aspects puisque de dimension fixe, est traitée par un classifieur qui peut être un réseau de neurones, une machine à support vectoriel plus connu sous sa dénomination anglaise Support Vector Machine (SVM), un arbre de décision...

La séquence est principalement modélisée par des modèles de Markov cachés. Ceux-ci ont été développés par Baum ([Baum 68], [Baum 72]) et s'inscrivent dans un problème plus général décrit par Dempster ([Dempster 77]) qui expose pour la première fois l'algorithme EM (expectation-maximisation), permettant d'estimer les paramètres d'un modèle prenant en compte des observations cachées. Les modèles de Markov cachés en sont un cas particulier. Pour terminer la liste des articles cités chaque fois, il faut mentionner une excellente introduction ([Rabiner 86]) dont l'étude est approfondie par [Levinson 83].

Toutefois, ces articles ne présentent que des chaînes de Markov cachées discrètes qu'il faut adapter afin qu'elles prennent en compte l'aspect forme qui, lui, est continu. Un modèle intégrant le couple séquence forme de manière séparée constitue un modèle appelé *hybride* ou *semi-continu*.

### **1.3.3 Choix d'une modélisation**

Choisir une modélisation mathématique pour un problème impose de faire certaines hypothèses sur les données à représenter (indépendance temporelle, la loi probabiliste qu'elles suivent, ...). Une fois celles-ci converties en un modèle, il peut contenir plus ou moins de degrés de liberté. Plus il en a, plus il peut s'adapter facilement aux données à modéliser et plus il s'adapte difficilement à de nouvelles données.

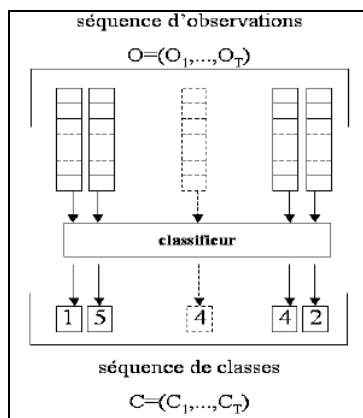
Les hypothèses doivent donc être les mieux adaptées, elles conditionnent la structure du modèle (modèles de Markov cachés, réseaux de neurones, ...). Il suffit de déterminer le choix du degré de liberté (ou nombre de coefficients) pour le modèle sélectionné. Le choix des hypothèses est toujours le plus important car il détermine les algorithmes d'estimation des paramètres et ceux de leur utilisation. Diverses modélisations sont présentées par la suite.

## **1.4 Modélisation**

### **1.4.1 Modèles de Markov cachés hybrides et classifieur quelconque**

Cette modélisation est utilisée depuis quelques années pour la reconnaissance de textes en écriture romaine et est plus récemment adaptée pour la reconnaissance de textes Arabes ([Khorsheed 03]). C'est le modèle le plus simple, à chaque vecteur de caractéristiques est associée une classe de manière à transformer la séquence de vecteurs d'observations continues en séquence d'observations discrètes (figure 1.9). Ce modèle convertit ensuite cette séquence en une probabilité : celle que le modèle émette cette séquence discrète.

Ce processus est répété pour chaque modèle de mot, le mot reconnu est celui ayant obtenu la meilleure probabilité d'émission.



**Fig. 1.9 :** *Modèle de Markov Caché + classifieur.*

Les classifieurs peuvent être de toutes sortes (séparateurs linéaires, centres mobiles, réseau de neurones, réseau de Kohonen, arbre de décision, SVM...). Le choix de ces modèles est influencé par le fait que les observations sont annotées (leur classe est connue) ou non. Si elles sont annotées, les graphèmes sont de taille suffisante pour être interprétés (lettres ou parties de lettres), dans ce cas, cette interprétation ou classification donnée est alors apprise par le classifieur. Cette méthode possède l'avantage de fixer le nombre de classes mais aussi l'inconvénient de parfois générer des ambiguïtés car certaines différences apparaissant sur l'image peuvent disparaître dans le vecteur de caractéristiques. De plus, au niveau de l'image, la lettre "u" divisée en deux parties ressemble fortement à deux lettres "i". Si elles ne sont pas annotées, la classification est automatique (centres mobiles par exemple) mais il reste à déterminer le nombre de classes idéal. Cette partie peut faire l'objet d'une étude statistique préalable.

Soit un mot mathématique noté  $O = (O_1, \dots, O_T)$  et la séquence des classes d'observations associée à ce mot  $C = (C_1, \dots, C_T)$  (annotation), on définit la probabilité que le modèle  $M$  émette la séquence  $O$  par :

$$P(O/M) = \sum P\left(O, q_1, \dots, q_t \middle/ M\right)$$

$$= \sum \left[ P\left(q_1 \middle/ M\right) P\left(C_1 \middle/ q_1, M\right) \prod_{t=2}^T P\left(q_t \middle/ q_{t-1}, M\right) P\left(C_2 \middle/ q_t, M\right) \right]$$

1.1

Où  $(q_1, \dots, q_T)$  est une séquence d'état du modèle.

L'équation (1.1) découle des hypothèses inhérentes aux chaînes de Markov cachées d'ordre un. Cette modélisation fait intervenir deux décisions:

1. La première est prise par le classifieur qui attribue une classe à chaque graphème.
2. La seconde est prise par le module de décision qui décide de lire dans l'image le mot dont le modèle associé est le plus probable.

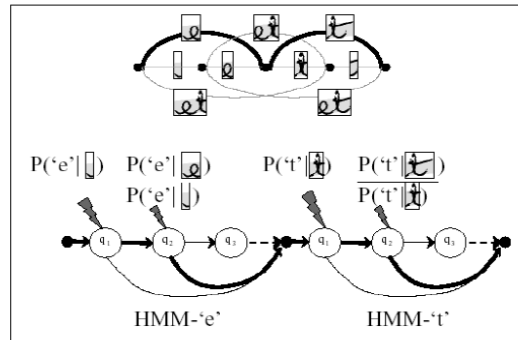
La seconde décision est incontournable mais la première peut l'être en adoptant une modélisation plus fine comme celle des paragraphes 1.4.2 et 1.4.3.

Ces modèles sont utilisés par [Knerr 01] qui compare leurs résultats à ceux de modèles de Markov cachés discrets. L'intérêt de cet article porte sur l'utilisation de leurs propriétés lors de la segmentation d'une image d'un mot en lettres (figure 1.10). Plusieurs topologies de chaînes de Markov sont exposées accompagnées de l'idée qui a présidé à leur élaboration. Ces modèles sont aussi utilisés dans la thèse [Augustin 01] qui aborde le problème de la recherche de la meilleure topologie de la

chaîne de Markov. Ces modèles font encore l'objet de recherche, [Koerich02b] présente leur utilisation pour de grands vocabulaires.

### 1.4.2 Modèles de Markov cachés hybrides et lois gaussiennes

Au lieu de classer les observations afin de les rendre discrètes, celles-ci sont supposées suivre une loi normale, différente pour chaque état du modèle ([Bottou 91]), ou être partagées par plusieurs états à la fois ([Bunke 95]).

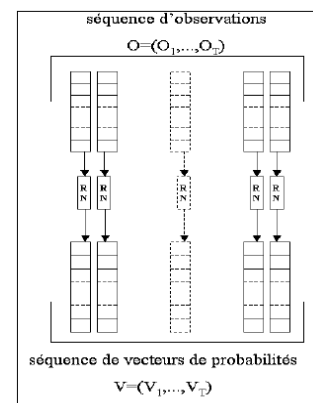


**Fig. 1.10:** Image extraite de [Knerr01], elle représente un graphe de segmentation du mot "et", parmi toutes les possibilités, celle en trait gras est la plus probable.

L'inconvénient de ce modèle est sa gourmandise : pour des vecteurs d'observations de 50 caractéristiques, la matrice carrée  $V_i$  contient 2500 coefficients et ce pour chaque état de la chaîne de Markov. Par conséquent, l'utilisation de ces modèles s'accompagne d'hypothèses simplificatrices :  $V_i$  est souvent supposée diagonale.

### 1.4.3 Modèles de Markov cachés hybrides et réseau de neurones

Le classifieur de la figure 1.10 est remplacé par un réseau de neurones illustré par la figure 1.11. Il associe à une séquence d'observations  $O = (O_1, \dots, O_T)$  une séquence de vecteurs de probabilités  $V = (V_1, \dots, V_T)$ .



**Fig. 1.11:** Modèle de Markov caché + réseau de neurones.

L'avantage de ce modèle par rapport à celui du paragraphe 1.4.1 est qu'une observation n'appartient plus impérativement à telle ou telle classe, le réseau de neurones ne donne que les probabilités d'appartenance, la décision n'est plus aussi brutale. Ne pas prendre de décision est une expression récurrente dans la reconnaissance de l'écriture, un leitmotiv. Une décision brutale implique un seuillage, c'est-à-dire l'utilisation d'une fonction non dérivable et difficile à apprendre.

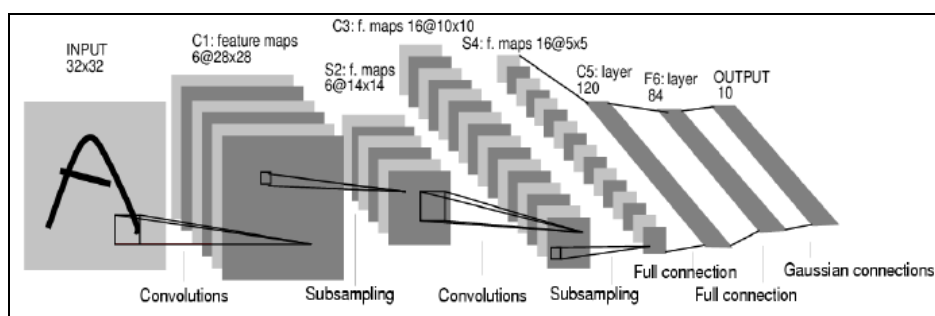
L'estimation du modèle est évidemment plus compliquée. L'initialisation du réseau de neurones s'effectue grâce à un classifieur, l'estimation des coefficients de la chaîne de Markov cachée est identique à celle du modèle 1.10. Une fois ces deux étapes effectuées, l'apprentissage du réseau de neurones est supervisé par la chaîne de Markov. L'apprentissage du modèle global résulte d'une alternance entre apprentissage chaîne de Markov et apprentissage réseau de neurones, jusqu'à convergence de l'un et de l'autre.

Les modèles utilisés chez la société A2iA contiennent entre 20000 et 70000 coefficients pour le réseau de neurones, entre 100 et 1000 coefficients pour la chaîne de Markov cachée. Le facteur forme (réseau de neurones) est donc beaucoup plus important que le facteur séquence (chaîne de Markov cachée).

Un apprentissage simultané est envisagé dans [Bengio 92], néanmoins, pour un aussi volumineux système, il est parfois préférable de scinder le problème.

#### 1.4.4 Réseau de neurones incluant des prétraitements d'images

Ce système présenté dans [LeCun 98] est un autre reconnaiseur de caractères basé sur un réseau de neurones. Il diffère des autres par l'incorporation de plusieurs couches de neurones dévolues au traitement d'image (figure 1.12). Son apprentissage est aussi différent des autres puisqu'il est appris à partir d'une base d'images de caractères annotés et de transformations de ces images (inclinaison, rotation, bruitage, ...).



**Fig. 1.12:** Figure extraite de [LeCun98] représentant l+e réseau baptisé LeNet-5

Un réseau de neurones à convolution pour la reconnaissance de caractères, tous les plans de la couche  $C_1$  partagent les mêmes coefficients, les sorties de cette couche forment un ensemble de caractéristiques pour la couche suivante puisque directement extraites de l'image.

L'ensemble présenté par [LeCun98] est un reconnaiseur de caractères mais il pourrait être adapté pour la classification des graphèmes.

# Reconnaissance automatique de l'écriture Arabe

## 2.1 Introduction

L'objectif de la reconnaissance de l'écriture est de transformer un texte en une représentation compréhensible et traitable par une machine. Plusieurs facteurs interviennent : la subjectivité de l'écriture, la diversité des polices de caractères, la grande variabilité des styles d'écriture (gras, italique, souligné, etc.), la complexité de mises en page, rendent cette tâche bien complexe.

Le problème de la reconnaissance de l'écriture a été abordé, depuis plus de la moitié d'un siècle, il a fait l'objet de recherches très intensives, et de nombreux articles ont été publiés à ce sujet [GOV 90, TAP 90, MOR 92, ALB1 95]. Ces recherches devraient répondre à des besoins énormes en matière d'automatisation d'offices, du tri postal, de la lecture optique de chèques, la traduction automatique, et l'amélioration d'interfaces homme-machine, ...etc.

Les études dans le domaine de la reconnaissance d'écriture sont largement influencées par le type d'écriture à identifier. De ce fait, la reconnaissance d'un texte imprimé n'est pas comme celle d'un texte manuscrit. Si dans le premier cas, les problèmes ont été relativement résolus (i.e., cas des langues latines), et les études ont donné lieu à des machines commercialisées, la situation est complètement différente en ce qui concerne la reconnaissance des textes manuscrits et spécifiquement cursifs (ex., écriture Arabe).

La reconnaissance de l'écriture Arabe s'intègre dans le cadre général de la reconnaissance de l'écriture cursive, avec des spécificités et des problèmes qui lui sont propres. Les premières recherches dans ce domaine remontent aux années 60. Mais compte tenu des performances matérielles et logicielles limitées obtenues à cette époque, elles n'ont pas abouti à des résultats satisfaisants. Elles connaissent actuellement un regain d'intérêt dû au progrès réalisé dans le domaine de l'informatique, de l'intelligence artificielle et aux développements technologiques récents (scanner, caméra, tablette graphique ...) [ZAH 90].

Contrairement à d'autres systèmes d'écriture (Latine, Chinoise, Japonaise...) peu de travaux ont été menés concernant la reconnaissance de l'écriture Arabe. Ce fait est surprenant, car plus de 300 millions de personnes réparties dans plus de 20 pays pratiquent la langue Arabe, en plus, les caractères Arabes sont utilisés dans l'écriture de plusieurs langues universelles (Persan (Farci), Ourdou (langue nationale du Pakistan) ...).

Le premier travail publié sur la reconnaissance optique de textes arabes AOTR (Arabic Optical Text Recognition), remonte jusqu'à l'année 1975 [ALB1 95].

Depuis les années 80, les travaux de recherche en AOTR, sont accentués, et sont devenus relativement importants durant ces dernières années.

Le retard qu'avait connu le domaine d'AOTR, en comparaison avec d'autres langues, d'après Badr-Al-Badr [ALB1 95], peut être attribué aux :

- ✚ manque de supports adéquats en terme de journaux, livres, congrès, consolidation, en plus du manque d'interactions entre les chercheurs dans ce domaine ;
- ✚ manque d'outils de soutien général tels que : les bases de données de textes arabes, dictionnaires, les outils de programmation et le personnel de soutien ;
- ✚ Le début tardif de la reconnaissance des textes Arabes ;
- ✚ les caractéristiques spéciales de l'écriture Arabe.

## 2.2 Caractéristiques de l'écriture Arabe

Les caractères arabes (voir table 2.1) diffèrent d'autres types de caractères (Latins, Chinois...) par leur propre structure et leur mode de liaison pour former un mot, ce qui rend l'application directe des techniques de reconnaissance développées pour les caractères chinois ou latins, par exemple, une tâche délicate pour leur reconnaissance.

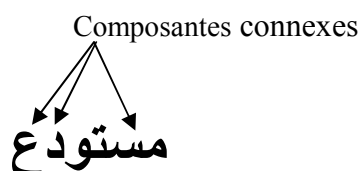
L'écriture Arabe se caractérise par :

- ✚ Les caractères qui s'écrivent de droite à gauche.
- ✚ L'Arabe a 28 caractères de base et il n'y a pas de majuscules.
- ✚ L'écriture Arabe est par nature cursive, qu'elle soit imprimée ou manuscrite.
- ✚ Selon sa position dans le mot (en tête, à l'intérieur, à la fin ou isolé), un même caractère Arabe peut avoir jusqu'à quatre formes différentes, ce qui multiplie le nombre de modèles (table 2.1).
- ✚ Certains caractères ont le même corps, mais la présence et la position d'un point ou d'un groupe de points, sont les traits déterminants pour distinguer ces caractères. La figure 2.1 montre quelques caractères qui ont le même corps qui se différencient seulement par la présence et le nombre de points au-dessus ou au-dessous de leurs corps.

ب ث ت ن

**Fig 2.1** : Caractères dépendant des points diacritiques.

- ✚ Un caractère Arabe peut contenir un trait vertical (TAA (ط)), un trait oblique (KAF (ك)) ou un zigzag (ALIF (أ)).
- ✚ Les points et le zigzag (Hamza) sont appelés des secondaires.
- ✚ Un mot Arabe se compose généralement d'une ou plusieurs composantes connexes, chacune contient un ou plusieurs caractères (figure 2.2).



**Fig 2.2** : Exemple de mot arabe avec trois composantes connexes.

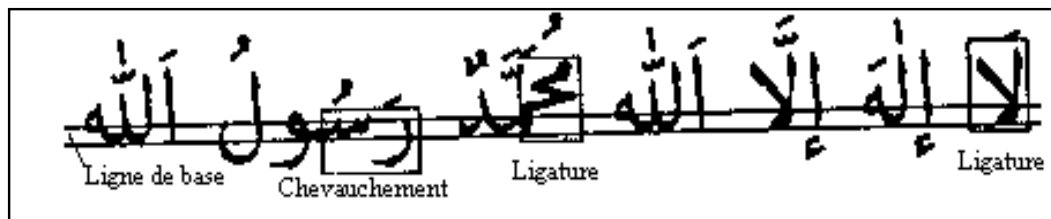
| Caractère. | Au début | Au milieu      | Fin | Isolé | Car.  | Au début | Au milieu      | Fin | Isolé |
|------------|----------|----------------|-----|-------|-------|----------|----------------|-----|-------|
| alif       | أ        | أ <sup>△</sup> | أ   | أ     | dhad  | ض        | ض              | ض   | ض     |
| ba         | ب        | ب              | ب   | ب     | tad   | ط        | ط              | ط   | ط     |
| tâ         | ت        | ت              | ت   | ت     | thad  | ظ        | ظ              | ظ   | ظ     |
| thâ        | ث        | ث              | ث   | ث     | ayn   | ع        | ع              | ع   | ع     |
| jim        | ج        | ج              | ج   | ج     | ghayn | غ        | غ              | غ   | غ     |
| ha         | ح        | ح              | ح   | ح     | fa    | ف        | ف              | ف   | ف     |
| kha        | خ        | خ              | خ   | خ     | kaf   | ق        | ق              | ق   | ق     |
| del        | د        | د <sup>△</sup> | د   | د     | kef   | ك        | ك              | ك   | ك     |
| dhel       | ذ        | ذ <sup>△</sup> | ذ   | ذ     | lam   | ل        | ل              | ل   | ل     |
| ra         | ر        | ر <sup>△</sup> | ر   | ر     | mim   | م        | م              | م   | م     |
| zei        | ز        | ز <sup>△</sup> | ز   | ز     | noun  | ن        | ن              | ن   | ن     |
| sin        | س        | س              | س   | س     | ha    | ه        | ه              | ه   | ه     |
| shin       | ش        | ش              | ش   | ش     | wew   | و        | و <sup>△</sup> | و   | و     |
| sad        | ص        | ص              | ص   | ص     | ya    | ي        | ي              | ي   | ي     |

**Table 2.1** : Caractères de l'alphabet arabe. Le triangle montre les caractères qui ne peuvent pas être attachés à leur successeur dans le mot.

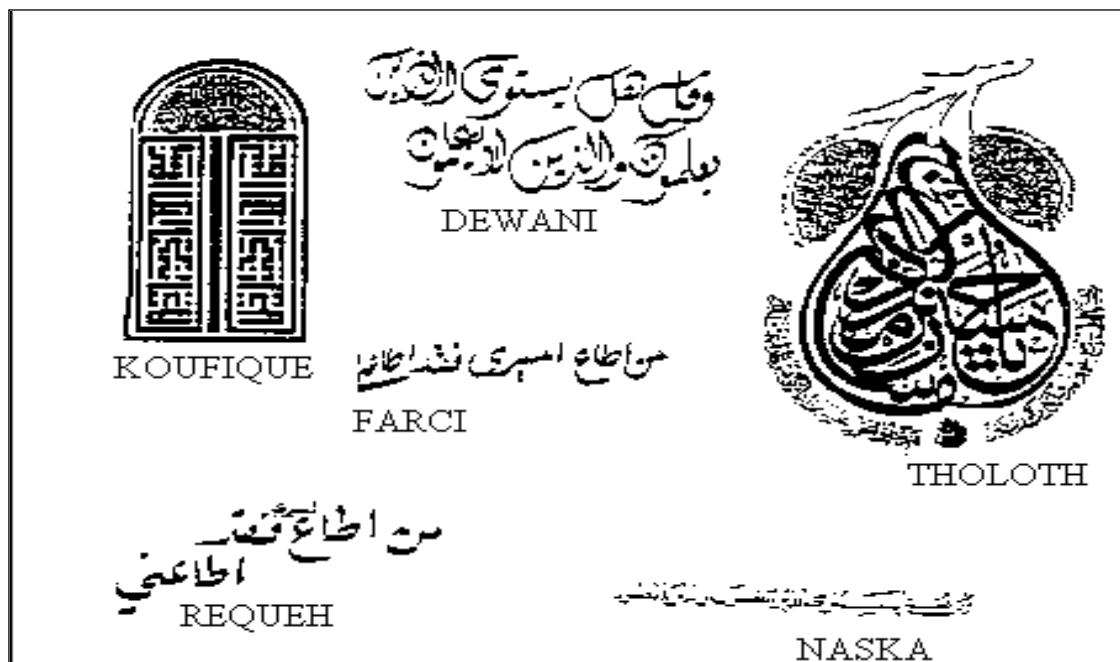
- ✚ La discontinuité entre les groupes de caractères d'un mot, est due aux six caractères de l'alphabet arabe qui ne peuvent pas être attachés à leur successeur dans le mot (table 2.1).
- ✚ Une caractéristique distinguée de l'écriture arabe, est la présence de ce qu'on appelle la ligne de base. Cette dernière est une ligne horizontale qui passe par les portions connexes du texte (i.e., où se trouvent les segments de connexion de caractères). La ligne de base possède un nombre maximum de pixels (noirs) dans le texte (figure 2.3).
- ✚ Certains caractères dans un mot, peuvent se chevaucher verticalement sans contact (figure 2.3).
- ✚ Les caractères arabes ne possédant pas une taille fixe (hauteur et largeur), leur taille varie d'un caractère à un autre et d'une forme à une autre à l'intérieur d'un même caractère (table 2.1).
- ✚ A l'intérieur d'un mot, les caractères peuvent avoir des diacritiques (voyelles courtes). Ces diacritiques sont écrites comme des traits au-dessus de la ligne de base :
- ✚ Fat-hah (َ), Dhammah (ُ), Shaddah (ّ), Maddah (◌◌), Soukoun (◌◌◌) ou au-dessous de la ligne de base comme Kasrah (ِ). En plus de Tanween qui peut être formé en faisant doubler le Fat-hah (◌◌◌), Dhammah (◌◌◌) ou Kasrah (◌◌◌).

Notons qu'une diacritique différente sur un caractère peut changer le sens d'un mot (ex. Drapeau (عَلَم), Science (عِلْم)).

- ✚ Généralement, les textes arabes ne sont pas voyellisés, notamment dans la presse et les livres. Les lecteurs de l'Arabe sont habitués à lire ces textes en déduisant le sens à partir du contexte.
- ✚ Plusieurs caractères peuvent se combiner verticalement pour former une ligature (voir figure 2.3).
- ✚ Comme l'écriture arabe est une écriture calligraphique, six styles graphiques différents sont souvent utilisés : Tholoth, Naska, Requeh, Dewani, Farci, et Koufique. La figure 2.4, montre un exemple de différents styles graphiques de l'écriture arabe.



**Fig 2.3** : Un échantillon de l'écriture arabe illustrant certaines de ses caractéristiques.



**Fig 2.4** : Différents styles graphiques de l'écriture arabe [ZAH 90].

En résumé, les principales propriétés graphiques de l'écriture arabe sont les suivantes:

- 1- L'écriture se fait de droite à gauche, et elle est cursive par nature,
- 2- La liaison entre les caractères se fait par l'intermédiaire de branches de faibles rayons de courbure. Les points de jonction se situent le plus souvent près de la ligne d'écriture,
- 3- La plupart des caractères sont composés de boucles et de courbes, souvent

tracées dans le sens horaire ; quelques-uns comportent des traits verticaux,

4- Le chevauchement de caractères et la présence de ligatures dans un mot, rendent l'utilisation des méthodes de segmentation, caractère par caractère, du mot à reconnaître une tâche très délicate,

5- Une même forme peut exister dans des caractères différents. Par exemple, la forme de  $\text{b} \left( \text{ } \right) \text{e}$  apparaît dans les caractères SSAD, DHAD, TTA, DDHAH, QAF, FA, mais peut apparaître seule comme pour le caractère MEEM.

6- L'Arabe contient 28 caractères de base, dont 16 incluent dans leur forme des points diacritiques qui peuvent être au nombre de 1, 2 ou 3. Ces points font la différence entre les caractères identiques et ils peuvent être situés en haut du caractère de base, en dessous ou au milieu.

7- La segmentation en caractères (ou en parties de caractères) ou encore en pseudo-mots, constitue le problème le plus ardu lié à la reconnaissance de l'écriture Arabe. Les difficultés rencontrées à ce niveau sont de même type que celles affrontées lors de la reconnaissance du Latin manuscrits, mais souvent plus complexes à cause de la diversité des formes du caractère arabe, de la courte liaison qui existe entre deux caractères successifs, de l'allongement des ligatures horizontales et de la présence de ligatures verticales. Tous ces problèmes se trouvent accentués dans le cas du manuscrit à cause de la variabilité du script.

### 2.3. Un modèle général pour la reconnaissance optique de textes Arabes (AOTR)

Le processus de reconnaissance de textes arabes peut être globalement divisé en 6 étapes : (1) Acquisition, (2) Prétraitement, (3) Segmentation, (4) Extraction de caractéristiques, (5) Classification, (6) Post-traitement.

✚ **Acquisition** : Cette opération consiste à capter l'image d'un texte à reconnaître au moyen de capteurs physiques (caméra, scanner...), puis, la convertit en une image numérique sur laquelle un traitement s'effectuera par un ordinateur.

✚ **Prétraitement** : C'est une opération qui renferme toutes les procédures qui appliquent des transformations successives sur l'image captée. Elle prend en entrée une image brute et essaie de la rehausser en réduisant le bruit et les distorsions, afin de simplifier la segmentation, l'extraction de caractéristiques, et par conséquent améliorer le taux de reconnaissance.

✚ **Segmentation** : Comme son nom l'indique, cette opération est appliquée sur l'image pour séparer les différentes parties logiques de celle-ci, comme les textes des graphiques, et extraire les lignes des paragraphes, et les caractères (ou parties de caractères) des mots.

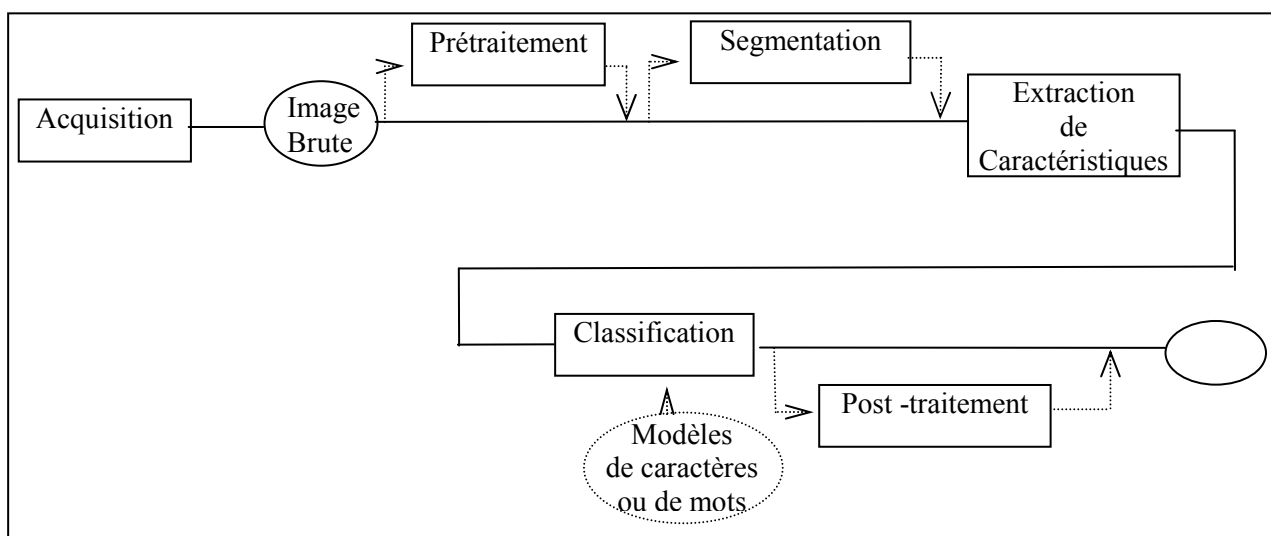
✚ **Extraction de caractéristiques** : Son rôle primordial est limité à l'analyse d'un segment de texte et à la sélection des primitives (traits) pertinentes qui peuvent être utilisées pour identifier sans ambiguïté ce segment de texte. Ensuite, les résultats obtenus dans cette étape seront communiqués sous une forme appropriée à l'étape suivante du processus.

✚ **Classification** : C'est le stade principal de décision dans un système de reconnaissance de texte. A cette étape, les primitives extraites dans l'étape précédente sont utilisées pour identifier le segment de texte selon des règles établies préalablement. Généralement, à ce niveau, on utilise des modèles obtenus dans une phase d'apprentissage (off-line), pour classifier les données de test.

✚ **Post-traitement** : C'est la dernière étape dans un système de reconnaissance de texte. Grâce à l'utilisation des informations d'ordre supérieur (lexicales, syntaxiques, sémantiques, contextuelles...), cette étape peut améliorer le taux de

reconnaissance en raffinant les décisions prises par l'étape précédente. Elle est finalement responsable de la crédibilité de la solution donnée est souvent implémentée comme un ensemble de techniques, qui dépendant du lexique, de la fréquence d'apparition de caractères, et des informations contextuelles.

La figure 2.5 montre le modèle général que nous avons adopté pour les systèmes AOTR. Notons que, les étapes (*prétraitement*, *segmentation*, *post-traitement*) ne sont pas nécessairement exécutées par tous les systèmes AOTR. Certains systèmes supposent l'entrée dénuée de bruit. Aussi, certaines techniques reconnaissent un caractère avant la segmentation, d'autres reconnaissent seulement les caractères isolés. Le post-traitement est abordé uniquement par peu de chercheurs dans ce domaine.



**Fig 2.5** : Un modèle général pour les systèmes AOTR.

### 2.3.1. Acquisition

Un système de reconnaissance peut capter le texte à traiter en deux modes différents : soit en mode en-ligne (**On-line**), soit en mode hors-ligne (**Off-line**).

#### 🌈 Les systèmes de reconnaissance On-line

Pendant que l'utilisateur est en train d'écrire, le système capte les informations temporelles ou dynamiques de l'écriture à travers un dispositif temps réel. Ces informations comprennent le nombre, la durée, et l'ordre de chaque tracé (un tracé est une suite de points entre deux levés de stylet). Typiquement, dans ce type de systèmes, on utilise soit une tablette à digitaliser soit des écrans/digitalizer, qui affichent instantanément ce que l'utilisateur écrit. L'information traitée dans ce cas est mono-dimensionnelle, i.e., un vecteur ordonné de points (X, Y). Les performances d'une tablette graphique sont exprimées en termes de résolution et vitesse d'échantillonnage. Pratiquement, la résolution atteint jusqu'à 200 points par

pouce, tandis que le taux d'échantillonnage atteint jusqu'à 100 points par seconde [ALB1 95].

Les systèmes on-line sont pratiquement limités à la reconnaissance de texte manuscrit. Certains systèmes sont limités à la reconnaissance des caractères isolés [AMI 80, EIW 89], d'autres reconnaissent les mots cursifs [AMI 82, AMI 85].

### ✚ **Les systèmes Off-line**

Contrairement aux systèmes On-line, ces systèmes reconnaissent les textes après que ceux-ci aient écrit ou imprimé. Ces systèmes sont mieux connus sous le nom OCR (Optical Text Recognition), ceci est dû au fait qu'ils emploient des procédés optiques dans l'acquisition (ex., [AMI 86, AMI 96, SOU1 96, BEN 98]).

Dans ce cas, l'information traitée est bi-dimensionnelle (une page de texte représentée par une matrice de pixels).

La reconnaissance Off-line peut être considérée comme le cas le plus général: aucun moyen spécial n'est nécessaire pour l'écriture.

Parmi les systèmes off-line utilisant une caméra vidéo pour l'acquisition, on trouve ceux de Amin et Zahour [AMI 86, ZAH 90].

La majorité des systèmes AOTR off-line, que nous avons examiné utilisent un scanner comme moyen d'acquisition.

Les systèmes off-line [AMI 96, ABU 94, SOU1 96] reconnaissent les caractères isolés. Parmi les systèmes qui reconnaissent les mots manuscrits, on trouve [AME 94, ALM 87, ALY 90, BEN 98]. La plupart des systèmes off-line reconnaissent les mots cursifs imprimés [AMI 86, AMI 89, ELD 90, ELS2 88, BEN 96, BEN 97], d'autres reconnaissent les formules mathématiques imprimées et/ou manuscrites (ex., [AMI 92]). Tandis que [ELS1 88, MAH 94] reconnaissent seulement les caractères imprimés isolés. Le système [GOW 90] est conçu pour la reconnaissance des mots cursifs composés (Typeset).

## 2.3.2. Prétraitement

La fiabilité de la reconnaissance est en corrélation forte avec la qualité du texte à traiter et le manque de bruit. En réalité, les méthodes d'OCR qui fonctionnent bien sur certain type d'images, fonctionnent très mal sur les images qui sont légèrement difficiles pour les lecteurs humains. La qualité d'un texte dépend de plusieurs facteurs [ALB1 95] :

✚ **Historique du document** : Un document qui a été copié ou faxé plusieurs fois, est plus difficile à lire qu'un document original.

✚ **Processus d'impression** : Un document composé (Typeset) est plus clair qu'un document imprimé, qui est à son tour plus clair que celui imprimé avec une imprimante de qualité inférieure. D'autres déformations peuvent avoir un rapport avec le processus d'impression, y compris la propagation et l'éclat de l'encre.

✚ **Acquisition d'image** : La binarisation de l'écriture On-line est limitée par la résolution et le taux d'échantillonnage de la tablette. Elle introduit souvent des distorsions comme les petits zigzags. La qualité d'un texte scanné est compromise par des variations de positionnement (pente, translation, élargissement, etc.), des châssis de document non propres, non mis au point, et de résolution limitée.

✚ **Conditions de document** : La présence des taches et des marques externes

rend la lecture très difficile.

Une fois que le texte est acquis, en mode on-line ou off-line, il doit être prétraité afin de simplifier sa reconnaissance. En fait, les opérations de prétraitement sont pratiquement des opérations de traitement d'image spécifiques qui transforment une image en une autre tout en réduisant le bruit et les variations. On y trouve : Binarisation, Filtrage et Lissage, Détection de la ligne d'écriture (ligne de base), Squelettisation, Alignement, et la Normalisation.

Peu de travaux ont été faits sur les caractéristiques de bruit affectant l'image de document ou sur l'analyse des effets du bruit sur le taux de reconnaissance, spécifiquement pour l'AOTR [ALB1 95].

### ✚ Binarisation (seuillage)

Cette opération consiste à transformer une image avec niveaux de gris en une image bi-modale (blanc et noir). Une procédure efficace de binarisation consiste à calculer l'histogramme d'image (i.e., représentation de la répartition des niveaux de gris). Celui-ci amène à déterminer le nombre de pixels correspondant à chaque valeur de niveau de gris. Après ce calcul, on cherche les pics, les vallées, les sommets, les creux, le centre et la dispersion de l'histogramme. Noter que, le creux (situé entre deux sommets de l'histogramme) représente le seuil de niveau de gris qu'il convient d'adopter pour une segmentation de l'image par binarisation : les pixels ayant un niveau de gris supérieur à ce seuil appartiennent au fond et ceux ayant une valeur inférieure appartiennent à l'objet [ZAH 90, BEN 98].

Certaines précautions doivent être prises pour sélectionner le seuil de binarisation : un seuil trop élevé conduit à un texte très mince et les portions connexes peuvent se disjoindre, tandis qu'un seuil trop bas produit un texte très épais et les portions séparées peuvent se rejoindre.

Des recherches ont commencé récemment, en essayant de reconnaître les images avec des niveaux de gris, sans binarisation à priori [PAV 96].

### ✚ Lissage et Filtrage

Son rôle majeur est d'éliminer le bruit sur les contours de l'image du texte. C'est une opération qui ne tient compte que du voisinage de point.

Divers opérateurs locaux de lissage ont été utilisés pour lisser les textes arabes. Amin et al. [AMI 80, AMI 82, AMI 85] appliquent 3 opérations pour le lissage d'un tracé on-line :

- ✚ élimination de tracé isolé,
- ✚ fusion de deux tracés lorsqu'ils se rapprochent suffisamment l'un de l'autre (i.e., si la distance séparant le point terminal d'un tracé donné de son successeur est inférieure à un seuil donné),
- ✚ Réduction de vibration du stylet en éliminant les points qui font un angle inférieur à un seuil préétabli.

Dans les images off-line, une méthode fréquemment utilisée pour le lissage, consiste à parcourir l'image pixel par pixel, en utilisant une fenêtre de taille 3 × 3 et à changer la valeur du pixel en se basant sur les valeurs de ses 8 voisins (Nord, Nord est, etc.). Cette fenêtre peut être utilisée, par exemple, pour remplir les trous et éliminer les points isolés [MAH 94, AMI 96]. Un point 1 est mis à 0 s'il n'y a pas assez de points (pixels) noirs autour de lui et vice versa.

Les opérateurs de la morphologie mathématique sont souvent utilisés pour le lissage [BED 95]. L'opérateur de fermeture peut éliminer les petits trous et remplir les vides sur le contour, tandis que l'opérateur d'ouverture peut couper les isthmes étroits et éliminer les petits îlots, les pics anguleux, et les capes.

Amin et al. [AMI 86] utilisent une méthode de lissage inspirée de la morphologie mathématique. Cette méthode consiste à remplacer la valeur de chaque point X (voir figure 3.6), par les expressions logiques suivantes :

- $X \leftarrow X + b \times g \times (d + e) + d \times e \times (b + g)$ ,
- $X \leftarrow (g + b \times d + b \times e + a \times h + c \times f)$ .

|   |   |   |
|---|---|---|
| a | b | c |
| d | X | e |
| f | g | h |

**Fig 2.6 :** Un point et ses 8 voisins.

Zahour [ZAH 90] utilise les 2 opérateurs de lissage suivants :

- a) **Dilatation :**
- |   |   |   |                        |
|---|---|---|------------------------|
| X | 1 | X | X vaut 0 ou 1          |
| 1 | P | X | dans ce cas P sera à 1 |
| X | X | X |                        |
- b) **Erosion :**
- |   |   |   |                        |
|---|---|---|------------------------|
| X | 1 | X | dans ce cas P sera à 0 |
| 0 | P | 0 |                        |
| 0 | 0 | 0 |                        |

### ✚ Squelettisation (Thinning)

Intuitivement, on peut imaginer ce que peut être le squelette d'une forme donnée : de même que le squelette d'un animal est la charpente sur laquelle il est construit, le squelette d'une forme est son axe médian [ZAH 90]. Par exemple : le squelette d'un disque est un point (son centre).

La squelettisation (amincissement), est une opération essentielle dans plusieurs systèmes AOTR [AMI 92, AMI 96, ALM 87, ABU 94, ZAH 90, BEN 98], elle consiste à réduire les formes à leurs squelettes, et exprime les relations structurelles dans la forme et réduit l'espace et le temps de traitement en simplifiant les structures de données (voir figure 2.7).

Généralement, les systèmes off-line de reconnaissance d'écriture manuscrite, utilisent la squelettisation pour atténuer la variabilité inhérente aux styles d'écriture [BEN 98], ou examinent le squelette d'un mot pour en simuler une représentation on-line [ABU 94]



**Fig 2.7 :** Squelette d'un mot binaire :(a) Image binaire du mot (واحد), (b) Squelette de (a).

### ✚ Normalisation

Etant donné que la taille des caractères arabes est très variable, la normalisation de la taille est souvent utilisée pour échelonner les caractères à une taille fixe et pour centrer les caractères avant leur reconnaissance. Cette opération est très utile dans les méthodes de reconnaissances qui sont sensibles aux petites variations dans la taille et la position comme c'est le cas dans le Template Matching et les méthodes de corrélation.

### ✚ **Détection de la ligne de base**

La ligne de base dans les textes arabes porte une information assez pertinente sur l'orientation de texte et sur les positions des points de connexion. La méthode la plus utilisée pour trouver la ligne de base est fondée sur la projection horizontale [GOW 90].

L'histogramme de la projection horizontale est tout simplement un vecteur ayant autant d'entrées qu'il y a de lignes dans l'image, chaque entrée contient le nombre de pixels noirs dans la ligne correspondante.

### **2.3.3. Segmentation**

La segmentation est une étape critique et décisive dans plusieurs systèmes de reconnaissance (ex., [AMI 82, AMI 85, AMI 86, AMI 89, ALM 87, BED 95]). En d'autres termes, l'efficacité d'un système de reconnaissance en dépend fortement.

Elle est définie, comme étant l'opération qui cherche à décomposer une image de texte en pseudo-images de symboles individuels. Le résultat de cette opération, est une forme isolée à partir d'une image et qui pourrait être un caractère ou non.

Cette opération est l'étape la plus critique dans plusieurs systèmes AOTR, elle est dans la plupart des cas sujette aux erreurs est très pénalisante en temps de calcul.

La segmentation d'une page de texte peut être divisée en deux niveaux : la décomposition de page et la segmentation du mot.

#### **2.3.3.1. Décomposition de page**

La décomposition de page fait partie du domaine de l'analyse de document. L'analyse de document [BEL1 92, BEL2 92] étudie la structure des documents et l'identification de ses différentes parties logiques.

Les travaux de reconnaissances de textes arabes que nous avons examinés, étaient limités à la séparation des différentes parties d'un bloc de textes (lignes, mots...). Cette opération peut se faire en suivant les étapes suivantes:

- A. la détermination de la pente (i.e., l'inclinaison dans laquelle le document pourrait avoir été scanné),
- B. La localisation des colonnes, paragraphes, ligne de textes, et mots ou pseudo-mots,
- C. La reconnaissance optique des caractères.

A cet effet, une méthode de séparation de ligne très connue, est basée sur l'utilisation de l'histogramme de projection horizontale. Les ruptures des lignes dans le texte correspond aux vides (ou minima) dans l'histogramme (ex., [AMI 86, AMI 89, BED 95]). Notons que, certains minima peuvent correspondre à un espace entre les caractères et leurs points. Dans certains travaux, des précautions sont prises pour ne pas séparer les points au-dessus ou au-dessous d'une ligne de texte de leurs propres lignes [ELD 90]. Cette méthode pourrait échouer lorsque l'espace inter-lignes est étroit, et les points bas d'une ligne apparaissent comme étant les points supérieurs de la ligne immédiatement inférieure.

Le problème de séparation de ligne peut être résolu, en trouvant en premier lieu la ligne de base par projection horizontale, puis en regroupant, comme ligne, toutes les composantes, qui sont plus proche d'une ligne de base que d'autre [ALB1 95] ou en faisant passer le profil horizontal par un filtre passe-bas pour trouver les minima locaux.

### 2.3.3.2. Segmentation du mot

Le problème de segmentation des mots latins qu'ils soient imprimés ou manuscrits a été largement étudié [DUN 92, FUG 92, LIA 94, CAS 96, LU 96,]. Malgré qu'il soit possible, parfois, d'appliquer les résultats de ces études pour l'Arabe, en général ils ne sont pas suffisants pour segmenter les mots arabes. Car un caractère arabe peut être constitué de plus d'une partie (primaire et secondaire) et la nature cursive de l'écriture arabe rend l'utilisation de ces méthodes insuffisante.

Dans la littérature de reconnaissance de l'écriture Arabe, on y trouve 5 approches principales pour traiter un caractère connecté à l'intérieur d'un mot :

- ✚ Supposer que le texte à reconnaître est déjà segmenté en caractère (i.e., elle ignore le problème de segmentation),
- ✚ Segmentation du mot à traiter en entités (primitives) plus petites qu'un caractère,
- ✚ Segmentation du mot à traiter en caractères (dissection),
- ✚ Segmentation des mots en se basant sur la reconnaissance (la segmentation étant un sous-produit de la reconnaissance),
- ✚ Méthode holistique (globale), dans laquelle le système cherche à reconnaître le mot comme un tout, en évitant ainsi le problème de segmentation en caractères.

### 2.3.4. Extraction de caractéristiques

En reconnaissance de formes en général et en reconnaissance de l'écriture en particulier, reconnaître n'importe quel objet ne peut se faire sans ses caractéristiques pertinentes. De ce fait, tous les systèmes OCR comportent une étape critique nommée extraction de caractéristiques. Cette étape était toujours considérée comme un problème d'une part difficile, et d'autre part important. La difficulté revient principalement au manque de règles préalablement établies pour obtenir les traits pertinents de l'objet à reconnaître.

L'importance est due à la pertinence des paramètres (caractéristiques) extraits à partir de l'image d'objet, car sans eux, la reconnaissance ne peut s'effectuer.

L'extraction de caractéristiques peut se définir comme étant le fait d'extraire, à partir de données brutes, l'information la plus pertinente pour des besoins de classification, afin de minimiser la variabilité intra-classes tout en l'augmentant entre les différentes classes de formes [TRI 96]. Elle est fortement liée à la classification. Puisque, c'est le type de caractéristiques extraites à ce niveau qui doit dicter la nature du classifieur à utiliser pour l'identification.

L'extraction de caractéristiques est une étape importante pour obtenir un bon système OCR. Cependant, les autres étapes (voir figure 2.5), nécessitent aussi d'être optimisées afin d'obtenir une meilleure performance.

Le choix d'une méthode d'extraction limite ou dicte la nature et les résultats de l'étape de prétraitement.

La sélection du type et de l'ensemble de caractéristiques peut contribuer énormément à améliorer le taux de reconnaissance d'un système OCR. Sont nombreux, les systèmes qui utilisent différents types de caractéristiques. Ces types peuvent être classés catégoriquement en quatre groupes principaux [ALB1 95] : (1) les caractéristiques structurelles, (2) les caractéristiques statistiques, (3) les transformations globales, et (4) la corrélation et le template matching.

### **I. Caractéristiques structurelles**

Les caractéristiques structurelles décrivent la forme en fonction de sa topologie et géométrie, tout en donnant ses propriétés globales et locales.

Ce type de caractéristiques peut tolérer des grandes variations et distorsions dans le style d'écriture (i.e., écriture manuscrite).

Cependant son extraction à partir de l'image du caractère ou du mot n'est pas toujours triviale.

Dans la littérature les caractéristiques structurelles utilisées dépendent du type de la forme à classifier.

Ces caractéristiques comprennent :

- ✚ Le nombre des traits (Strokes) [AMI 80, AMI 82], leurs tailles, directions, et pentes [EL-90],
- ✚ les points extrémaux (end points) [AMI 80, AMI 82, AMI 96, ZAH 90, BEN 98],
- ✚ les points d'intersection et de croisement [AMI 80, ZAH 90, AMI 96, BEN 98],
- ✚ La hauteur et la largeur du caractère [AMI 80, AMI 82, ZAH 90],
- ✚ La Catégorie de la forme (partie primaire, point diacritique, etc.) [ZAH 90, AMI 96],
- ✚ Le nombre de points diacritiques et leurs positions par rapport à la ligne de base, et les zigzags (Hamza's)[ELS 1 88, GOW 90, MAH 94, BEN 96, BEN 98].
- ✚ Les points de rebroussement et d'inflexion [ZAH 90, BEN 98],
- ✚ Le nombre d'occlusions (boucles)[ALM 87, AME 94, AMI 96, MIL 97, SOU 98],
- ✚ Les concavités et les convexités dans les quatre principales directions (ouest, nord, est, sud) [MAH 94],
- ✚ Les Jambages (descendants) et les hampes (ascendants) [BEN 96, BEN 97, MIL 97, SOU 98],
- ✚ La taille du rectangle englobant du tracé ou du caractère [AMI 80, AMI 82, ALM 87, ZAH 90],
- ✚ Le nombre de composantes connexes [AME 94, MIL 97, BEN 98, SOU 98],
- ✚ Les courbures avec leurs directions (ouest, est, nord, sud) [AMI 85, AMI 96, BEN 98],
- ✚ Les segments droits et obliques avec leurs directions [AMI 96, BEN 98].

### **II. Caractéristiques statistiques**

Dans l'approche statistique un ensemble de mesures caractéristiques est extrait pour chaque forme. Ces mesures sont appelées les caractéristiques statistiques de la forme, et elles sont utilisées pour la reconnaître. Dans ce cas, toutes les caractéristiques de toutes les formes ensemble, avec leurs affectations aux classes composent l'espace caractéristique du système de reconnaissance [ELD 90].

Les caractéristiques statistiques utilisées pour la reconnaissance des textes arabes comprennent : Le zonage (Zonning), les caractéristiques de lieu géométrique, et les moments [ALB1 95].

Les moments invariants sont souvent utilisés comme caractéristiques statistiques [ELD 90, ALY 92, BED 95]. Ces caractéristiques sont invariantes en ce qui concerne la taille, la translation et la rotation.

L'inconvénient majeur de cette approche, peut être attribué principalement au temps de calcul des moments, ce qui limite la vitesse de lecture du système (ex. [ELD 90]).

D'autres chercheurs utilisent le zonage de pixels comme des caractéristiques en divisant le caractère en régions non chevauchantes ou chevauchantes et utilisent la densité de pixels dans ces régions comme des caractéristiques [SOU1 96].

La méthode de caractéristiques Loci est basée sur le calcul de nombre des segments blancs et de segments noirs le long d'une ligne verticale traversant la forme, ainsi que leurs longueurs. Tandis que, La méthode de croisement calcule le nombre de fois qu'un ensemble de lignes prédéterminées à différents angles (ex. 16 lignes à 0°, 22.5°, 45, etc.) traversent la forme.

Cette méthode tolère des distorsions et des variations légères, et le calcul y est facile [ALB1 95].

### III. Les Transformations globales

Leur principe consiste à convertir la représentation pixels de la forme en une représentation alternative plus abstraite qui conserve l'allure générale de la forme tout en réduisant la taille des caractéristiques.

La représentation du squelette ou du contour de la forme comme une chaîne de codes (Chain codes : i.e., une suite de codes de Freeman (voir figure 2.8)) de directions est l'exemple le plus courant de ces transformations. Ces codes de directions peuvent correspondre aux 8 directions principales comme dans le code de Freeman [MAH 94, AMI 89].

Amin et al. [AMI 86] utilisent une transformation basée sur les projections horizontale et verticale de la forme. Durant le balayage de chaque ligne de l'image du caractère, tous les points connectés sont désignés selon leur ordre d'apparition. L'algorithme d'extraction de primitives procède de la manière suivante :

- ✚ Génération de chaîne :
- ✚ Constituer l'histogramme de chaque ordre ;
- ✚ Calculer la valeur moyenne de chaque histogramme. Le groupe éventuel des points et de segments secondaires sont pris en considération.

### IV. Template Matching et Corrélation

Elles se basent sur la comparaison de la forme pixel par pixel à un ensemble de patrons (Template) de formes.

Dans ce cas, l'image entière du caractère ou du mot est utilisée comme un vecteur caractéristique. Au niveau de la reconnaissance, une mesure de similarité (ou dissimilarité) entre chaque patron  $T_j$  et l'image de la forme  $Z$  est calculé.

Le patron  $T_k$ , ayant la mesure la plus élevée de similarité (ou la plus petite dissimilarité), est identifié à condition que cette similarité soit supérieure à un certain seuil.

Cette méthode est rapide de point de vue calcul, mais elle est très sensible aux distorsions. Pour remédier à cela, les chercheurs utilisent différentes variantes de

cette méthode dont l'une des plus fondamentales consiste à choisir plusieurs pixels dans des positions clés pour la corrélation.

D'autres travaux décrivent des algorithmes pour sélectionner un ensemble de pixels connectés et communs à un nombre de caractères, en faisant correspondre chaque caractère à tous les caractères dans chaque position déplaçante. Ces fragments sont considérés comme des caractéristiques qui seront utilisées pour apparier les caractères [ALB1 95].

### **2.3.5. Classification**

Après avoir extrait les caractéristiques de la forme à reconnaître, le système OCR procède automatiquement à la classification. La classification est le stade principal de prise de décision. Celle-ci concerne l'identification de la forme, comme étant le membre d'une classe. Elle se fait en comparant les caractéristiques de la forme à celles d'un modèle de référence. Souvent, le résultat de cette étape est un ensemble de solutions au lieu de générer une solution unique.

La classification cherche, à partir des caractéristiques de la forme, parmi les modèles d'apprentissages en présence, ceux qui sont les plus proches. La nature de la proximité a un sens différent en fonction de la nature de représentation et du type de méthode. La décision peut conduire à un succès si la réponse est unique. Elle peut conduire à une confusion si la réponse est multiple. Enfin, la décision peut conduire à un rejet de la forme si aucun modèle ne correspond à sa description.

### **2.3.6. Les approches Hybrides**

Pour améliorer, la performance, une tendance actuelle consiste à construire, des systèmes hybrides qui utilisent divers types de caractéristiques et combinent plusieurs classificateurs arrangés en couches. De ce fait, de plus en plus de chercheurs utilisent actuellement une combinaison de plusieurs types de caractéristiques (structurelles, statistiques,...) et des techniques de classification.

Almmallim et Yamaguchi [ALM 87] utilisent trois niveaux d'extraction/ classification de caractéristiques pour reconnaître les mots manuscrits. Suivant ce schéma, la reconnaissance d'un mot se fait : premièrement en identifiant le groupe de chaque tracé (5 groupes ont été déterminés pour la classification des tracés) par utilisation de ses caractéristiques structurelles. Deuxièmement, en classifiant proprement dit chaque tracé, en calculent à partir de celui-ci un vecteur de certaines caractéristiques spécifiques, et en trouvant sa distance par rapport à un ensemble de vecteurs d'identification prédéterminé. Finalement, en convertissant la chaîne de tracés identifiés comme étant des caractères par une analyse syntaxique de cette chaîne.

Amin et al. [AMI 86] segmentent premièrement, un mot imprimé en caractères et divisent chaque caractère en (jusqu'à 7) primitives en utilisant la longueur des lignes (colonnes) des projections horizontale et verticale comme des caractéristiques. Les secondaires (points, Hamza) sont, deuxièmement, identifiés en utilisant leur densité. L'histogramme de projection est codé comme une chaîne de symboles qui sont les indications où une ligne (colonne) est de longueur zéro, plus grande que la moyenne de projection ou plus petit que la moyenne. Puis, La

classification de la chaîne simplifiée est effectuée en utilisant quatre arbres de décision, un pour chaque forme possible de chaque caractère.

D'autres systèmes ne suivent pas le paradigme classique de la reconnaissance de forme qui est l'extraction de caractéristiques suivie par la classification [ALB 2 95].

Al-Badr et Haralik [ALB2 95] utilisent un ensemble de 12 primitives prédéterminées de forme comme étant des caractéristiques. Puis ils détectent ces primitives de forme sur l'image du mot en utilisant des opérations de la morphologie mathématique. Dans ce schéma de représentation, un mot est défini comme un ensemble de primitives dépendantes les unes des autres. Le système conçu fait correspondre les primitives détectées avec les symboles modèles (description du symbole ou de forme en terme de primitives). Les primitives du symbole modèle représentent les primitives détectées sur l'image idéale (non bruitée) du symbole et les déformations attendues. Ces symboles modèle sont générés lors de l'étape d'apprentissage. La reconnaissance conduit automatiquement à un arrangement spatial des modèles de symboles appris. Le système effectue une recherche dans l'espace d'arrangement des modèles et choisira l'arrangement ayant la plus haute probabilité comme étant la reconnaissance de mot.

Trenkle et al. [TRE 95] proposent 3 approches séparées pour la reconnaissance des textes arabes imprimés, qu'ils les intègrent dans un seul système de reconnaissance.

La première approche est basée sur un réseau de neurones, elle consiste à détecter les positions potentielles des centres de caractères sur l'image du mot et procède à la reconnaissance des caractères sur ces positions éventuelles.

La deuxième approche, est basée sur la sur-segmentation du caractère, qui divise l'image de mot en plusieurs segments de pseudo-caractères et ensuite assemble ces dernières en caractère. La troisième approche calcule des caractéristiques sur l'image de mot (sans segmentation). Les caractéristiques extraites de l'image sont obtenues en appliquant certaines opérations de la morphologie mathématique. Dans chacune de ces 3 approches, un lexique de mots arabes était utilisé. L'intégration de ces trois approches en un seul système s'est fait grâce à des stratégies de décision définies par les auteurs.

### 2.3.7. Apprentissage

L'apprentissage se charge d'acquérir les connaissances et de les organiser en classes ou modèle de références. Dans un système OCR, l'apprentissage désigne deux phénomènes : l'entraînement, et l'adaptation.

L'entraînement (ou modélisation) est le processus qui consiste à faire apprendre au système les descriptions des caractères en utilisant des données étiquetées avec les noms des caractères. Ce processus doit être effectué en mode Off-line (avant la reconnaissance). Dans la littérature de reconnaissance de forme, on y trouve 2 types d'entraînement : supervisé et non supervisé.

- ✚ ***l'entraînement supervisé*** les différentes familles de caractères sont connues a priori et la tâche d'entraînement est guidée par un superviseur ou professeur. Le professeur, c'est-à-dire le concepteur indique pour chaque caractère (échantillon rentré), le nom de la famille qui le contient. Tous les systèmes AOTR, que nous avons examiné utilisent ce mode d'entraînement.

- ✚ **L'entraînement non supervisé**, on l'appelle suivant l'approche utilisée classification automatique, inférence ou encore entraînement sans professeur [BEL1 92], il s'agit dans ce cas, à partir d'échantillons de référence et de règles de regroupement (ou de modélisation) de construire automatiquement les classes (ou les modèles) sans intervention du superviseur.

Ce mode d'entraînement nécessite un nombre élevé d'échantillons et des règles de construction précises et non contradictoires pour bien assurer la fonction correcte des classes.

Le processus d'apprentissage off-line supervisé est une caractéristique principale d'un classifieur statistique, puisque le concepteur n'a pas besoin de construire manuellement le classifieur. Par exemple, un classifieur Bayésien, apprend les probabilités conditionnelles des caractéristiques sachant les caractères ou les mots [ALB2 95], tandis qu'un classifieur du plus proche voisin utilise la notion de proximité (distance) pour apprendre les données en vue de leur classification [ELW 89].

- ✚ **L'adaptation** : est le processus d'amélioration des performances d'un système OCR en profitant des expériences antérieures.

Certains systèmes permettent à l'utilisateur d'identifier le caractère lorsqu'ils n'ont pas pu le reconnaître. Ils utilisent la réponse de l'utilisateur afin de reconnaître le caractère quand ils le rencontreront une autre fois. L'adaptation est recommandée spécialement pour s'adapter à l'écriture particulière d'un utilisateur [AMI 92].

### 2.3.8. Post-traitement

La dernière étape dans le processus de reconnaissance est le post-traitement. Grâce à l'utilisation des informations contextuelles, un post-traitement peut contribuer efficacement dans l'amélioration du taux de reconnaissance d'un mot. Cette étape est souvent implémentée comme un ensemble de techniques qui dépend de plusieurs facteurs entre autres : les fréquences d'apparition des mots, les lexiques, et d'autres informations sur le contexte. Comme la classification produit parfois un ensemble de solutions au lieu d'en générer une seule, le post-traitement est responsable de la sélection de la solution exacte en ayant recours à des informations de haut niveau (lexicales, syntaxiques, sémantiques, pragmatiques,...) qui ne sont pas disponibles au niveau du classifieur. Parmi les opérations de post-traitement on trouve la vérification et la correction d'orthographe. La vérification d'orthographe peut être aussi simple qu'une recherche des mots dans un lexique.

Dans le but d'accélérer la vitesse de vérification d'orthographe des mots reconnus, Amin et al. [AMI 86, AMI 85] représentent le lexique de la reconnaissance par un arbre. Dans ce cas, la recherche sur mot donné revient à trouver un chemin dans l'arbre correspondant à celui-ci. Dans un autre travail, Amin et al. [AMI 89] utilisent l'algorithme de Viterbi [FOR 73] lorsque la vérification d'orthographe échoue, pour trouver les mots possibles dont les caractères peuvent être substitués avec une grande probabilité par les caractères originaux (fournis par le module de la reconnaissance), en utilisant un modèle de Markov caché pour chaque mot.

Traditionnellement, lorsqu'on cherche les mots dans un dictionnaire, on doit tout d'abord enlever les préfixes et suffixes, on réduit alors le mot résultant en sa racine. Cette méthode de représentation d'un mot comme une racine et une forme

(la forme spécifique comment la racine est reliée au mot original) peut réduire considérablement la taille du dictionnaire.

Dans ses premiers travaux Amin [AMI 82] construit tous les mots possibles à partir des résultats de la classification. Puis il utilise des règles phonétiques spécifiques à un lexique de taille réduite pour corriger les mots incorrects.

## 2.4 Les Perspectives

Pour atteindre son objectif, un système OCR doit reconnaître ses entrées correctement. Kahan et al. [KAH 87] ont rapporté que pour qu'un système OCR atteigne la performance d'un dactylographe (Typist), le taux de reconnaissance minimal acceptable doit être de 99.9%, et la vitesse de lecture d'au moins 5 caractères/seconde, sachant que toutes les erreurs seraient des rejets plutôt que des substitutions. Une expérience pratique affirme qu'une partie considérable du coût total de reconnaissance est consacrée à éditer manuellement les erreurs du système OCR [NAG1 92].

Les systèmes OCR Latins peuvent atteindre des taux arrivant jusqu'à 99.9% sur des documents imprimés à haute qualité de fonte homogène [KAH 87] et un taux de 99% pour la reconnaissance on-line. En revanche, les systèmes AOTR sont loin d'atteindre le taux de 99.9%. La Table 2.2 résume certaines techniques et taux de reconnaissances de quelques systèmes AOTR.

| Références         | Capacité                       | Segmentation            | Caractéristiques               | Classification                        | Post-Traitement                | Performance       |
|--------------------|--------------------------------|-------------------------|--------------------------------|---------------------------------------|--------------------------------|-------------------|
| Al-Emami [AL- 90]  | Mots manuscrits On-line        | Primitives              | Structurelles, Chaîne de codes | Arbre de décision                     |                                | TM 86-100 %       |
| Almuallim [ALM 87] | Mots manuscrits off-line       | Primitives              | Structurelles                  | Syntaxique, à distance                |                                | TC 91 %           |
| Amin [AMI 80]      | Caractères On-line             |                         | Chaîne de codes                | Plus proche voisin                    |                                | TC 95.4 %         |
| Amin [AMI 82]      | Mots on-line                   | Caractères              | Structurelles                  | Recherche dans un dictionnaire        | Lexique                        | TC 80 %           |
| Amin [AMI 89]      | Textes multi-fontes            | Caractères              | Chaîne de codes                | Arbre de décision                     | Lexique, Algorithme de Viterbi | TC 90 % à 180 CPM |
| Amin [AMI 96]      | Caractères manuscrits off-line |                         | Structurelles                  | Réseau de neurones                    |                                | TC 92 %           |
| EL-Dabi [ELD 90]   | Textes imprimés                | Après la reconnaissance | Moments                        | Recherches dans un dictionnaire       |                                | TC 94 % à 180 CPM |
| EL-GOWELY [GOW 90] | Textes multi-fontes composés   | Caractères (récursive)  | Statistiques                   | Arbre de décision, Règle heuristiques |                                | TC 94 % à 240 MPM |

**Table 2.2 :** *Caractéristiques et performance de certains systèmes OCR arabe (TC : Taux de reconnaissance de caractères; TM : Taux de reconnaissance de mot; CPM : Caractères par minute; MPM : Mots par minute).*

La plupart des résultats de performance rapportés dans la littérature n'étant pas statistiquement valides, on ne peut pas les utiliser pour comparer les performances des systèmes OCR.

Les résultats de performance rapportés soit dans la littérature soit par les constructeurs des systèmes OCR sont généralement effectués sur des ensembles de test qui sont petits, à fonte spécifique, et non bruités ou ils peuvent avoir été utilisés dans la conception avec un peu d'analyse éventuelle des résultats [ALB1 95]. Ainsi, les résultats rapportés dans les papiers peuvent s'interpréter, généralement, comme la meilleure performance possible, sous les conditions les plus favorables.

Le domaine d'AOTR nécessite d'une manière décisive un ensemble standard de documents de test, à la fois en format d'images et de caractères, et un ensemble d'outils d'évaluation de performances. Ceci permettrait vraiment la comparaison de la performance des différents systèmes AOTR. Plusieurs ensembles (bases) de test standards sont disponibles, pour l'Anglais sous forme de CD-ROM.

La majorité des systèmes AOTR n'abordent pas directement le problème d'entrée déformée et dégradée. En réalité, il est difficile d'avoir les documents en parfait état. L'utilisation des modèles de bruit peut améliorer la performance d'OCR pour une entrée déformée. Un modèle de distorsion pour le document en entrée permet l'étude de la dégradation du taux de reconnaissance comme une fonction de la distorsion dans le document entré. Avec un tel modèle, la distorsion dans l'entrée peut varier d'une manière continue et contrôlée.

De plus, la caractérisation du processus de dégradation facilite la conception des algorithmes de prétraitement pour renverser la dégradation [ALB 1 95].

Actuellement, la tendance principale de recherches en AOTR s'oriente vers la production de systèmes ayant de bons taux de reconnaissance. Comme la différence essentielle entre la reconnaissance de L'Arabe et du Latin est la nécessité d'un stade de segmentation, la majorité des recherches est concentrée sur la conception des algorithmes de segmentation [ex., [AMI 92]] ou sur l'utilisation des méthodes qui ne nécessitent pas de segmentation [ALB2 95].

Le paradigme de séparation des stades de segmentation et de reconnaissance a une limitation évidente (les erreurs en segmentation sont fatales), puisqu'elles conduisent automatiquement à une reconnaissance erronée.

Certaines recherches s'orientent actuellement à des stades itératifs de segmentation et reconnaissance. Une variante de ce paradigme est que lorsque la reconnaissance d'une forme échoue, le système OCR réessaie de re-segmenter la forme (ex., [GOW 90]).

Vu que chaque type de caractéristiques et chaque méthode de classification a ses points forts et ses points faibles, nous avons décrit préalablement quelques systèmes hybrides qui combinent plusieurs types de caractéristiques et méthodes de classification pour améliorer la performance.

Une autre manière d'y arriver est d'utiliser simultanément des classifieurs multiples, puis de combiner les résultats obtenus par les différents classifieurs [NAG2 92, XU 92, SRI 93, SUE 93, TRE 96, CHO 97] ou bien combiner les deux approches de reconnaissance (i.e., Analytique et Globale) [MOH 96, PAQ 93].

Pour effectuer un post-traitement efficace, les systèmes OCR doivent utiliser des informations de plus haut niveau, comme l'information sur le sujet du document, la sémantique du langage, ainsi que des informations sur l'uniformité du type de fonte, son style, et sa taille dans les mots et les paragraphes.

Avec une technologie d'AOTR avancée, il serait possible d'étudier la reconnaissance de texte avec des marques diacritiques (ex., Kasrah, Fathah,...) et les textes calligraphiques, qui sont encore plus difficiles à reconnaître que les documents composés (Typest).

Comme les systèmes OCR ont progressé, des recherches sont actuellement concentrées sur l'analyse de document (ex., [BEL1 92, BEL2 92]). L'analyse de document vise entre autre, la compréhension de la disposition topographique de papier. Ceci permettra de trouver automatiquement l'ordre correct de la lecture pour les documents multi-colonnes et multi-langues et séparer le texte des illustrations, formules mathématiques, et figures. L'analyse de la topographie permet de reconnaître les documents multi-langues en séparant et en identifiant automatiquement la langue des zones homogènes sur la page.

## 2.5 Conclusion

Dans ce chapitre nous avons présenté une étude détaillée des recherches sur la reconnaissance optique de texte arabe (AOTR). Les recherches sur AOTR sont considérées moins avancées que d'autres langues (ex., Latine, Chinoise, Japonaise...). Ce retard peut être attribué principalement au début tardif des recherches (au milieu des années soixante-dix pour l'Arabe, en comparaison, au milieu des années quarante pour le Latin) et aux caractéristiques spéciales de l'écriture arabe qui est par nature cursive. Cependant, Ces dernières années on constate une augmentation considérable dans le nombre de chercheurs et de publications visant L'AOTR [ALB1 95]. Parallèlement, des systèmes opérationnels sont apparus sur le marché pour L'OCR arabe (malgré qu'ils ne soient pas aussi sophistiqués que ceux pour le Latin).

Nous avons en premier lieu discuté les principaux aspects liés à la reconnaissance automatique de l'écriture d'une manière générale. Puis, nous avons détaillé les caractéristiques de l'écriture arabe qui influencent la reconnaissance. Un modèle général à six étapes a été introduit dans la troisième partie. Ce dernier reflète d'une manière générale le processus de reconnaissance. Ensuite, nous avons abordé chaque étape du modèle (Acquisition, prétraitement, segmentation, extraction de caractéristiques, classification, post-traitement). A chaque étape, nous avons analysé et catégorisé les méthodes de recherche d'AOTR, et indiqué leurs limitations, particularités, et avantages. La segmentation peut être effectuée au niveau de la page et au niveau du mot. Au niveau du mot, nous avons cité les 5 approches de segmentation. Les types principaux de caractéristiques utilisés dans AOTR sont : *Structuraux*, *Statistiques*, *Globaux*, et *le Template Matching*; tandis que les principales approches de classification sont : structurelles, statistiques, et neuronales. Par ailleurs, nous avons cité les systèmes utilisant divers types de caractéristiques hybrides et des approches hybrides de classification. Nous avons constaté des insuffisances concernant l'utilisation du post-traitement dans les travaux sur l'OATR.

Nous avons arrivé à la conclusion que le domaine d'OATR, d'une manière générale nécessite des bases de données d'images de documents arabes (et leurs textes associés) contenant assez de données pour que les chercheurs puissent les utiliser et rapporter la performance de leurs travaux.

De plus, l'analyse des effets de dégradation d'images dans l'AOTR est limitée, et beaucoup de travail est nécessaire dans ce sens.

Comme l'écriture arabe est par nature cursive, l'un des grands obstacles rencontrés dans un effort pour améliorer l'AOTR est le manque de solutions effectives et efficaces au problème de segmentation. Les recherches sur la segmentation en particulier, et sur la reconnaissance en général, doivent se concentrer sur le traitement des textes dégradés, italiques et inclinés, ligaturés et les textes ayant des caractères qui se chevauchent, les textes avec des diacritiques, et omni-fonte et calligraphiques.

Pour améliorer les taux de reconnaissance, les chercheurs peuvent utiliser les systèmes hybrides, les systèmes multi-classifieurs, et les informations lexicales et contextuelles. Finalement, pour améliorer la vitesse, il pourrait être nécessaire d'utiliser un traitement parallèle et d'implémenter certaines fonctions d'AOTR en Hardware.

## **Chapitre 3**

# Méthodes et outils actuels de la reconnaissance des textes manuscrits

### **3. Introduction**

L'extraction de l'information est la première étape de la reconnaissance. Elle est réalisée à l'aide de l'extraction des primitives, qui sont des informations élémentaires. La primitive est l'outil de la reconnaissance.

Nous traiterons en premier lieu de l'extraction des primitives dans le cas général, c'est-à-dire quel qu'en soit le niveau (lettre, mot ou texte). Dans une seconde partie, nous nous concentrerons sur l'exploitation de ces primitives pour la reconnaissance des mots manuscrits.

### **3.1. Les primitives, outils de la reconnaissance**

Il n'est pas dans nos intentions de reprendre une étude qui maintes fois a été faite mais seulement de rappeler certains outils, auxquels nous ferons souvent appel par la suite. Auparavant, nous préciserons les objectifs ainsi que la problématique de l'extraction de l'information.

#### **3.1.1. Les objectifs de l'extraction des primitives**

Au cours de l'extraction des primitives, plusieurs objectifs, qui précèdent la reconnaissance, peuvent être envisagés. Les principaux objectifs que nous définirons dans la perspective de la reconnaissance de l'écriture manuscrite sont : l'analyse, la segmentation, le paramétrage, la modélisation, le codage et la classification.

✚ **L'Analyse**, dont la définition littérale est "la décomposition d'un tout en ses parties", consiste généralement en l'extraction d'un ensemble d'attributs caractéristiques du texte. Concrètement, l'analyse d'un texte manuscrit consiste à recueillir des informations statistiques telles que la disposition des lignes d'écriture,

leur orientation, leur régularité, l'espacement des mots et des lettres, la régularité et l'inclinaison des lettres, l'épaisseur du trait, ainsi que la ligature des lettres à l'intérieur des mots. L'objectif de l'analyse peut aboutir éventuellement à l'identification du scripteur, le signifiant [LORETTE 92], plutôt que directement à la reconnaissance du texte, le signifié.

✚ La **Segmentation** désigne la séparation de l'information en ses éléments constitutifs afin de les identifier isolément. La définition de la segmentation est donc assez proche du sens littéral du mot analyse. On parle de sur-segmentation lorsque l'élément constitutif est lui-même fragmenté, et de sous-segmentation lorsque plusieurs éléments constitutifs n'ont pas pu être isolés. Dans le cas de la reconnaissance des mots, les éléments constitutifs sont naturellement les lettres du mot. Il est nécessaire de distinguer la segmentation logique de la segmentation physique du document [LORETTE 92]. Les éléments logiques sont les éléments sémantiques composant le texte, c'est-à-dire les lettres ou les mots, ils ne

correspondent pas toujours aux éléments physiques qui sont liés aux pixels de l'image. L'extraction des composantes connexes est un exemple typique de segmentation physique.

✚ Le **Paramétrage** consiste à établir une liste d'attributs représentés par une variable binaire (attribut présent ou absent) ou multivaluée (proportionnelle à l'importance de l'attribut), qui ont été détectés et évalués dans l'image. A la différence de l'analyse, le paramétrage ne concerne qu'un mot ou qu'une lettre en vue de sa reconnaissance.

✚ La **Modélisation** est la construction d'une représentation approximative de la forme entière. A la différence du paramétrage, l'objectif est la réduction de l'information utile, au minimum nécessaire pour représenter complètement la forme, en particulier son aspect et sa structure. Si la modélisation est une description proche d'aspect de la forme originale, on parle alors d'une schématisation, avec une approximation plus ou moins importante. Sinon, il s'agit d'un codage de l'information. Dans ce cas, on ne se soucie pas de l'aspect de la modélisation, mais seulement de la pertinence de l'information qu'elle apporte à la reconnaissance. On remarque que, d'une façon générale, la schématisation par exemple d'un mot est une structure de données en deux dimensions (2D), tandis que le codage aboutit à une structure de données arborescentes (arbre de primitives 1D-2D), alors que le paramétrage se traduit par une liste ou bien un vecteur (1D).

✚ La **classification** peut être considérée comme une identification partielle de l'information. L'objectif de la classification est, lorsque l'on ne dispose pas de toute l'information nécessaire à l'identification complète de la forme (cas de la reconnaissance), de déterminer quand même une catégorie à laquelle elle appartient. Le principe est que moins d'information est nécessaire pour distinguer les caractères que pour les reconnaître.

Ces différents objectifs participent à la **Segmentation**, la **Transformation** (ou le Changement de Représentation) et la **Réduction** de l'information, qui sont les trois axes de la reconnaissance.

### 3.1.2. La problématique de l'extraction de l'information

L'objectif de l'étape de l'extraction de l'information est la sélection de l'information pertinente qui se trouve noyée dans la masse de l'information brute acquise. La problématique de cette étape a pour origine le risque de perte d'information signifiante pour la reconnaissance. La minimisation de ce risque est conditionnée par deux dilemmes liés chacun à un paradoxe de causalité. Il s'agit du dilemme de la réduction et du dilemme de la segmentation.

Nous allons développer dans cette partie la nature de ces dilemmes en présentant les différentes origines de perte d'information lors du processus de la reconnaissance de l'écriture manuscrite.

La première étape du processus est l'acquisition de l'information. Il est évident que l'on ne peut pas extraire une information qui n'est pas présente dans le tracé. C'est pourquoi, avant de traiter de la perte d'information pendant la phase d'acquisition, nous considérerons la perte d'information pendant la phase de production de l'écriture.

#### *- La perte d'information pendant la phase de production de l'écriture*

Cette première origine de perte d'information, qui a lieu pendant l'écriture elle-même, est due au facteur humain. Cette question, qui s'écarte un peu de notre sujet, a cependant des conséquences déterminantes sur la reconnaissance.

Nous pensons que l'origine de la perte d'information est due à la projection dans un espace de dimension inférieure, du concept des mots lors de leur tracé linéaire dans le plan.

Le caractère subjectif de l'écriture tient au fait que le lecteur étant aussi le scripteur, il peut ne pas s'apercevoir que son écriture est illisible !

L'auteur a en outre la liberté de s'appliquer à ne bien tracer que les lettres permettant de distinguer les mots. Cette attitude a une conséquence importante sur la stratégie de lecture à adopter pour la reconnaissance. En effet, elle implique d'avoir la connaissance du vocabulaire parmi lequel l'auteur a estimé pouvoir distinguer les mots.

Cette perte d'information peut se manifester par une grande variation dans le tracé de l'écriture, par rapport au modèle scolaire ou à un tracé moyen, ou bien par une influence entre le tracé des lettres adjacentes dans le mot, qui augmentera ensuite la difficulté de les séparer.

La limite à considérer pour ce type de perte d'information est que si un humain ne peut pas lire un texte, un ordinateur ne le pourra probablement pas non plus. Cela fixe une borne inférieure à la qualité d'écriture admissible pour un système de reconnaissance.

#### *- La perte d'information pendant la phase d'acquisition*

L'objectif de cette étape est d'acquérir le maximum d'informations, afin d'obtenir des images numériques les plus précises. La perte d'information pendant l'acquisition est liée le plus souvent à la faiblesse des capteurs (scanner ou éventuellement caméra), ou aux conditions d'acquisition.

A ce niveau du processus, il reste à déterminer si l'information signifiante est accessible ou non parmi toute l'information acquise.

#### *- La perte d'information pendant la phase de réduction*

La phase de réduction consiste à extraire l'information signifiante qui est noyée dans le bruit.

Dans le cas de la classification, l'objectif est d'éliminer davantage de bruit que d'information afin de faciliter la discrimination.

Dans le cas de l'approximation, l'objectif est de modéliser en minimisant la perte d'information. Mais, plus précise est la modélisation, plus la procédure de

reconnaissance est complexe et plus la représentation est sensible aux variations de tracé (fioritures non signifiantes pour la reconnaissance). Comment s'affranchir de cette forte variabilité du tracé grâce à l'approximation, sans perdre l'information signifiante nécessaire à la reconnaissance ?

Ce dilemme de la réduction repose sur le paradoxe que l'on ne peut savoir qu'une information est signifiante ou non tant que la forme n'est pas reconnue.

Il en résulte une certaine subjectivité de l'étape de réduction, ce qui implique un risque de perte d'information.

Une autre perte d'information se produit également au cours de la phase de segmentation.

*- La perte d'information pendant la phase de segmentation*

A l'instar du dilemme de la réduction, le problème de la segmentation nous ramène une seconde fois au paradoxe de causalité.

Pour pouvoir segmenter le mot en lettres afin de les reconnaître isolément, il est nécessaire au préalable de les localiser, or comment localiser ces lettres sans les avoir reconnues auparavant?

## **3. 2. L'extraction des primitives**

### **3.2.1. Les approches de l'extraction des primitives**

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, tandis que, 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.

Au-delà de cette classification un peu formelle, la différence entre les approches systématique et heuristique comme entre le caractère a priori ou a posteriori, s'avère plus nuancée dans la pratique.

### **3. 2.2. Les catégories de primitives**

Nous avons distingué quatre catégories principales de primitives : les primitives topologiques, statistiques, structurelles et globales.

#### ***Les primitives topologiques ou métriques***

Le terme métrique désigne la mesure d'une distance. La topologie est "l'étude des propriétés de l'espace (et des ensembles) du seul point de vue qualitatif". Concrètement, la topologie consiste, à l'aide de sondes appliquées directement sur l'image "brute", à effectuer par exemple sur l'échantillon les mesures et les tests suivants :

- ✚ compter dans une forme le nombre de trous,
- ✚ évaluer les concavités,

- ✚ mesurer des pentes et autres paramètres de courbures et évaluer des orientations principales,
- ✚ mesurer la longueur et l'épaisseur des traits,
- ✚ détecter les croisements et les jonctions des traits,
- ✚ mesurer les surfaces, les périmètres,
- ✚ déterminer le rectangle délimitant l'échantillon, ou le polygone convexe,
- ✚ évaluer le rapport d'élongation (ou allongement) longueur/largeur, ...
- ✚ rendre compte de la disposition relative de ces primitives.

### ***Les primitives structurelles***

A la différence des primitives topologiques, les primitives structurelles sont généralement extraites non pas de l'image brute, mais à partir d'une représentation de la forme par le squelette ou par le contour. Ainsi, on ne parle plus de trous, mais de boucles ou de cycles dans une représentation filiforme du caractère. Cependant, pour le reste, les primitives structurelles correspondent à peu près aux primitives topologiques, il s'agit principalement :

- ✚ des segments de droite,
- ✚ des arcs, boucles et concavités, des pentes,
- ✚ des angularités, points extrémité et points terminaux, jonctions et croisements.

### ***Les primitives statistiques***

Elles véhiculent une information qui est distribuée sur toute l'image. L'histogramme, qui représente le nombre de pixels sur chaque ligne ou colonne de l'image, en est un exemple classique et simple à calculer. L'histogramme directionnel est plus long à calculer car il nécessite par exemple l'utilisation d'un algorithme de BRESENHAM [BRET 88] permettant de compter le nombre de pixels contenus sur une ligne de direction quelconque de l'image. L'histogramme des transitions, comme l'indique son nom, ne retient que le nombre des transitions 0-1 et 1-0.

Une application pratique des primitives statistiques de la sorte est réalisée grâce à l'intersection de la forme avec un réseau de droites pour la reconnaissance des chiffres manuscrits [AUTRET 91]. Dans ce cas, des histogrammes de transition sont construits seulement avec un échantillon de quelques droites au lieu de la totalité de celles-ci, ce qui permet une réduction a priori des données caractéristiques.

Pour s'affranchir du choix arbitraire de ces droites (espacement et orientation des droites du réseau) qui doit nécessairement être fixe entre l'apprentissage et la reconnaissance, on peut faire intervenir les probabilités en calculant la fréquence des intersections de la forme avec une série de droites aléatoires [HEUTTE 93].

On utilise aussi une autre primitive statistique basée sur un moyennage des pixels situés à l'intérieur d'un masque rectangulaire. La construction d'une matrice de masque recouvrant la totalité de la forme permet une représentation statistique à partir d'un nombre très réduit de valeurs correspondant à chaque masque.

### ***Les primitives globales***

Elles sont naturellement basées sur une transformation globale de l'image. La caractéristique d'une primitive globale est de dépendre de la totalité des pixels d'une image.

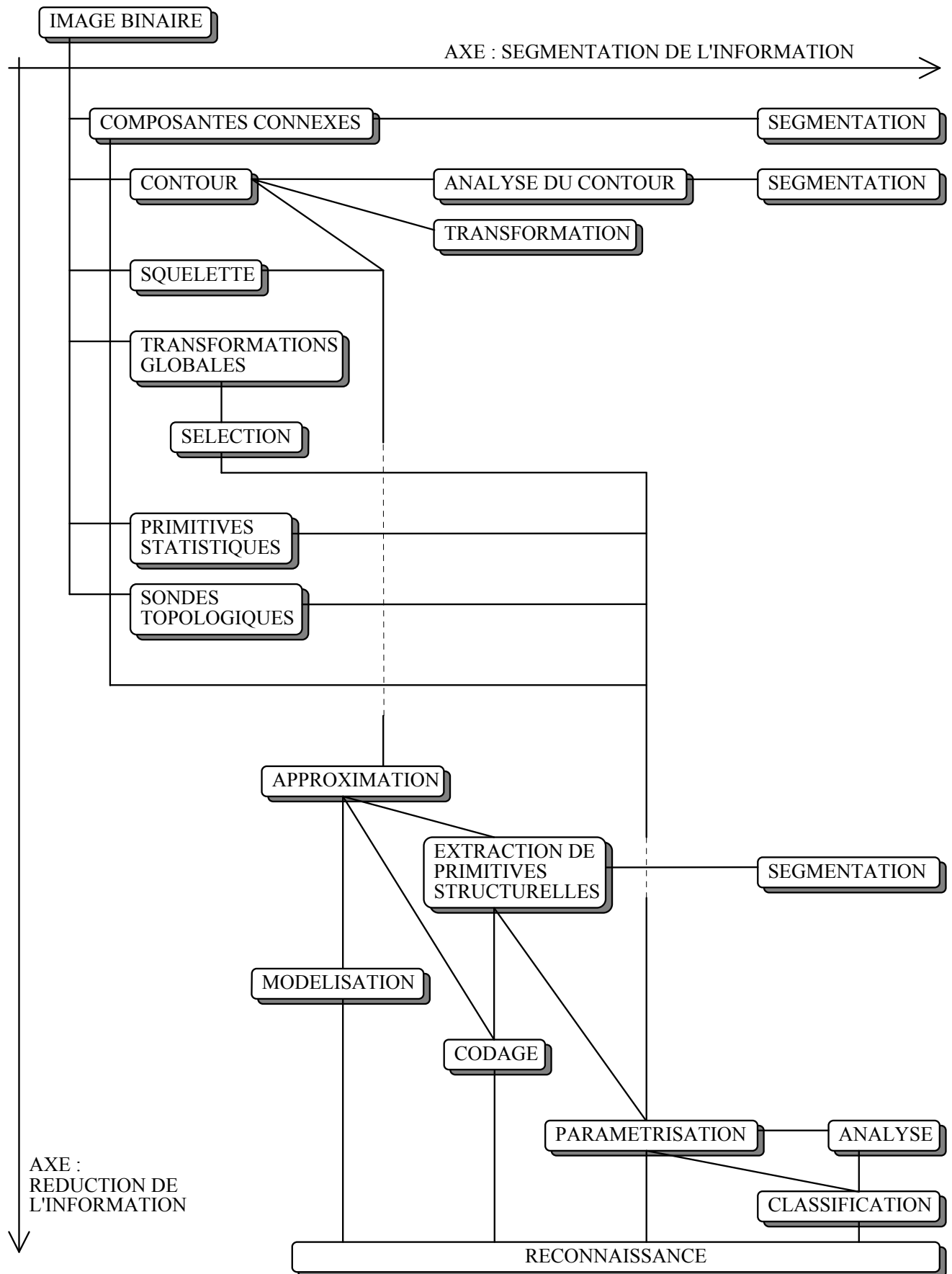
### 3.2.3. Les étapes de l'extraction des primitives

Il peut y avoir plusieurs étapes intermédiaires pour parvenir à l'objectif fixé. La première étape importante de la reconnaissance est le changement de représentation de l'information. A partir de l'image numérique brute, qui est la représentation originelle, on relève quatre autres représentations possibles de l'information, ce sont :

- 1°) L'extraction des composantes connexes.
- 2°) L'extraction du contour.
- 3°) La détermination du squelette (au sens large du terme).
- 4°) Les transformations mathématiques qui opèrent globalement sur l'image.

On le voit, le vocabulaire lié à l'extraction des primitives est riche et varié. Aussi, afin d'ordonner toutes les étapes suivant la progression normale du traitement de l'information et en les situant les unes par rapport aux autres, nous avons regroupé sur un schéma l'ensemble de ces opérations suivant deux axes indépendants qui représentent la **segmentation** et la **réduction** de l'information. La réduction vise à l'élimination de l'information redondante ou du bruit. Son axe va dans le sens de la reconnaissance (l'identification est la réduction ultime de l'information), ce qui n'est pas le cas de la segmentation.

Si la visualisation du schéma en trois dimensions avait été aisée, nous aurions ajouté un troisième axe, indépendant des deux autres, intitulé la **transformation** (ou le changement de représentation) de l'information. En effet, cet acte n'est pas une segmentation et ne conduit pas obligatoirement à une réduction de l'information. Pratiquement, nous avons projeté cet axe dans le plan (segmentation, réduction).



**Fig 3.1 :** Les objectifs et les étapes de l'extraction des primitives

L'ensemble des échantillons d'écriture traités dans notre travail est constitué par des images numériques de type binaire. Or, dans le cas où les échantillons sont numérisés avec une caméra CCD ou un scanner multi niveau, les images obtenues sont de type niveaux de gris, ce qui nécessite souvent une étape préliminaire de binarisation et n'est pas sans poser le problème du choix du seuil lorsque le fond de l'image a un niveau uniforme. Certaines études [PETIER 91] montrent alors l'intérêt de pratiquer l'étape d'extraction des primitives directement sur l'image en niveaux de gris.

En revanche, lorsque le fond de l'image est uniforme, la binarisation n'entraîne pas une perte d'information significative, à condition que la résolution soit suffisante.

D'autre part, nous devons mentionner un problème entraînant une étape supplémentaire de segmentation, que nous n'avons pas fait figurer dans notre schéma, lorsqu'une information non signifiante pour la reconnaissance est superposée au texte à décrypter. Cette information peut être par exemple les barres obliques dans le cas des chèques ou encore un bruit quelconque. Le traitement consiste alors à extraire l'information utile du reste de l'image.

### **3.3 Conclusion**

Lors de l'extraction des primitives, l'objectif est de sélectionner l'information pertinente qui se trouvait noyée dans la masse de l'information brute acquise. Il s'agit d'éliminer plus de bruit que d'information signifiante afin d'augmenter la discrimination. Autrement dit, la question qui se pose est comment réduire sans être réducteur ?

Le dilemme qui se présente est alors le suivant : une trop grande approximation entraîne une perte d'information. En effet, certaines lettres étant plus informantes que d'autres, il est nécessaire de conserver une description précise du mot. Or, la complexité du système de reconnaissance augmente avec le niveau descriptif des primitives.

Pour résoudre ce problème de complexité, la segmentation du mot en lettres semble la seule issue qui nous permette d'utiliser des primitives évoluées. Il est inutile de rappeler la difficulté de segmenter les mots du fait de l'interdépendance du tracé des lettres.

D'une façon plus générale, le résultat de chaque extraction de primitives peut être considéré comme une forme particulière de segmentation ou de réduction de l'information. La segmentation implique toujours le risque de fragmentation d'une unité d'information, qu'il s'agisse de la segmentation du mot en lettres ou bien de l'extraction de primitives sur la lettre. D'autre part, la réduction occasionnée par l'approximation constitue une sélection souvent arbitraire de l'information. Dans ces deux cas, il y a donc inévitablement un risque de perte d'information.

# Extraction des paramètres et méthodes de reconnaissance

I - Une Famille de représentation pour la reconnaissance hors-ligne de l'écriture.

### 4. I.1-Introduction

L'extraction de l'information est la première étape de la reconnaissance. Elle est réalisée à l'aide de l'extraction des primitives, qui sont des informations élémentaires. La primitive est l'outil de la reconnaissance.

L'objectif de l'étape de l'extraction de l'information est la sélection de l'information pertinente qui se trouve noyée dans la masse de l'information brute acquise. La problématique de cette étape a pour origine le risque de perte d'information signifiante pour la reconnaissance. La minimisation de ce risque est conditionnée par deux dilemmes liés chacun à un paradoxe de causalité. Il s'agit du dilemme de la réduction et du dilemme de la segmentation.

### 4. I.2-Prétraitement :

#### 4. I.2.1 Numérisation :

La numérisation d'une image se fait par des cartes d'acquisition équipées de convertisseurs analogiques/numériques. La manière la plus simple pour numériser l'image est alors de diviser la plage totale de la variation de la luminosité du noir au blanc, en nombre d'intervalles égaux et adjacents. Le codage d'un pixel à P éléments permet de définir 2P niveaux de gris entre le noir et le blanc. La numérisation de l'image fait réduire l'effet de bruit qui est accompagné avec l'opération d'acquisition.

Le seuillage de l'image permet d'obtenir une image binaire selon son histogramme ou sa densité lumineuse. Elle permet d'obtenir une image binaire "0" pour fond et "1" pour les caractères. Les systèmes modernes sont équipés de circuits capables de réaliser cette opération.

#### 4. I.2.2 La perception des lignes :

Certains processus de reconnaissance exigent l'extraction de plusieurs paramètres fondamentaux comme l'épaisseur moyenne du trait, la définition de la longueur comprise entre la ligne de base inférieure et la ligne de base supérieure, ainsi que la localisation précise des hampes et des jambages peuvent être extrait de façon plus robuste si cette extraction est envisagée avant d'isoler le mot de son-contexte, le principe retenu consiste à compter dans une certaine bande de l'image le nombre de transitions noir/blanc de chaque ligne de pixels, un profil de transition est obtenu (*Figure 4.1*), lorsque le texte est écrit horizontalement, après un lissage de ce profil un seuillage fait apparaître les positions des lignes de base.

Ainsi, il est possible de déterminer le début de chaque ligne et sa direction ou encore la hauteur de l'interligne, "zone intermédiaire composé entre la ligne de base inférieure et la ligne de base supérieure". Avant de faire la séparation des lignes des autres par la procédure du contour partiel, on applique d'abord l'opération d'amincissement.



Figure 4.1 : Exemple d'un profil des transitions.

#### 4. I.2.3- Amincissement et Redressement :

L'amincissement et le redressement du tracé permettent d'uniformiser et de dépersonnaliser l'écrit. La procédure d'amincissement qui a pour but principal d'éliminer l'épaisseur du tracé et de ne garder que le squelette, qui est bien adapté, pour un suivi de contour sans problème. L'amincissement est un prétraitement qui offre la possibilité de simplifier la forme de l'image.

#### 4. I.2.4-Suivi du contour partiel :

Le principe du suivi de contour partiel consiste à tracer un contour supérieur et un contour inférieur de la ligne de texte initialisé à l'aide de profil de transition, ces contours ne s'écartent de la gauche de direction trouvée que pour contourner un éventuel « obstacle » formé par le texte lui-même, surtout les hampes pour le contour supérieur, et les jambages pour le contour inférieur.



Figure 4.2 : Principe de l'algorithme de suivi de contour partiel.

tion, et permet en plus la détection des hampes et des jambages qui sont parmi les caractéristiques visuelles les plus importantes dans le processus de reconnaissance humaine et qui seront des éléments essentiels pour la classification des mots.

|   |   |   |
|---|---|---|
| 3 | 2 | 1 |
| 4 | P | 0 |
| 5 | 6 | 7 |

Figure 4.3 : Le code Freeman

### **Algorithme de suivi de contour partiel (Cas du contour inférieur : Cinf)**

*Paramètres d'entrée » : d-inf : droit d'appui du contour inférieur  
d : direction courante.*

*Début*

*Répéter*

*D=0*

*Pas-de-chemin est vrai ;*

*Tant que d<8 et pas-de-chemin faire*

*Début*

*Si pas-de-pixel (d) et direction (d) autorisé alors.*

*Début*

*Le point testé est mémorisé comme point du contour ; pas-de-chemin est faux*

*Aller à ce point ;*

*Fin ;*

*Si pas-de-chemin alors*

*Début*

*Autoriser toutes les directions ;*

*Noter la présence d'une boucle ouverte de jambage ;*

*Fin*

### **Algorithme de gestion des interdictions**

*Paramètre d'entrée :*

*D : direction courante du code de FREEMAN ;*

*Début ;*

*Suivant d faire.*

*Début*

*Cas 0 ou 1 : Si dans boucle ouverte de jambage*

*Alors direction 0 à 4 interdits ;*

*Sinon direction 0 à 4 autorisé*

*Cas 2 ou 3 : direction 0 à 4 autorisées ;*

*Cas 4,5 ou 6 : direction 0 à 2 interdites ;*

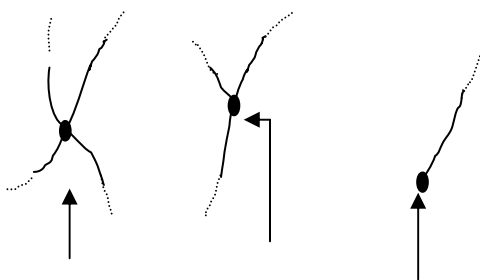
#### 4. I.3- Segmentation en graphèmes :

L'écriture manuscrite peut être considérée comme une succession d'unités graphémiques : les *allographes*. Les allographes sont propres à chaque scripteur : ce sont les représentations de ses propres graphèmes. Les graphèmes sont reliés entre eux par des liaisons personnalisées pour former les mots. Les allographes sont constitués par un enchaînement de traits de plume différents que l'on appelle les *allographes*, de telle sorte qu'il faut considérer le graphisme de chaque mot comme une succession d'allographes. Ces derniers sont donc les éléments de bases permettant l'édification des mots : ils obéissent à des lois comme n'importe quelle trace écrite. Des allographes différents, tracés par un même scripteur, contiennent souvent des allographes semblables : (par exemple ceux qui correspondent aux jambages des lettres).

Ainsi, la détection de l'allure générale des principaux traits doit permettre d'obtenir une bonne caractérisation de l'écriture. La méthode proposée basée sur la modélisation en graphème, ne demande aucun processus de segmentation préliminaire pour localiser les lettres. Son objectif est de fournir au processus de reconnaissance une description cohérente comprenant des segments de lettres.

En considérant l'écriture comme succession de segments, qui on été choisis comme des parties importantes du mot.

Cette représentation est basée sur un ensemble de points critiques ou d'ancrages. Dans ce travail, un segment "graphème " est défini comme une courbe continue représentée par une chaîne de coordonnées  $(X_i, Y_i)$  où l'ordre de ces coordonnées représente une approximation du mouvement du stylo pendant l'écriture, le premier et le dernier point sont appelés respectivement point de départ et point de fin. L'extraction d'un segment s'effectue par une recherche d'un point d'extrémité ou point de fin. La *Figure 4.4* montre les points critiques "Intersection, branchement, fin ".





courant, jusqu'à l'obtention d'un point de fin. Un point de fin d'un segment est sélectionné parmi les points critiques :

- a) Un point de branchement.
- b) Un point d'intersection.
- c) Un point de fin.
- d) Un point avec un changement brusque de la courbe au voisinage de la zone centrale " milieu d'interligne ".

Le processus précédent est répété jusqu'à l'extraction de tous les segments formant le mot. La Figure (4.6) donne un exemple du processus de segmentation d'un mot arabe composé d'une seule composante connexe.

#### 4. I.4- Les Différents Groupes de Segments

Les différents segments (graphèmes) qui peuvent être générés par cette méthode sont divisés en cinq groupes :

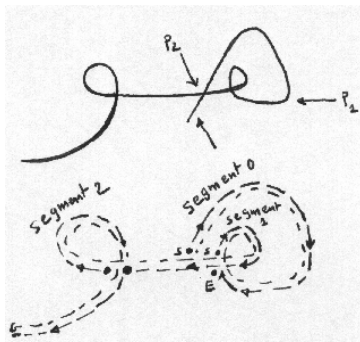
**GROUPE 1 (GR1)** : points définis comme des segments isolés avec une longueur très petite.

**GROUPE 2 (GR2)**: Hamza défini comme un petit zigzag isolé.

**GROUPE 3 (GR3)**: segments avec boucles.

**GROUPE 4 (GR4)**: segments sans boucle et qui se terminent avec des points de branchement ou d'intersection (connectés avec les segments suivants).

**GROUPE 5 (GR5)**: segments sans boucles et qui se terminent avec un point de fin.



**Figure 4. 6 :** Exemple des segments générés à partir d'un mot arabe composé de deux caractères.

La modélisation de l'écriture en graphèmes permet de transformer les images des mots manuscrits en séquences de segments. Ces segments ou portions d'images sont parfois réduits à des vecteurs de taille limitée, voire à des *symboles discrets*. Cette représentation schématique des images peut être traitée par des

modèles de Markov ou par d'autres stratégies élaborées pour faire la reconnaissance.

#### **4. I.5-Les caractéristiques extraites des images binaires**

Une image binaire est obtenue par une binarisation globale ou locale qui s'adapte à l'entrée image à niveau de gris. En général, la segmentation des caractères est réalisée en isolant des composants. Cependant, quelques caractères peuvent être collés au ayant des chevauchements.

Trier et al [Trier 95] ont développé une méthode basée sur l'analyse topographique, dans laquelle ils ont intégré la binarisation et la segmentation. Cette méthode donne une meilleure performance, la représentation d'une trame binaire d'un caractère est une simplification de la représentation à niveaux de gris.

La fonction de l'image  $Z(x, y)$ , prend "0" et "1" au lieu de 256 valeurs; cela veut dire que toutes les méthodes utilisées pour la représentation à niveaux de gris sont applicables à la représentation de la trame binaire.

Un caractère binaire peut être converti aux autres représentations tels que les contours externes des caractères, les profils ou le squelette du caractère. Les caractéristiques peuvent être extraites de chacune de ces représentations. L'objectif de ses conversions est la conservation de l'information pertinente au niveau de la forme du caractère.

##### **4. I.5.1- Répartition en Zone**

Bokser [Bokser 92] décrit un système OCR qui utilise la répartition en zone sur les caractères binaires, le système a été conçu pour reconnaître les caractères imprimés ou écrits par machine. Mais le problème est que toute police de caractère non décoratif peut être dégradée. Par exemple, la méthode de la répartition en zone a été utilisée pour calculer le pourcentage des pixels noirs dans chaque zone.

Kimura et Shridhar [Kimura 91] ont utilisé la répartition en zone sur le contour de la courbe. Dans chaque zone, les symboles de lignes des contours entre les pixels noirs ont été groupés par orientation **horizontale (0°)**, **verticale (90°)** et les deux orientations diagonales (45°, 135°), le nombre de segments de la ligne de chaque orientation a été compté.

Takahachie [Takahachie 91] a aussi utilisé des histogrammes de l'orientation des zones, des tranches verticales, horizontales et diagonales sont utilisées comme zone; les orientations ont été extraites des contours intérieurs.

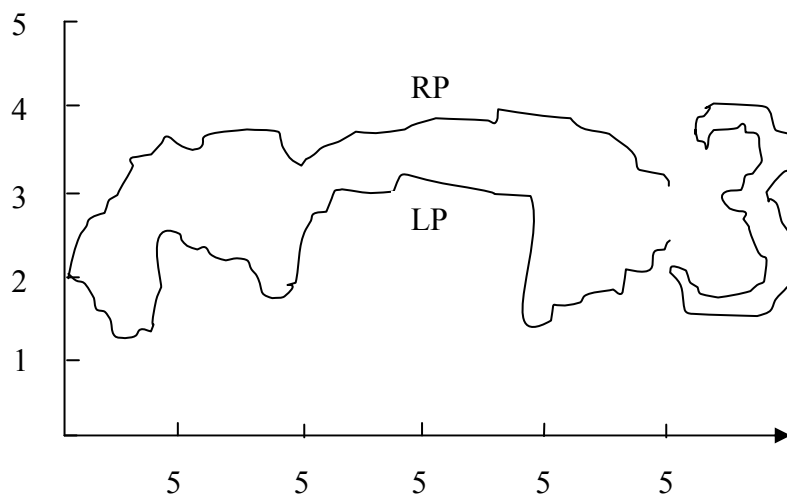
#### 4. I.5.2- Les profils des contours

La motivation derrière l'emploi des profils du contour est que chaque demi-contour (*Figure 4.7*) peut être rapproché par une fonction discrète d'une variable spéciale  $x$  ou  $y$ , alors, les caractéristiques peuvent être extraites de fonctions discrètes, nous pouvons utiliser des profils verticaux ou horizontaux, ils peuvent être des profils externes ou internes.

Dans un classifieur toutes les caractéristiques associées à un caractère sont dérivées depuis les profils de ses contours externes. Ces profils sont décrits par :

- a) Les profils gauches  $\{Lp(k), k=1,2,\dots,Np\}$ , collection de distance des profils gauches.
- b) Les profils droits  $\{Rp(k), k=1,2,\dots,Np\}$ , collection de distance des profils droits.

*Les profils gauches et droits pour le caractère 3 sont donnés par la figure 4.7.*



**Figure 4.7 :** des profils gauches et droits du caractère 3.

Les profils sont utilisés comme caractéristiques globales en plus deux autres caractéristiques globales sont définies par la différence des profils gauches et droits :

c)  $LDIF(k) = LP(k) - LP(k - 1)$ ,  $k=2,3,\dots,NP$ .

d)  $RDIF(k) = RP(k) - RP(k - 1)$ ,  $k=2,3,\dots,NP$ .

### 4.1.5.3-Les moments de projections

Les moments de projections sont extraits à partir des projections verticale et horizontale. Notons au passage que les moments et les moments invariants ont été utilisés largement dans les modèles de reconnaissance. Par exemple, il est possible de définir les caractéristiques suivantes :

(1) et (2) les mesures de *kurtosis* :

$$(1). KV = \frac{\mu_4^v}{(\mu_2^v)^2} \quad \text{et} \quad (2). KH = \frac{\mu_4^h}{(\mu_2^h)^2}.$$

Où  $\mu_r$  est défini par  $\mu_r = \sum_i (x_i - \bar{x})^r g(x)$ .

Et  $g(x)$  est  $v(x)$  ou  $h(x)$  définie par les équations suivantes :

$$h(i) = \sum_j g(i, j) \quad ; \quad v(j) = \sum_i g(i, j).$$

Où  $g(i, j)$  vaut Un ou Zéro,  $i$  l'indice des lignes et  $j$  est l'indice des colonnes.

Les indices supérieurs  $v$  et  $h$  indiquent que les mesures sont dérivées à partir des projections verticales et horizontales respectivement.

(3) et (4) sont les mesures de **skewness**.

$$(3) : SV = \frac{\mu_3^v}{(\mu_2^v)^{1.5}} \quad \text{et} \quad (4) : SH = \frac{\mu_3^h}{(\mu_2^h)^{1.5}}.$$

(5) et (6) sont les mesures de **skewness et kurtosis normalisées**.

$$(5) : NV = \frac{\mu_3^v}{(\mu_4^v)^{0.75}} \quad \text{et} \quad (6) : NH = \frac{\mu_3^h}{(\mu_4^h)^{0.75}}.$$

(6) et (8) et (9) les mesures des **rapports** entre la projection verticale et horizontale, pour le même caractère.

$$(7) : LW = \left| \frac{\mu_1^v}{\mu_1^h} \right|, (8) : VR = \frac{\mu_2^v}{\mu_2^h}, \text{ et } (9) : VV = \frac{\mu_4^v}{\mu_4^h}.$$

#### 4. I.5.4 Les Paramètres en Barres

Les paramètres en barres "bar-features" ont été utilisés avec succès dans plusieurs travaux, ils sont calculés sur des images binaires des caractères isolés qui ne nécessitent pas de normalisation. Premièrement, huit images de paramètres sont générées qui correspondent aux quatre directions (Est(e), Nord-Est (ne), Nord (n) et Nord-Ouest (nw)) pour le fond et pour le caractère. Chaque image de paramètre a une valeur entière dans chaque site représentant la longueur de la barre la plus longue qui existe dans la même direction

Un algorithme récursif, à deux passes est utilisé pour calculer les images paramètres. Dans la passe directe, l'image est scannée de haut en bas dans le sens gauche-droite. Dans chaque point, chacune des images paramètres de fond et du caractère est adaptée. Dans la deuxième passe de retour, le maximum est propagé dans le sens inverse, c'est à dire de haut en bas dans le sens droite-gauche. Un pseudo code montrant comment calculer ces paramètres est le suivant :

```

/*---- passe directe ----*/
FOR i=1,2,...., nrows DO
  FOR j=1,2,...., ncols DO
    { e( i, j) = e( i, j-1)+1
      ne ( i, j)= ne (i-1, j+1)+1
      n ( i, j)= n (i-1, j)+1
      nw ( i, j)= n (i-1, j-1)+1
    }
/*---- passe de retour ----*/
FOR i =nrows, nrows-1,....,1 DO
  FOR j=ncols, ncols-1,....,1 DO
    { e(i, j) = max(e(i, j), e(i,j+1))
      ne(i, j) = max( ne ( i, j), ne( i+1, j-1))
      n(i, j) = max( n(i, j), n(i+1, j))
      nw(i,j) = max( nw ( i, j), nw( I+1, j+1))
    }

```

où nrows, ncols représentent les nombres de lignes et de colonnes respectivement

Les paramètres sont calculés à partir des images paramètres en utilisant des zones en chevauchement pour assurer un certain degré de lissage. Quinze zones rectangulaires arrangées en cinq lignes avec trois zones sur chaque ligne; chaque zone est approximativement de taille  $h/3 \times w/2$  où h et w sont respectivement la hauteur et la largeur des images. Les coins de haut à gauche des zones sont approximativement aux positions  $\{(r, c) \mid r=0, h/6, 2h/6, 3h/6, 4h/6 \text{ et } c=0, w/4, 2w/4\}$ . Les approximations sont dues au fait que toutes les tailles d'images ne sont pas exactement divisibles par quatre ou six. Les valeurs dans chaque zone des images de paramètres sont sommées. Les sommes sont normalisées entre zéro et un en divisant par le maximum possible de la somme dans une zone. Le vecteur paramètre est de dimension  $15 \times 8 = 120$ . Plus précisément, soient  $f_1, f_2, \dots, f_8$  les images des paramètres associées avec une forme en entrée et soit  $Z_i$  dénotes une zone rectangulaire de taille  $h/3 \times w/2$  avec le coin de haut à gauche est  $(r_0, c_0)$ .

La valeur du  $k^{\text{ième}}$  paramètres associés à la zone  $Z_i$  est :

$$Z_{ik} = \left(\frac{1}{N}\right) \sum_{r=r_0}^{(r_0 + \frac{w}{2})} \sum_{c=0}^{(c_0 + \frac{h}{3})} f_k(r, c).$$

où  $N = w$  si  $f_k = e, ne$  ou  $nw$   
 $=h$  si  $f_k = n$ .

Le vecteur de 120 valeurs ( $Z_{ik} : i= 1,2,\dots,15$  et  $k =1,2,\dots,8$ ) sont les paramètres en barres (" bar-features ") qui sont utilisés comme entrées aux réseaux de neurones de reconnaissance de caractères.

## 4. I.6-Les Caractéristiques Extraites des Contours

### 4. I.6.1-Descripteurs de Fourier

Des descripteurs de Fourier ont été développés par Zahn et Roskies et Graulund [Zahn 72] pour décrire des formes planes (pièce de fonderie, chromosomes, caractères, ...etc.). Dans cette méthode la différence angulaire  $A_\phi$  entre deux segments consécutifs de ligne sur les contours est mesurée à chaque pixel le long des contours ; les descripteurs suivants peuvent être extraits par les équations suivantes :

$$a_n = -\frac{1}{n\pi} \sum_{k=1}^n \Delta\phi_k \sin \frac{2\pi n t_k}{T}$$

$$b_n = \frac{1}{n\pi} \sum_{k=1}^n \Delta\varphi_k \cos \frac{2\pi n t_k}{T}$$

où  $T$  est la longueur de la courbe,  $t_k$  est la longueur accumulée entre le point courant et le point de départ jusqu'au point  $p_k$  et  $\Delta\varphi_k$  est l'angle entre les vecteurs  $[p_{k-1} p_k]$  et  $[p_k p_{k+1}]$ .

L'invariance de la rotation peut être obtenue en transformant les coordonnées polaires ; les amplitudes sont indépendantes de la rotation, par contre la phase dépend de la rotation.

$$A_n = \sqrt{a_n^2 + b_n^2} \quad , \quad x_n = \tan(a_n/b_n) .$$

Graulund [GRA 72] utilisait un nombre complexe  $Z(t)=x(t) - y(t)$ , dénoté les points sur le contour alors les contours peuvent être exprimés comme des séries de Fourier

$$Z(t) = \sum_{n=-r}^r a_n e^{j2\pi n t T} \quad , \quad \text{où} \quad a_n = \frac{1}{T} \int_0^T Z(t) e^{-j2\pi n t T} dt$$

#### 4.1.6.2- Codage des Contours

Le codage des contours est une technique exprimant les frontières discrètes de l'image du caractère par une séquence de codes de la chaîne de Freeman spécifiant la direction de déplacement d'un vertex à un autre. Notons également que la technique de zonage des courbes des contours, souvent employée sous la forme d'un calcul des histogrammes directionnels, permet d'affiner la représentation des caractères à représentations iconiques proches (caractéristiques globales similaires). En effet, la forme d'un caractère manuscrit arabe varie en fonction de sa taille, du style et de la fonte. Par conséquent, des paramètres locaux sont nécessaires pour affiner la représentation des caractères. Les techniques de codage des contours offrent de bons paramètres de discrimination. Dans ce processus, le rectangle englobant le contour externe de la partie primaire du caractère traité est divisé en 2x2 zones rectangulaires. Dans chaque zone, un histogramme local des codes de chaîne est calculé. Les distributions des huit directions dans les quatre zones du caractère primaire forment un ensemble de 8x4 paramètres locaux et 8 paramètres globaux. Les différents paramètres sont calculés comme suit :

**Paramètres locaux :**

$$F_{ij} = \frac{S_{ij}}{N_i S_j} \text{ pour } i = 1, 2, 3, 4 \text{ et } j = 0, 1, \dots, 7$$

où  $S_{ij}$  = nombre des composantes avec pente  $j$  à partir de la sous image  $i$ ,

$N_i$  = nombre de composantes à partir de la sous image  $i$  et

$$s_j = \max_i \left( \frac{S_{ij}}{N_i} \right)$$

Paramètres globaux :

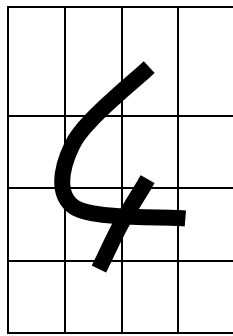
$$F_j = \frac{\sum_i S_{ij}}{\sum_i N_i} \quad \text{Pour } i=1, \dots, 4 \text{ et } j=0, 1, \dots, 7$$

#### 4.I.7- Représentation par le squelette

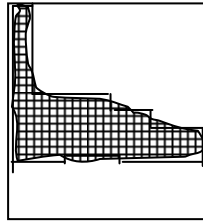
L'extraction du squelette désigne, au sens large du terme, un ensemble de techniques visant à représenter les formes à l'aide d'une structure fine continue qui passe à égale distance des contours. La squelettisation est une opération souvent utilisée en modélisation ou en représentation de l'écriture manuscrite, car l'écriture est assimilable à un long ruban replié sur lui-même et d'épaisseur constante. Le modèle à l'origine de la production de l'écriture est un modèle linéaire sans épaisseur, de sorte que la représentation par le squelette est naturelle et d'aspect proche de l'écriture. En ce sens, elle est proche de la représentation "en-ligne" ("on-line"), sauf en ce qui concerne l'ordre temporel du tracé ainsi que les "barbules" qui ne sont pas significatives du tracé.

##### **Caractéristiques extraites du squelette**

Le squelette du caractère est obtenu en amincissant la représentation de la trame binaire par des algorithmes d'amincissement. Un graphique des caractères peut être dérivé du squelette en l'approximant par plusieurs segments de lignes droites et des points de jonction. Les arcs peuvent être utilisés pour les parties courbées du squelette.



(a)

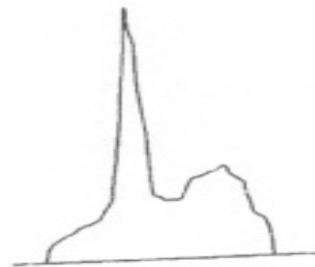
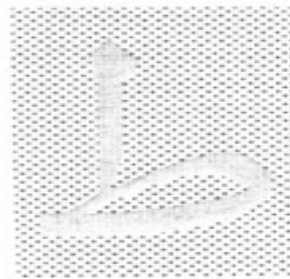
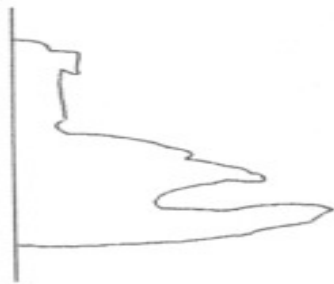


(b)

| Direction | Nombre |
|-----------|--------|
| 0° (-)    | 22     |
| 45° (/)   | 2      |
| 90° ( )   | 7      |
| 135° (\)  | 5      |

(c)

**Figure 4.8 :** Exemple des caractéristiques extraites (a) le caractère normalisé (b) la zone des codes chaîne (c) la direction d'histogramme



**Figure 4.9 :** Projections du caractère THA (a) verticale (b) horizontale.

## Discussion

Avant de sélectionner une méthode d'extraction des caractéristiques spécifiques, il est nécessaire de considérer le système global de la reconnaissance; pour quels types de caractères le système est dédié? est-ce que des caractères variables appartiennent à la même classe? Quel est le type des images (à niveaux de gris ou binaires) ? Quelle est la résolution recherchée? Quel est le prix attendu du système,

de telles questions ont besoin d'être répondues pour faire une sélection objective de la méthode d'extraction des caractéristiques appropriées. En général, une seule méthode d'extraction de caractéristiques n'est pas suffisante pour obtenir le bon pouvoir de discrimination.

#### **4. I.8. Conclusion**

Lors de l'extraction des primitives, l'objectif est de sélectionner l'information pertinente qui se trouvait noyée dans la masse de l'information brute acquise. Il s'agit d'éliminer plus de bruit que d'information signifiante afin d'augmenter la discrimination. Autrement dit, la question qui se pose est comment réduire sans être réducteur ?

Le dilemme qui se présente est alors le suivant : une trop grande approximation entraîne une perte d'information. En effet, certaines lettres étant plus informantes que d'autres, il est nécessaire de conserver une description précise du mot. Or, la complexité du système de reconnaissance augmente avec le niveau descriptif des primitives.

Pour résoudre ce problème de complexité, la segmentation du mot en lettres semble la seule issue qui nous permette d'utiliser des primitives évoluées. Il est inutile de rappeler la difficulté de segmenter les mots du fait de l'interdépendance du tracé des lettres.

# Application : Reconnaissance par modèles de Markov cachés et réseaux de neurones

### 5.1 Introduction

Dans le domaine de la reconnaissance des formes, la définition d'un facteur de forme (ou représentation) est un problème fondamental qui relève entièrement de la science de l'expérimentateur. Évidemment, le cas de la reconnaissance hors-ligne de l'écriture manuscrite qui repose uniquement sur l'étude de l'image du mot n'y échappe pas. Les difficultés inférentes à la reconnaissance hors-ligne de l'écriture sont multiples.

Devijver et Kittler [Devijver 82] définissaient l'extraction des caractéristiques comme le problème «d'extraire les informations les plus pertinentes à partir des données pour une classification dans le sens de minimiser la variabilité des critères d'intra-classe, tout en augmentant la variabilité des critères d'inter-classe», donc il devient claire que les différentes méthodes d'extraction des caractéristiques qui accomplissent cette exigence dépendent des problèmes traités et des données disponibles.

Les caractéristiques extraites doivent être des invariants aux distorsions attendues, et aux variations que les caractères peuvent subir dans une application spécifique.

Par exemple nous devons considérer si les caractères sont reconnus ?, S'ils sont manuscrits ou imprimés ? Et à quel degré sont ils dégradés ? Aussi est il nécessaire de définir plusieurs classes pour caractériser les caractères qui peuvent être écrits de deux manières ou de deux styles plus distincts comme par exemples "4" et " Ч ", " م " et " م ".

L'établissement d'une taxonomie des représentations et des méthodes d'extraction des caractéristiques dépend essentiellement de la forme des caractères; images à niveaux de gris, images binaires, contours et squelettes.

Les modèles de Markov cachés sont aujourd'hui couramment employés en reconnaissance hors ligne de l'écriture manuscrite. Dans certaines de ces approches, les images des mots sont transformées en séquences de segments d'image au moyen d'une procédure de segmentation explicite. Ces segments sont ensuite transmis à un module chargé d'estimer la probabilité selon laquelle chaque segment apparaît lorsque l'état correspondant de la chaîne de Markov est un certain état  $e$  (probabilités d'émission). Les probabilités du modèle sont généralement optimisées selon le principe du maximum de vraisemblance.

Dans ce chapitre, nous proposons de mettre en œuvre deux techniques de reconnaissance basées sur une modélisation de l'écriture arabe en graphèmes. La première technique utilise le traitement d'image et la programmation dynamique afin de calculer des scores de reconnaissance. Les réseaux de neurones qui sont employés dans l'identification des graphèmes permettent d'introduire la notion d'ambiguïté entre les graphèmes. La deuxième technique utilise des HMM. Notons que le fait d'introduire la notion de graphème permet d'obtenir un système hiérarchique composé de deux niveaux ; le premier niveau offre une liste de mots candidats en se basant sur le nombre de graphèmes (taille du mot). Dans le deuxième niveau s'effectue la reconnaissance et le post-traitement.

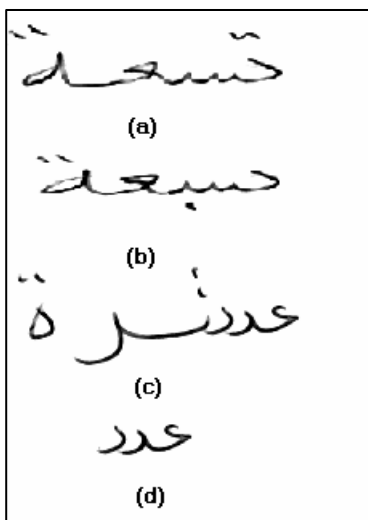
## **5.2. Reconnaissance des graphèmes par des réseaux de neurones**

### **a) Structure des réseaux**

Des réseaux séparés sont utilisés pour les différents groupes de graphèmes, désignés par GR3, GR4 et GR5 (Figure 5.1). Tous les réseaux sont entraînés en utilisant l'algorithme de retro-propagation. Chacun des réseaux possède des couches; une d'entrée, une de sortie et deux cachées. Chaque unité cachée ou de sortie possède un biais. Par exemple dans le cas des paramètres en barre, chaque réseau à 120 unités d'entrées, 65 unités dans la première couche cachée, 30 dans la seconde couche cachée, et  $M$  unités de sortie ( $M=10$  pour GR3,  $M=13$  pour GR4 et  $M= 11$  pour GR5) ; une pour chaque classe. Les unités de sortie sont des fonctions sigmoïdes dont les valeurs varient de  $-0.4$  à  $0.4$  est utilisée.

|     |                                |
|-----|--------------------------------|
| GR1 | •                              |
| GR2 | ء                              |
| GR3 | و و و و و و و<br>م ه د         |
| GR4 | ر ر ر ر ر ر ر<br>ر ر ر ر ر ر ر |
| GR5 | ل ل ل ل ل ل ل<br>ا ز ز ز       |

*Figure 5.1 : Les différents groupes de graphèmes*



*Figure 5.2 : Exemple d'ambiguïté entre des mots manuscrits*

### **b) Calcul des sorties désirées**

Afin de donner aux réseaux de neurones les comportements désirés, nous avons construit un ensemble d'apprentissage de graphèmes. Dès lors, des images de graphèmes sont extraites à partir des images des mots. Les sorties désirées pour les réseaux 'CRISP' (ou Hard) sont obtenues en mettant la sortie désirée pour la vraie

classe à 0.4, et pour les réseaux flous, les sorties désirées sont calculées en utilisant l'algorithme flou des K plus proches voisins (**k-PPV**) proposé par Keller et al [Keller 85].

L'idée est d'assigner des degrés d'appartenance basés sur le pourcentage des images des différentes classes dans les k-PPV d'un ensemble d'apprentissage avec une distance euclidienne. Pour cela, il est possible d'utiliser deux variantes de l'algorithme [Keller 85], chacune d'elles avec une contrainte différente sur la sortie désirée de la vraie classe :

☞ La sortie désirée de la vraie classe ne doit pas être inférieure à la sortie désirée de n'importe quelle autre classe.

☞ La sortie désirée de la vraie classe est supérieure à certain seuil.

La première variante est une formulation de Keller [Gader 92], la seconde variante peut être vue comme une interprétation des degrés d'appartenance comme une *distribution de possibilité*; la possibilité pour qu'une forme soit utilisée comme la vraie classe est élevée mais nous savons qu'elle est utilisée au moins une fois comme représentative de cette classe. Cet algorithme est le suivant :

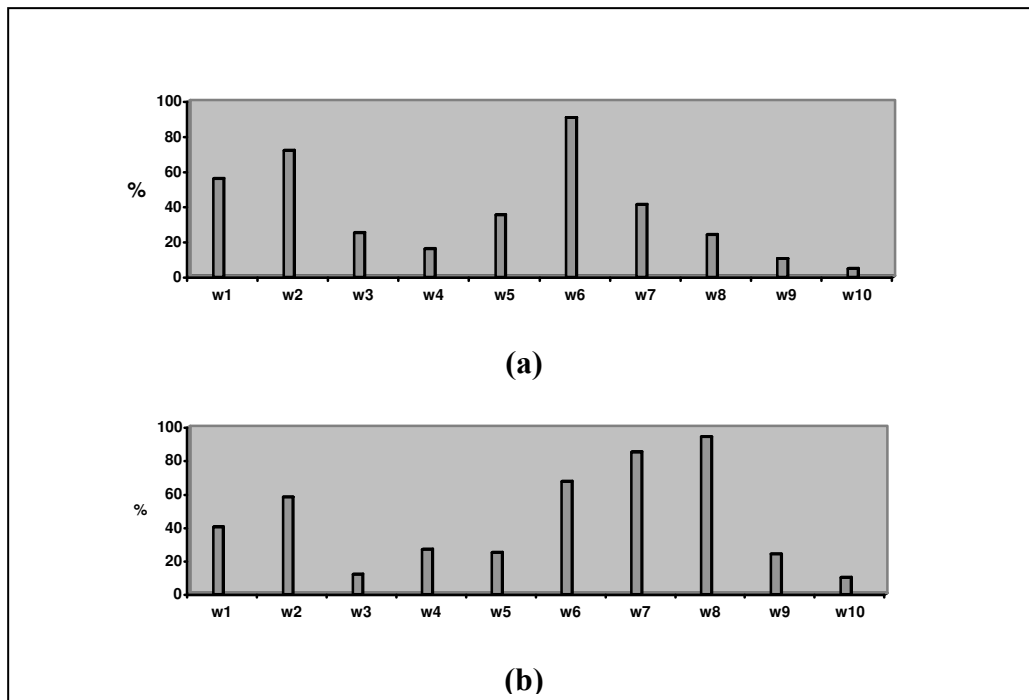
**Algorithme :**

- Calculer les paramètres d'attribut (par exemple les paramètres en barre) pour tous les échantillons de l'ensemble d'apprentissage.
  - Pour chaque graphème S de l'ensemble d'apprentissage.
    - Mettre la sortie désirée pour la vraie classe à 1.
    - Pour chaque autre classe C, mettre les sorties désirées à  $\mathbf{a} (N_c / k)$
- où :  $N_c$  est le nombre d'échantillons de la classe C parmi les k-PPV de S.
- $\mathbf{a}$  est un paramètre entre 0 et 1 permettant d'ajuster les degrés d'ambiguïtés.

Les sorties désirées sont ensuite transformées à l'intervalle  $[-0.4, 0.4]$  pour l'apprentissage.

Nous choisissons l'utilisation de 15 proches voisins d'un échantillon d'apprentissage avec comme métrique la distance euclidienne. Des exemples de

sorties désirées calculées par l'algorithme de Keller sont montrés dans la figure 5.3. Rappelons que la vraie classe est souvent la classe dont la sortie est la plus élevée.

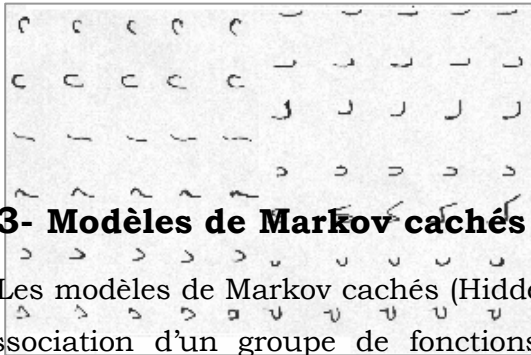


*Figure 5.3 : Quelques graphèmes du groupe GR3 et leurs degrés d'appartenance. Les vraies classes de ces graphème sont respectivement « ڨ » et « ڤ ».*

### **b) Résultats de reconnaissance des graphèmes**

L'ensemble d'apprentissage des graphèmes a été construit en relevant des graphèmes à partir des images de mots manuscrits. Les positions des graphèmes dans les mots sont identifiés manuellement, ensuite les graphèmes sont extraits et triés par classe. Les images des mots manuscrits sont obtenues à partir d'une base de données du département d'Electronique université d'Annaba.

Le résultat est un ensemble d'apprentissage formé de 50 images de graphème par classe. Des graphèmes échantillons sont montrés dans la Figure 5.4. Cependant, l'objectif est de reconnaître des mots; alors les tests doivent être effectués au niveau des mots.



**Figure 5.4 :** Des échantillons typiques de l'ensemble d'apprentissage des graphèmes.

### 5.3- Modèles de Markov cachés

Les modèles de Markov cachés (Hidden Markov model ou HMM) se produisent de l'association d'un groupe de fonctions de densité de probabilité d'émission de symboles et d'une chaîne de Markov qui peut être considérée comme un automate probabiliste [Rabiner 89]. Il se trouve deux classes principales de HMM : *La classe (modèle) ergodique* qui est un modèle dont les contraintes sur les connexions n'ont pas un lieu (c'est à dire où toutes les transitions d'un état vers un autre sont possible). La deuxième classe est *le modèle de type gauche-droite soumis* à des contraintes de transition (pas de retour arrière,...).

Les modèles de classe gauche-droite semblent bien adaptés pour modéliser les phénomènes continus qui ne nécessitent pas de retour-arrière. Pour cette raison, l'écriture et la parole avaient une dynamique qui s'intègre bien à ce dernier type de modèle.

Cependant, il faut définir les différents paramètres ou niveau de la structure du modèle (nombre d'états, classe de transitions,.....). Les choix concernant la structure exacte du modèle peuvent être identifiés de façon empirique relativement ; ainsi ils demeurent non facilement justifiables. Dans la reconnaissance de la parole, en précisément lorsqu'il s'agit de reconnaître le phonème, on utilise une topologie de modèle qui doit être tendu à se justifier par la structuration même du phonème. On veut, par exemple, modéliser au travers de la topologie du modèle, les phases stables et les phases transitoires du phonème.

Néanmoins, la structuration de ces modèles avait perdu sa justification en passant à la reconnaissance de l'écriture.

### 5.4- Reconnaissance de mots

La première méthode de reconnaissance de mots utilise le traitement d'image et la programmation dynamique. Les entrées sont des images binaires d'un mot et un lexique.

La programmation dynamique est utilisée pour trouver le meilleur chemin, qui est la meilleure séquence de segments pour trouver une chaîne donnée. La valeur d'un chemin est la somme des scores des segments dans le chemin des graphèmes dans la chaîne comme donnée par les activations des sorties des réseaux de neurones. Un schéma de l'algorithme ainsi décrit est donné pour illustrer son principe de fonctionnement. Le calcul du score se fait une seule fois pour chaque image mot. Pour chaque image mot, la recherche par programmation dynamique est faite une seule fois pour chaque chemin dans le dictionnaire associé à l'image.

Pour le chemin correct, plusieurs autres recherches doivent être faites avec la même façon.

**a) Optimisation par programmation dynamique :**

L'optimisation programmation dynamique prend une image d'un mot et une chaîne de primitives d'une image du mot et donne une valeur comprise entre "zéro" et "un" indiquant le degré avec lequel l'image du mot représente la chaîne. L'algorithme est implémenté en utilisant une approche matrice.

Les lignes de cette matrice correspondent aux graphèmes dans la chaîne. Les colonnes de la matrice correspondent aux primitives. Soit l'élément  $i, j$  le meilleur score "match" entre les  $i$  premiers graphèmes de la chaîne et les  $j$  premières primitives. Cette valeur doit être  $-\infty$  s'il y a pas de "match" légal.

Soit les primitives  $S_1, S_2, \dots, S_p$  et les graphèmes dans la chaîne sont  $C_1, C_2, \dots, C_w$ . Soit  $m(s, c) = \max \{ \text{la classe } c \text{ des activations des sorties du réseau étant donnée l'entrée } s \}$ . La valeur  $m(s, c)$  est la confiance pour que  $s$  représente  $c$ . L'élément de la matrice est calculé comme suit :

**Algorithme :**

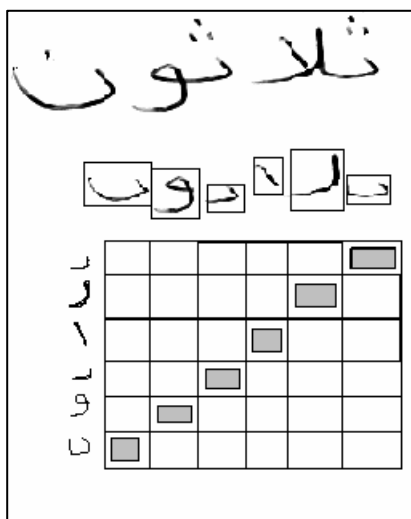
*Si  $i=1$ , (chercher parmi les premiers graphèmes) Alors*  
 *$V(i,j)=m(S_{1j}, C_j)$  pour tous  $j$  tels que  $S_{1j}$  est conforme.*  
 *$= -\infty$  autrement.*

*Si  $i>1$  Alors*

où  $S_{kj}$  représente l'union des segments primitives  $S_k \cup S_{k+1} \cup \dots \cup S_j$ .

Une union des segments est considérée comme conforme ou légale si les segments passent par une séquence de tests. Les tests reflètent une certaine mesure de proximité et de complexité.

La deuxième méthode de reconnaissance des mots utilise les modèles de Markov cachés décrits dans la section précédente. Dans ce cas un modèle est construit pour chaque mot du lexique dont les paramètres sont estimés par l'algorithme Forward-Backward de Baum-Welch et les scores de reconnaissance par l'algorithme de Viterbi [For73].



**Figure 5.5 :** Principe de calcul du score de reconnaissance par programmation dynamique.

### Remarque

Nous avons décrit deux méthodes de reconnaissance des mots manuscrits arabes basées sur la modélisation de l'écriture en graphèmes. La première supporte l'utilisation des degrés d'appartenance flous aux classes de graphème dans l'affectation des mesures de confidences, pour la reconnaissance des mots. La seconde est basée sur l'emploi de différents types de modèles de Markov cachés où la reconnaissance est obtenue par calcul de la probabilité a posteriori de la classe d'une forme. Ce calcul fait intervenir plusieurs termes qui, suivant certaines

hypothèses de dépendance liées à l'application traitées, peuvent se décomposer en probabilités conditionnelles élémentaires

Nous avons constaté que des représentations adéquates d'ambiguïté entre graphèmes peuvent être plus intéressantes que des taux de reconnaissance élevés. Pour cela des méthodes de mesure de succès de représentation d'ambiguïté entre les classes de graphèmes sont demandées. Ceci est un problème difficile et un cas spécial du problème d'estimation des fonctions des degrés d'appartenance. L'utilité d'une représentation d'ambiguïté doit être directement reliée au succès d'achèvement du but final, dans ce cas des taux de reconnaissance de mots élevés, donc ni les taux de reconnaissance des graphèmes ni l'erreur RMS sont capables de capter cette qualité.

Ainsi, l'application de certaines mesures spécifiques représente sans doute la voie la plus appropriée. Par exemple dans la reconnaissance de mots la meilleure mesure du succès est que l'algorithme de reconnaissance de mots atteigne des taux de reconnaissance élevés.

Un autre point important concernant la deuxième méthode et plus précisément le choix des modèles de Markov les plus appropriés (nombre d'état...etc.).

## **5.5- Résultats expérimentaux**

### **5.5.1 Collection des données**

*Construction De Base de données.*

Dans cette recherche de reconnaissance de mots, le rôle de la base de données des modèles est essentiel. En employant une grande base de données, deux aspects doivent être considérés un est le problème de mettre en application un bas coût, système de fichiers. Un disque compact-ROM sera une solution à ce problème. L'autre est récupération efficace des modèles intéressants de la grande base de données.

Une base de données a été construite afin d'examiner le système. Cette base de données se compose de 5000 mots de montant de chèque (pour 50 classes, 100 échantillons pour chaque classe de taille 139 Mo), 1000 échantillons pour les chiffres (10 classe, 100 échantillons de chaque classe de taille 22.9 Mo) et 14000 caractères dans ces différents formes début, milieu, fin, isolé (35 classes, 100 échantillons pour chaque classe de taille 400 Mo) rassemblés de 50 objets avec différent âge l'origine, et le niveau culturel. Les écrivains ont été invités à écrire l'alphabet dans son propre modèle sur une feuille blanche dans de petites places,

84% étaient droitiers, et 16% gauchers. Cette base est développée au département d'Electronique, Université de Annaba.

Le lexique utilisé dans ce projet est composé pratiquement des mots qui entre dans l'expression des montants littéraux des chèques postaux (de taille 48 mots). Pour cela, 1200 échantillons (25 pour chaque classe) ont servi à l'apprentissage, et le reste de 1200 échantillons (25 échantillons par classe) ont été utilisés pour l'évaluation des performances [Zermi 00].

### 5.5.2 Prétraitement

Les opérations de filtrage et de suppression des bruits sont obligatoires car le processus d'acquisition et de numérisation est accompagné souvent par des bruits provoquant des irrégularités, surtout au niveau des contours des caractères, qui ont un effet non désirable sur le système de reconnaissance. Toutefois, l'opération de filtrage nécessite une modélisation plus au moins rigoureuses des distorsions optiques subies par les images. Comme la plupart des systèmes d'OCR, nous avons adopté une stratégie de lissage simple qui consiste à filtrer l'image à niveaux de gris par la morphologie mathématique ; une opération d'ouverture par un élément structurant carré de taille 3x3 pixels.. L'opération de binarisation repose sur le calcul des moments de l'histogramme des niveaux de gris pour sélectionner un seuil optimal.

Dans le cas des deux méthodes opérant par **graphèmes**, le système tente, après l'étape de prétraitement, d'isoler les graphèmes en fenêtres rectangulaires et de les caractériser par un vecteur de paramètres caractéristiques pour les connaître par la suite. Les images sont normalisées à une hauteur de 64 pixels et en largeur (tableau 1). Le mot est classé à l'une des trois catégories :

| Taille du mot | Nombre de graphèmes |
|---------------|---------------------|
| Faible        | $\leq 4$            |
| Moyenne :     | $5 \div 9$          |
| Grande :      | $\geq 10$           |

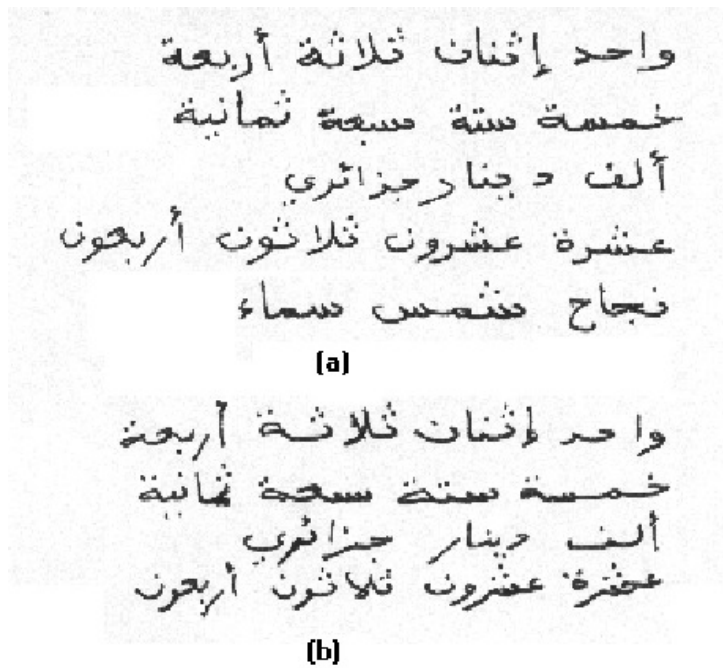
Dans le cas de la troisième méthode, les images binaires des mots manuscrits sont transformées en images à niveaux de gris en calculant au site (r,c) la somme des pixels blancs (valeur « un ») dans une fenêtre de taille N\*N (où N=7) centré sur le pixel r,c).

### 5.5.3 Les tests :

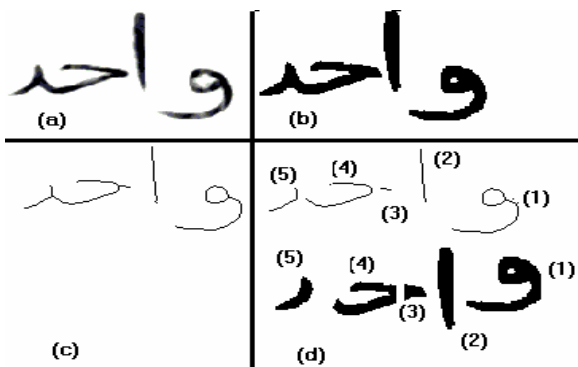
Pour la caractérisation des graphèmes deux types de vecteurs d'attributs ont été utilisés parmi la famille de représentations décrites dans le chapitre 4. Il s'agit **des paramètres du codage des contours** et **des paramètres en barre**. L'insertion des scores de reconnaissance des graphèmes dans deux algorithmes de reconnaissance des mots améliore sensiblement les taux de reconnaissance. Nous mentionnons en particulier l'emploi des degrés d'appartenance flous qui permet de tenir compte des ambiguïtés entre les graphèmes et par conséquent entre les mots. Nous avons évité de dresser des taux de reconnaissance d'erreurs et de rejet, souvent nécessaires pour évaluer la qualité de tout système de classification, puisque la sélection des paramètres de forme ou de discrimination relève entièrement de la science de l'expérimentateur et nécessite des bases de données très élaborées. Ceci dit, qu'il reste un grand travail à faire pour trouver la meilleure représentation d'une part et la meilleure architecture de classification même si le choix est orienté d'avance vers les nouvelles techniques (réseaux de neurones et logique floue).

Afin de montrer et d'illustrer les différentes étapes de traitement depuis l'obtention du mot manuscrit jusqu'à l'identification des graphèmes en passant par les procédures d'amincissement (squelettisation) et de détection, nous avons sélectionné et dressé quelques figures. La figure 5.6 donne des échantillons typiques de l'ensemble de données.

La figure 5.7 illustre le principe du processus de localisation des graphèmes en fenêtre rectangulaires pour les reconnaître par la suite.



**Figure 5.6 :** (a) Exemple de mots manuscrits correctement reconnus. (b) Exemple de mots manuscrits non reconnus.



**5.7 :** Les différentes étapes de traitement  
 (a) image à niveau de gris.  
 (b) image binaire améliorée. (c) image squelette.  
 (d) localisation des graphèmes en fenêtres rectangulaires.

En ce qui concerne la technique utilisant des réseaux de neurones pour la reconnaissance des graphèmes des groupes (GR3, GR4 et GR5), les résultats d'apprentissage sont montrés dans le Tableau 5.1. Ce tableau donne le pourcentage des graphèmes pour lesquels la sortie la plus grande est associée à la vraie classe (le taux de reconnaissance) et donne aussi l'erreur quadratique moyenne (pour root mean squared « RMS » error). Notons que le taux de reconnaissance est une mesure standard de performance des classifieurs mais le « RMS » est une autre mesure utilisée lors de l'apprentissage des réseaux de neurones. Il est évident que des taux de reconnaissance élevés et des RMS faibles sont bons. Les réseaux « crisp » atteignent des taux de reconnaissance supérieurs à ceux des réseaux flous avec des

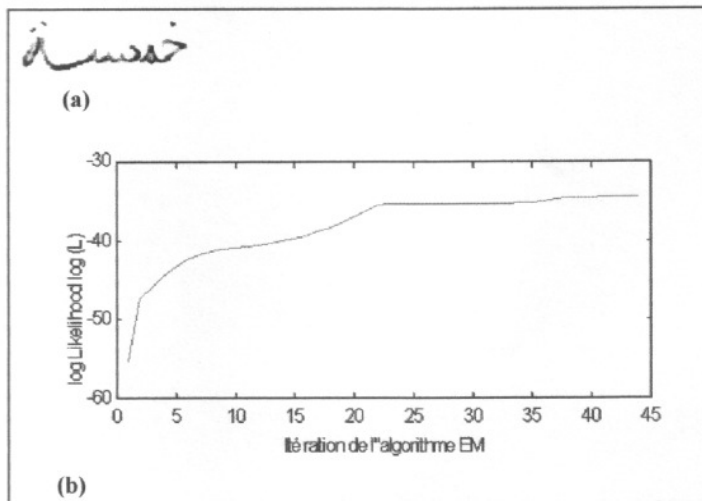
erreurs RMS similaires ou plus élevées. Une interprétation est qu'une meilleure représentation d'ambiguïté augmente la probabilité pour qu'un réseau commette une erreur quant il est normalement forcé à décider au niveau des graphèmes.

**Tableau 5.1 :**

*Résultats d'apprentissage pour les réseaux " CRISP" et " Flous "*  
*aux niveaux des graphèmes*

|                          | Taux de reconnaissance "Flou" | Erreur RMS "Floue" | Taux de reconnaissance " CRISP" | Erreur RMS " CRISP" |
|--------------------------|-------------------------------|--------------------|---------------------------------|---------------------|
| <b>Cas du groupe GR3</b> | 87.4%                         | 0.091              | 90.5%                           | 0.088               |
| <b>Cas du groupe GR4</b> | 85.8%                         | 0.098              | 91.2%                           | 0.092               |
| <b>Cas du groupe GR5</b> | 89.2%                         | 0.081              | 93.3%                           | 0.076               |

Quant au deuxième technique basée sur les HMMs, nous avons utilisé des HMMs à probabilité continues de 12 états, où chaque graphème peut appartenir à n'importe quel état suivant une certaine probabilité. Dans le cas où chaque graphème est caractérisé par des paramètres en barre, les taux de reconnaissance sur la base de données précédente sont pratiquement de l'ordre de 89%. La figure 5.8 montre l'évolution du logarithme de la vraisemblance en fonction du nombre d'itération de l'algorithme EM (Expectation-Maximisation) de Baum-Welch lors de l'apprentissage du modèle du mot cinq.



**Figure 5.8 :** (b) Evolution du logarithme de la vraisemblance lors de l'apprentissage du modèle HMM du mot (a).

## Partie II : Reconnaissance de chiffres manuscrits

### 5. II.1. Introduction

Nous avons proposé dans ce travail un prototype d'un système de lecture automatique de montants numériques de chèques arabes. Cette partie s'articulait essentiellement autour d'une partie de reconnaissance de chiffres à l'aide des réseaux de neurones MLP. Nous avons présenté un nouveau jeu de primitives basé sur le contour externe comme caractéristiques. Les résultats sont très satisfaisants au vu de la taille de la base utilisée, avec un taux de 96.20%.

### 5. II.2. Reconnaissance de chèque

Dans la littérature, très peu de travaux ont abordé des applications de lecture de chèques postaux arabe. La lecture automatique des montants de chèques nécessite le décodage de leurs expressions en chiffres et en mots.

Nous présentons dans ce travail, une partie d'un système global de reconnaissance de chèques arabes en cours de développement au laboratoire **LASA** à **Annaba** dont la tâche est la reconnaissance de montants littéraux et numériques du chèque.

### **5. II.3. Conception et réalisation**

Le but de cette tâche est de réaliser un système de reconnaissance des chiffres manuscrits composant la zone numérique du chèque à reconnaître, les chiffres se trouvent dans des cadres de taille **64 x 64** afin de faciliter la détection de chaque chiffre, d'éviter les problèmes liés à la séparation et enfin d'extraire les caractéristiques.

Pour réaliser ce système, nous avons utilisé les réseaux de neurones multicouches à une seule couche cachée. Ces réseaux présentent deux grands avantages :

- *Puissance d'approximation* : En théorie les réseaux de neurones sont capables d'approximer n'importe quelle fonction de transfert si le nombre de neurones dans la couche cachée est adéquat.
- *Ils modélisent les surfaces de séparation entre les classes.* Dans la phase d'apprentissage l'erreur diminue en prenant compte la structure interclasses des données.

Notre système se compose des modules suivants :

- ***Un module du prétraitement local.***
- ***Un module d'extraction des caractéristiques.***
- ***Un module d'apprentissage.***
- ***Un module de reconnaissance.***

Le système est omni scripteur et se présente sous la forme suivante :



## **5. II.4 Module du prétraitement local :**

### **5. II.4.1 Normalisation :**

La taille des caractères (ici les chiffres manuscrits), varie selon le scripteur et selon l'outil d'écriture ce qui peut causer l'instabilité des paramètres à extraire. La normalisation consiste à ramener les caractères à la même taille.

### **5. II.4.2 Binarisation :**

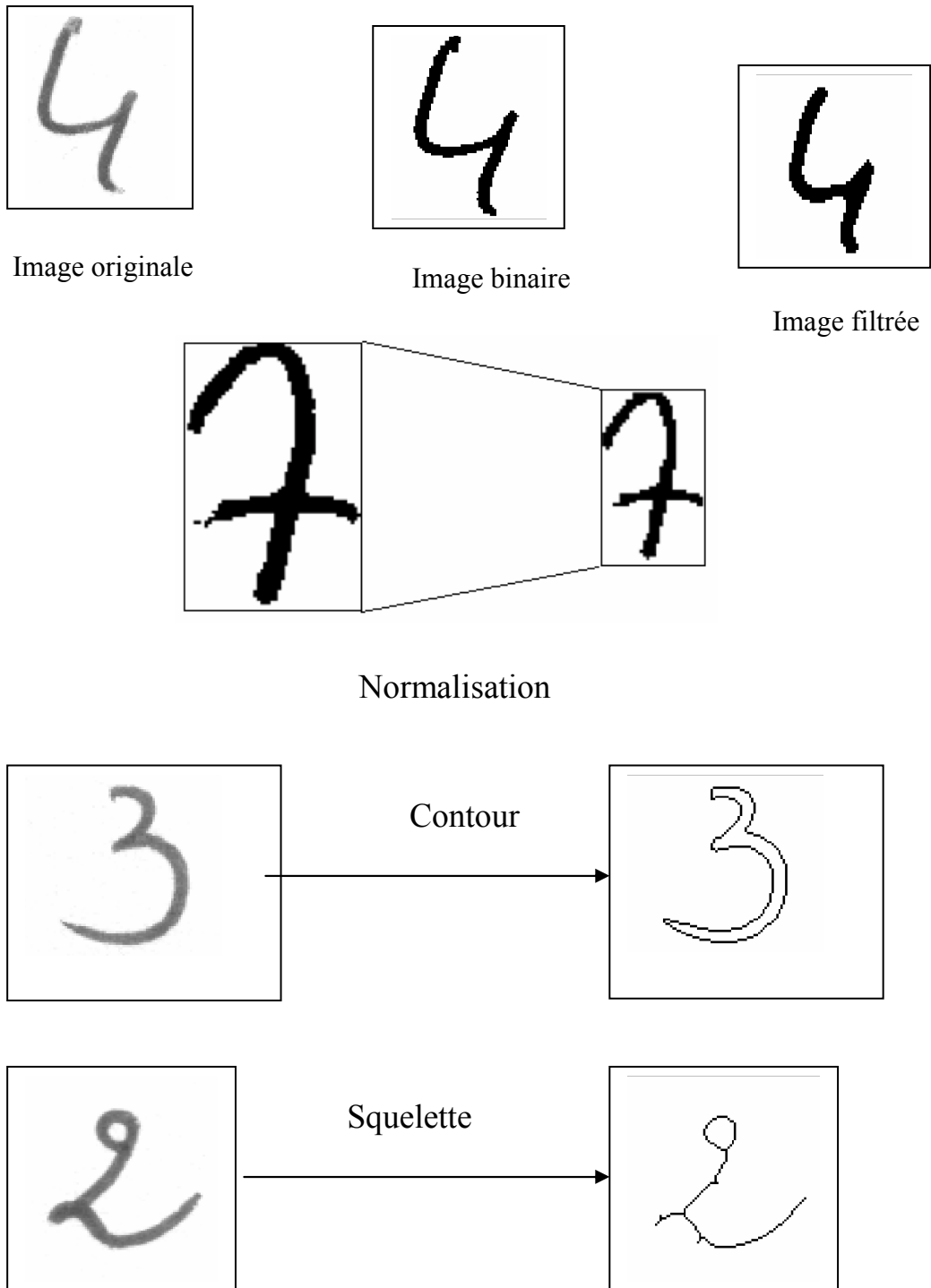
La binarisation transforme une image avec niveaux de gris en une image bimodale (blanc et noir).

### **5. II.4.3 Elimination du bruit :**

L'opération d'acquisition et de binarisation ne sont pas parfaites, elles peuvent introduire de bruit sur les contours de l'image de mot. Pour éliminer ce bruit, on applique sur celle-ci une opération de lissage, afin de simplifier les traitements ultérieurs.

### **5. II.4.4 La Squelettisation :**

La procédure de squelettisation s'effectue sur une image binaire, et a pour but de réduire l'épaisseur du tracé d'un caractère à un pixel seulement, tout en conservant la continuité de celui-ci. Le principe de cette procédure est d'effectuer une succession d'opérations d'érosion conditionnelle. Une opération d'érosion consiste à éliminer dans une image tous les pixels d'intensité non nulle qui sont adjacents aux pixels de l'arrière-plan.



**Figure 5.10 :** les différents prés traitements Binarisation, filtrage, normalisation, Contour et squelettisation

## 5. II.5 Module d'extraction des caractéristiques :

L'extraction de caractéristiques est une étape importante et nécessaire dans tout système de reconnaissance des formes, car elle permet de réduire l'information contenue dans une image ainsi que l'information redondante.

Toutes les caractéristiques sont extraites à partir du contour extérieur du chiffre (le profil droit et gauche du chiffre).

La figure suivante illustre comment calculer les profils droit et gauche.

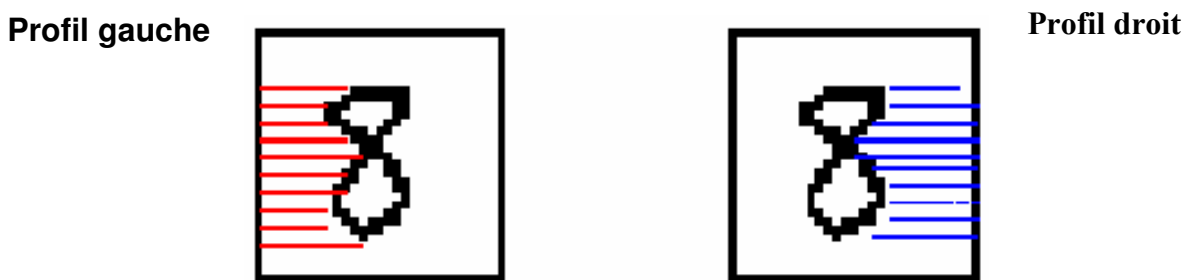


Figure.5.11 : Profil gauche et droit du chiffre 8

Nous distinguerons des caractéristiques 'Globales' et 'Locales' ; chaque chiffre et sous-classe de chiffre va être exprimé en utilisant ces caractéristiques.

### ➤ Les Caractéristiques Globales :

**a- le profil gauche :**  $\{PG(k), =1,2,\dots, NP\}$

Collection des distances entre le profil Gauche et le bord gauche du repère. (Notre repère à deux dimensions va accueillir le chiffre normalisé sur  $64*64$  pixels).

**b- le profil droit:**  $\{PD(k), =1,2,\dots, NP\}$

**c- DIFG (k) :**  $PG(k) - PG(k-1), k = 2,3,\dots, NP$

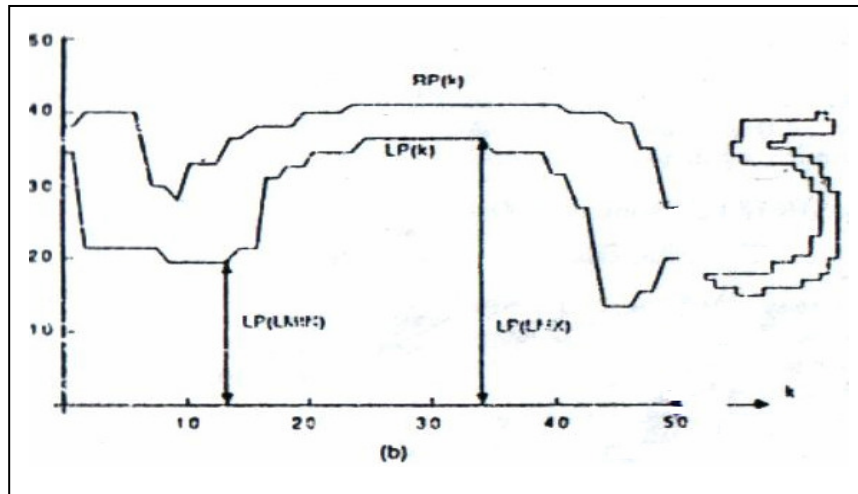
**d- DIFD (k) :**  $PD(k) - PD(k-1), k = 2,3,\dots, NP$

Ces deux primitives expriment la différence entre 2 pixels successifs du profil droit (DIFD) ou gauche (DIFG).

**e- L (k):**  $PD(k) - PG(k), k = 1, 2,\dots, NP$

Définit la largeur entre deux pixels de même abscisse.

**Remarque :** NP est le nombre de pixels (64).



**Figure 5.12 : profil haut et bas du chiffre 5**

➤ **Les caractéristiques Locales :**

A- **Le Ratio** =  $NP / \text{MAX} \{L(k)\}$ . C'est le rapport entre la hauteur du chiffre (64pixels) et sa largeur maximum.

b- **L'emplacement des maxima et minima :**

**MXG** = emplacement du  $\text{MAX} \{PG(k)\}$  sur R1,

**MXD** = emplacement du  $\text{MAX} \{PD(k)\}$  sur R1,

Où R1 est l'intervalle dans lequel MXD et MXG sont évalués.

De la même manière on définit MING et MIND.

c- **Les pics droits et gauches :**

**PICG+** =  $\text{MAX} \{DIFG(k)\}$  sur R2,

**PICG-** =  $\text{MIN} \{DIFG(k)\}$  sur R2,

Où R2 est l'intervalle où DIFD et DIFG sont spécifiés.

De la même façon les pics droits sont définis, et on aura :

$$\mathbf{PICG} = |\mathbf{PICG+}| + |\mathbf{PICG-}|$$

$$\mathbf{PICD} = |\mathbf{PICD+}| + |\mathbf{PICD-}|$$

On pourra remarquer que les pics positifs indiquent une remarquable déviation dans le contour, tandis que les pics négatifs y montrent un quasi parallélisme.

## 5. II.6 Module d'apprentissage

Les images obtenues contiennent du bruit, donc nous avons réalisé un prétraitement après l'opération de binarisation. A la fin on a obtenu dix images,

avec 100 chiffres pour chacune, c'est-à-dire 1000 chiffres en total. Nous avons sélectionné 500 chiffres pour l'apprentissage et 500 pour le test.

## 5. II.7 Conception du réseau de neurones

Le module d'apprentissage est un simulateur du réseau de neurones multicouches adapté à la classification. Il permet de concevoir le réseau et de faire l'apprentissage. Pour simuler le réseau destiné à l'apprentissage et le test on a choisi les réseaux de neurones multicouches à une couche cachée et la fonction d'activation **Logistic** pour tous les neurones du réseau.

L'algorithme de rétro\_propagation avec moment « momentum » est l'algorithme choisi pour faire l'apprentissage.

Les paramètres donnés en entrée pour ce module sont :

- La taille du vecteur des caractéristiques.
- Le nombre de neurones dans la couche cachée.
- Le nombre de neurones dans la couche de sortie.
- $\beta$  : Le momentum.
- L'erreur de sortie pour la fin d'apprentissage d'un exemple.

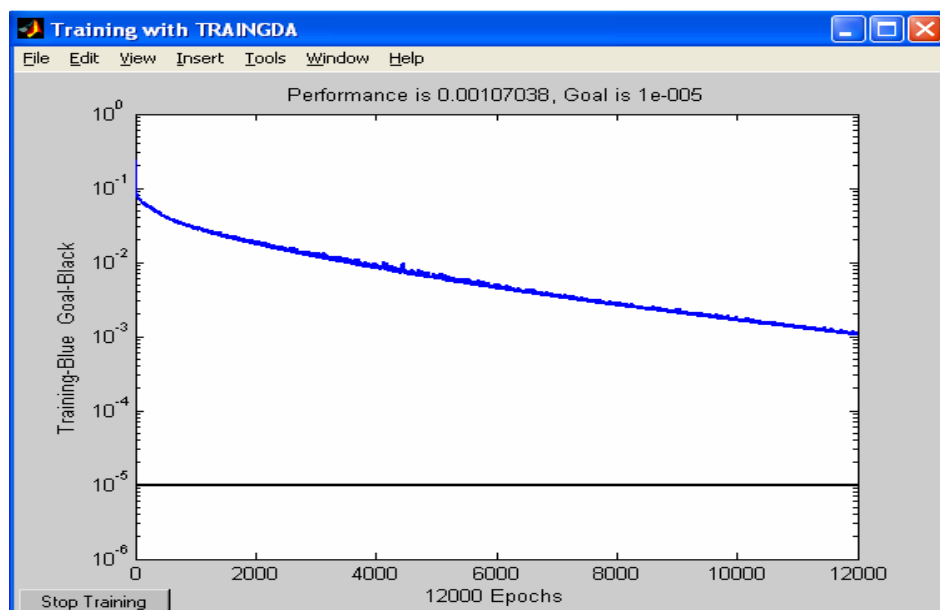


Figure 5.13 : apprentissage du réseau de neurone

## 5. II.8. Description générale des expériences

Le paramètre le plus important dans tout système de reconnaissance des formes est sans doute le taux de classification (taux de reconnaissance). Pour

déterminer ce paramètre, les images disponibles doivent être divisées en deux ensembles, l'un destiné à l'apprentissage et l'autre au test. Le taux de classification est obtenu en divisant le nombre d'images (chiffres) bien classées pendant la phase de reconnaissance, par le nombre total d'images du test.

Notre système est un classificateur à dix classes, c'est-à-dire les images des chiffres de « 0 » à « 9 », sans tenir compte des problèmes de rotation et changement d'échelle, et comme les chiffres se trouvent dans des cadres, on n'aura pas les problèmes liés à la séparation des chiffres. Dans ce qui suit, nous allons détailler toutes les étapes nécessaires à la conception de ce système et on va donner les différents résultats obtenus.

### **5. II.9. Test et résultat**

Comme il est cité dans le module d'apprentissage, nous avons commencé par concevoir les deux bases de chiffres. Dans ce qui suit nous allons citer les différents tests effectués pour fixer les différents paramètres du réseau de neurones.

#### **5. II.9.1 Tests pour fixer les paramètres du réseau :**

Pour l'obtention de meilleurs résultats de reconnaissance, les paramètres du système doivent être choisis d'une façon adéquate. La conception du réseau se fait de manière intuitive et expérimentale, la difficulté se pose généralement au moment du choix du nombre de couches intermédiaires et du nombre de neurones dans chacune de celle-ci. Pour la couche de sortie, elle contient autant de neurones que de classes à discriminer.

#### **5. II.9.2 Choix du nombre de couches cachées et le nombre de neurones dans chacune :**

Pour déterminer le nombre de couches intermédiaires et le nombre de neurones dans chaque couche, nous avons procédé de la manière suivante :

1. Fixer le nombre de neurones en sortie.
2. désigner le nombre de couches intermédiaires.
3. déterminer le nombre de neurones dans chaque couche.
4. lancer le processus d'apprentissage :

Tant que le processus diverge (n'arrive pas à stabiliser les poids), augmenter le nombre de neurones. Si celui-ci est trop élevé, augmenter le nombre de couches cachées.

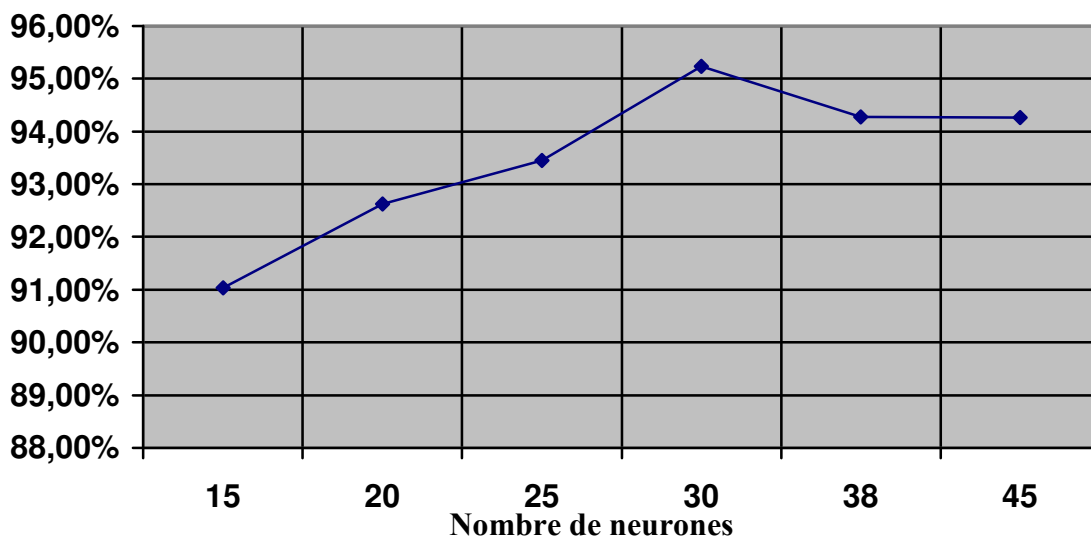
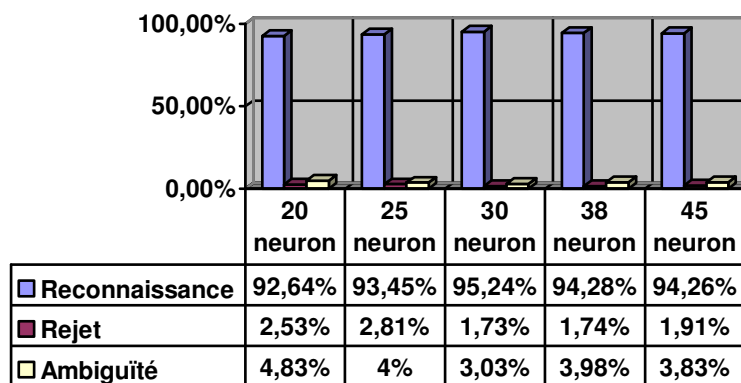
5. Fixer alors le nombre de couches et le nombre de neurones.

D'après la description précédente, nous avons commencé par :

- Une couche de sortie constituée de dix neurones.
- Une couche d'entrée reçoit directement le vecteur des caractéristiques.
- Une seule couche intermédiaire constituée de 15 neurones.
- Une base d'apprentissage de 500 chiffres (50 exemples pour chaque classe).

Avec une telle structure, le processus d'apprentissage n'a pas convergé. Nous avons alors augmenté le nombre de neurones de la couche intermédiaire, pour lesquels le processus a convergé. Reste alors à déterminer le nombre de neurones qui correspond au taux de reconnaissance le plus élevé.

Les résultats obtenus sont représentés sur les graphes suivants :



**Figure 5.14** : L'effet du nombre de neurones dans la couche cachée sur le taux de reconnaissance.

D'après le résultat du test, nous avons remarqué qu'au-delà de 30 neurones dans la couche cachée, le taux de reconnaissance sur la base de test du système a diminué, donc le nombre idéal de neurones dans la couche cachée est 30 neurones.

L'architecture du système que nous avons gardé est :

- Une seule couche cachée ayant **30** neurones.
- Une couche de sortie ayant **10** neurones.

L'apprentissage a duré **12 mn** (douze minutes) et a nécessité 90 passages à travers la base. Le temps d'apprentissage est très petit par rapport à celui écoulé dans beaucoup de travaux dans le même domaine:

### 5. II.9.3 Détermination du momentum

Pour déterminer le momentum  $\beta$ , nous avons procédé de la manière suivante :

- Varier le momentum dans l'intervalle [0.2, 0.9], jusqu'à la convergence du réseau.

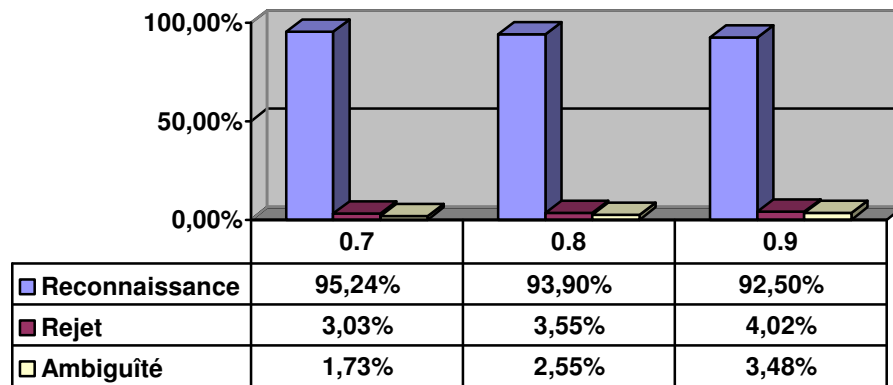
Avec cette méthode le réseau a convergé pour les valeurs du momentum suivantes :

✓  $\beta$  appartient à l'intervalle [0.7, 0.9].

Reste à déterminer les deux valeurs pour lesquelles le taux de reconnaissance soit le plus élevé, pour cela nous avons fait les quatre essais suivants (le nombre de neurones dans la couche cachée est fixé à 18 neurones) :

- $\beta = 0.7$ .
- $\beta = 0.8$ .
- $\beta = 0.9$ .

Pour le dernier essai le réseau n'a pas convergé. Les résultats obtenus pour les autres essais sont représentés sur le graphe suivant :



Après ces tests, on a fixé les valeurs :

Le momentum  $\beta=0.7$  ;

#### 5. II.9.4 Choix de l'erreur de la fin d'apprentissage d'un exemple de la base

Nous avons testé les deux valeurs d'erreur suivantes :

Erreur=0.01 et Erreur=0.005, les autres paramètres sont fixés comme suit :

$\beta=0.7$ , 25 neurones dans la couche cachée, alors nous avons obtenu les résultats présentés sur le graphe ci-dessous :

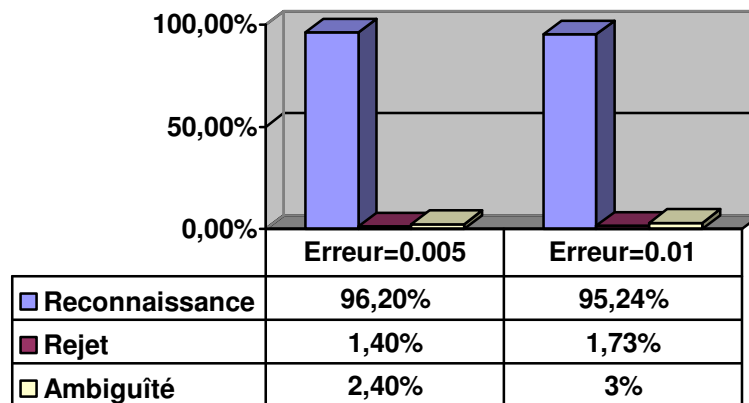


Figure 5.15 : Variation du taux de reconnaissance selon les valeurs de l'erreur.

D'après ce test on a fixé l'erreur à : **Erreur=0.005**.

A la fin de tous les tests nous avons fixé les paramètres du réseau de neurones de la manière suivante :

| Paramètre                                | Valeur |
|--|--------|
| Taille de la base d'apprentissage        | 500    |
| Nombre de couches cachées                | 1      |
| Nombre de neurones dans la couche cachée | 30     |
| Taille du vecteur des caractéristiques   | 135    |
| Taille de la couche de sortie            | 10     |
| Le momentum                              | 0.7    |
| L'erreur de la fin d'apprentissage       | 0.005  |

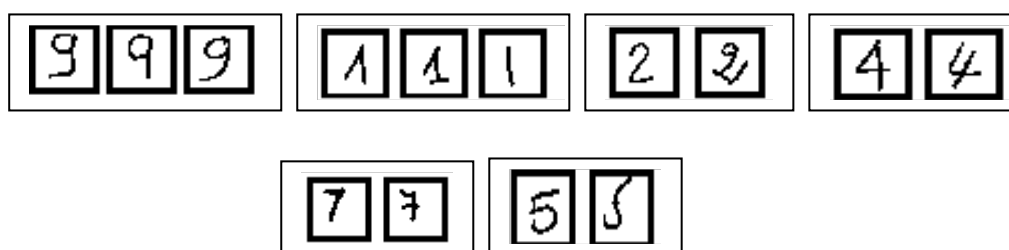
**Tableau 5.2:** Les paramètres du réseau de neurones.

## 5. II.10. Analyse des résultats

Les expériences que nous avons réalisées appellent plusieurs commentaires :

- ✚ Les caractéristiques que nous avons utilisées sont de bonne qualité, toutefois, la distorsion du contour du chiffre, l'inclinaison, la dissymétrie et le mauvais raccordement modifient les formes générales des chiffres. Cela explique les quelques erreurs de classification du système.
- ✚ Le système a donné de meilleurs résultats (96.20%, taux de reconnaissance).
- ✚ Les résultats obtenus démontrent la robustesse et la capacité de généralisation des réseaux de neurones multicouches pour la classification.

En analysant les formes des chiffres, on trouve qu'il y a parmi ces derniers qui sont de la même classe, mais elles ont des différences morphologiques considérables. La figure ci-dessous montre cette différence :



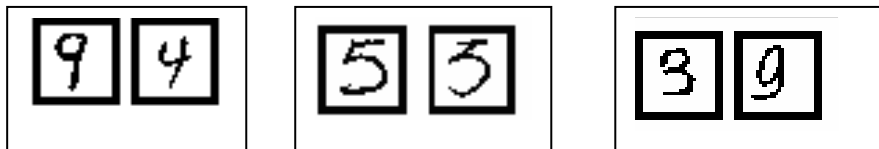
**Figure 5.16:** La différence morphologique entre les chiffres de même classe.

Le tableau suivant résume les résultats obtenus par classes :

| Classe       | Taux de reconnaissance | Taux de rejet | Taux d'ambiguïté |
|--------------|------------------------|---------------|------------------|
| Zéro         | 96 %                   | 2 %           | 2 %              |
| Un           | 100 %                  | 0 %           | 0 %              |
| Deux         | 100 %                  | 0 %           | 0 %              |
| Trois        | 96 %                   | 2 %           | 2 %              |
| Quatre       | 100 %                  | 0 %           | 0 %              |
| Cinq         | 92 %                   | 2 %           | 6 %              |
| Six          | 96 %                   | 2 %           | 2 %              |
| Sept         | 96 %                   | 0 %           | 4 %              |
| Huit         | 90 %                   | 4 %           | 6 %              |
| Neuf         | 96 %                   | 0 %           | 4 %              |
| <b>Total</b> | <b>96.2 %</b>          | <b>1.2 %</b>  | <b>2.4 %</b>     |

**Tableau 5.3:** *Les résultats du test par classe.*

Lors de notre étude, on a remarqué la ressemblance entre un certain nombre de chiffres qui ne sont pas de même classe. Dans la phase de reconnaissance et test, un nombre important de confusion ont été dues principalement à ce problème.



Le taux de confusion entre classes est représenté dans la table suivante :

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Total |
|---|---|---|---|---|---|---|---|---|---|---|-------|
| 0 | - | - | - | - | - | - | - | - | 1 | - | 1     |
| 1 | - | - | - | - | - | - | - | - | - | - | 0     |
| 2 | - | - | - | - | - | - | - | - | - | - | 0     |
| 3 | - | - | - | - | - | - | - | - | - | 1 | 1     |
| 4 | - | - | - | - | - | - | - | - | - | - | 0     |
| 5 | - | - | - | - | - | - | - | - | 2 | 1 | 3     |
| 6 | - | - | - | - | 1 | - | - | - | - | - | 1     |
| 7 | - | 2 | - | - | - | - | - | - | - | - | 2     |
| 8 | - | - | 2 | - | - | - | - | - | - | 1 | 3     |
| 9 | - | - | - | 2 | - | - | - | - | - | - | 2     |

**Tableau 5.4 :** *Les différents cas de confusion entre classes.*

L'objectif de notre projet a été de réaliser un système capable d'effectuer une classification automatique des images des chiffres, vu les taux présentés dans les tests nous pouvons dire que cet objectif est atteint.

D'après les expériences réalisées dans le système de reconnaissance des chiffres manuscrits, on peut dire que :

- Les caractéristiques métriques permettent une extraction des caractéristiques de bonne qualité.
- La qualité des résultats obtenus a confirmé sa pertinence, et en a montré l'intérêt pour un classificateur de type perceptron multicouches utilisant l'algorithme de retro propagation de gradient avec momentum.
- un nombre important de confusion ont été dues principalement à la ressemblance entre un certain nombre de chiffres qui ne sont pas de même classe.
- Le processus d'apprentissage est très rapide.

En perspectives:

- 🚧 Doter notre système d'un mécanisme d'acquisition des images avec des conditions invariantes.
- 🚧 Doter le système aussi d'un processus capable de détecter un chiffre dans une image et le localiser.
- 🚧 Trouver une solution au problème de multi-résolution.
- 🚧 Exploiter le parallélisme dans le calcul des probabilités pour améliorer le temps de reconnaissance.

## **Conclusion générale**

Nous avons présenté les différents processus d'un système de reconnaissance hors-ligne de l'écriture, dédié à une application spécifique à vocabulaire limité : la lecture automatique des montants littéraux et numériques de chèques postaux. Cependant, deux méthodes générales ont été exposées qui peuvent être appliquées à n'importe quel autre problème de reconnaissance des mots manuscrits.

Dans les deux méthodes, les images des mots sont transformées en séquences de segments. Ces segments (ou graphèmes) sont ensuite transmis à un module chargé d'estimer les probabilités d'émission. En effet, l'introduction de la notion des graphèmes est très importante puisqu'elle permet de fournir une liste de mots candidats (suivant un critère de longueur) et d'employer un processus de segmentation sans contrainte, et par conséquent d'éviter la segmentation du mot en lettres séparées.

Etant donné le but de l'application à vocabulaire limité, toute exploitation du système dépend essentiellement des taux d'erreur. Il serait donc nécessaire d'envisager plusieurs sources d'information et de les intégrer par un mécanisme de fusion des données (réseaux de neurones, logique floue, vote majoritaire, ...etc). Par exemple, la coopération avec un module de reconnaissance des chiffres peut améliorer les taux de bonne classification.

Par ailleurs, nous avons présenté les résultats préliminaires d'une autre méthode de reconnaissance des mots fondée sur une classe particulière de modèles de Markov 2D. Cette méthode a pour avantage de prendre en compte la dimension 2 des images à traiter. Contrairement aux applications des champs de Markov, elle ne sert pas à rechercher une information cachée attachée à chaque pixel mais plutôt à évaluer la probabilité suivant laquelle

un modèle engendre une image. Cette évaluation est faite par une variante de l'algorithme **EM** (Expectation Maximisation).

En perspective, nous préconisons d'améliorer le processus de segmentation du mot en graphèmes, de définir un facteur de forme optimal et de combiner plusieurs techniques. Pour la troisième méthode, une des voies d'amélioration envisageable des trois méthodes probabilistes est l'enrichissement des observations associés à chaque pixel. Si le modèle employé identifie des informations d'orientation locale du contour (dans les directions horizontale, verticale et diagonale), rien n'interdit d'adjoindre à celles-ci d'autres informations (valeur de luminance binaires, orientation locale du contour).

## BIBLIOGRAPHIE

- [**ABU 94**] Abuhaiba I.S.I., Mahmoud S.A, Green R.J., "Recognition of handwritten cursive Arabic Character", IEEE Transactions on pattern Analysis and Machine Intelligence, Vol. 16, pp.664-672, June 1994..
- [**ALB1 95**] Al Badr B., Mahmoud S.A., "Survey and bibliography of Arabic optical text recognition", Signal processing, Vol 41, pp 49-77, 1995.
- [**ALM 87**] Almuallim H, Yamaguchi S., "A method of recognition of Arabic cursive handwriting", IEEE Transactions on pattern Analysis and Machine Intelligence, Vol. 9, No. 5, pp.715-722, Sep 1987.
- [**ALY 92**] Al-Yousefi H., Udpa S.S., "Recognition of Arabic characters", IEEE Transactions on pattern Analysis and Machine Intelligence, Vol. 14, pp 853-857, Aug 1992.
- [**AME 94**] Ameer A., Romeo-Pakker K., Miled H., Cheriet M., "Approche globale pour la reconnaissance de mots manuscrits Arabes", Actes CNED'94, 3ème Colloque National sur l'Écrit et le Document, pp. 151-156, Juillet 1994.
- [**Ami 80**] A. Amin, A. Kaced : "Reconnaissance des caractères arabes manuscrits". Actes du congrès AFCET ,pp.35-44, Nancy, 1980.
- [**Ami 82**] A. Amin, G. Masini : "Machine recognition of cursive Arabic words". SPIE's Application of digital image processing IV, vol.359, pp.286-292, San Diego, CA, 1982.
- [**AMI 85**] Amin A., Masini G., Deux méthodes de reconnaissance de mots pour l'écriture arabe manuscrite », Actes RFIA'85, 5ème congrès Reconnaissance des formes et Intelligence Artificielle, Tomt 2, pp.837-848, 1985.
- [**AMI 85a**] A. Amin : "Un système pour la reconnaissance et pour la compréhension de l'arabe écrit et manuscrit". Thèse de doctorat, Université de Nancy I, 1985.
- [**AMI86**] A. Amin , G. Masini : "Machine recognition of multi-font printed arabic texts". Proc. 8<sup>th</sup> International joint conference on pattern recognition, pp.392-395, France, 1986.
- [**AMI 89**] A. Amin , J.F. Mari : "Machine recognition and correction of printed Arabic text". IEEE Transaction on systems, man, and cybernetics, vol.19, n°5, pp.1300-1304, September October 1989.
- [**AMI 92**] A. Amin ,H.B. Al-Sadoun : "A new segmentation technique of arabic text".IEEE. Proc .11<sup>th</sup> IAPR ,pp. 441-445, the Hague, the Netherlands, 1992.
- [**AMI 94**] A. Amin ,H.B. Al-sadoun : "Handprinted arabic character recognition , system". Proc. IEEE, 12<sup>th</sup> International conference on pattern recognition , pp.536-539, 1994.
- [**AMI96**] A. Amin , H.B. Al-Sadoun , S. Fischer : "Handprinted arabic character recognition system using an artificial network". Pattern recognition , vol .29, N°4,pp.663-675, 1996.
- [**AMI 97a**] A. Amin : "Off-line arabic character recognition- A survey". IEEE Proc .4<sup>th</sup> International conference on document analysis and recognition (ICDAR'97), pp.596-599, Ulm, Germany, 1997.
- [**Ami 97b**] A. Amin , W. Mansoor : "Recognition of printed arabic text using neural networks ". IEEE Proc .4<sup>th</sup> International conference on document analysis and recognition (ICDAR'97), pp.612-615, Ulm, Germany, 1997.
- [**AMI 97c**] A. Amin : "Arabic character recognition". Handbook of character recognition and document image analysis , pp. 397-420, Edition H. Bunke and P.S. Wang.1997.
- [**AMI 98**] A. Amin : "Off-line arabic character recognition : the state of the art". Pattern recognition , vol .31, n°5, pp.517-530.
- [**AMIT97**] Y. Amit, D. Geman, Shape Quantization and Recognition with Randomized Trees, Neural Computation 9 (1997), pp 1545-1588
- [**Augustin 01**] E. Augustin, Reconnaissance de mots manuscrits par systèmes hybrides réseaux de neurones et modèles de Markov cachés, Thèse de l'Université Paris V (2001)
- [**AUTRET 91**] J. Anigbogu : "Reconnaissance de textes imprimés multifontes à l'aide de modèles stochastiques et métriques ". Thèse de doctorat, Université de Nancy I, 1991.
- [**Baum 68**] L. E. Baum L.E., G. R. Sell, Growth transformation for functions on manifolds, Pac. J. Math. 27 (1968), pp 211-227
- [**Baum 72**] L. E. Baum, An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process, Inequalities 3 (1972), pp 1-8
- [**BED 95**] Bedda M., Ramdani M., Doghmane N., "Classification des caractères arabes manuscrits par un perceptron multi-couches", Conférence méditerranéenne sur l'automatique MCEA, France, 1995.

- [BEL 92]** Belaid A., Belaid Y., « Reconnaissance des formes : Méthodes et applications », InterEditions, 1992.
- [BEL 97]** Belaid A., Saon G., « Utilisation des processus markoviens en reconnaissance de l'écriture », *Traitement du Signal*, Vol. 14, No. 2, pp.161-177, 1997.
- [BEN 96]** Benyoub B., « Une application industrielle de reconnaissance d'adresses », Actes CNED'96, 4ème Colloque National sur l'écrit et le document , pp. 93-100, Nantes, France, Juillet 1996.
- [BEN 97]** N.Ben Amara : "Application des PHMMs pour la reconnaissance de l'écriture arabe imprimée". 1<sup>ères</sup> Journées Scientifiques et techniques (JST FRANCIL) pp.389-392, Avignon, France, 1997.
- [BEN 98]** BENSRAHAIR A., MOUSSET S., MICHE P., LEE SG. : "Real-time stereo vision system for axial motion detection"; 9th Conference on Sensor Technology, Kyungpook, Korea, november 6-7 1998; pp 50-58 (1998) .
- [Bengio92]** Y. Bengio, R. De Mori, G. Flammia, R. Kompe, Global Optimization of a Neural Network- Hidden Markov Model Hybrid, *IEEE Transactions on Neural Networks* 3(2) (1992), pp 252-259
- [Bengio96]** Y. Bengio, P. Frasconi, Input/Output HMMs for sequence processing, *IEEE Transactions on Neural Network*, 7(5) 1231 (1996), pp 1249-
- [Bishop95]** C. M. Bishop, *Neural networks for pattern recognition*, Oxford University Press (1995)
- [Bottou91]** L. Bottou, Une approche théorique de l'apprentissage connexionniste, Application à la reconnaissance de la parole, Thèse de l'Université de Paris Sud, Centre d'Orsay (1991)
- [BRET 88]** G. BRET, K.M. Kulkarni: "A high accuracy algorithm for recognition of handwritten numerals". *pattern recognition* , vol.21, pp.287-291, 1988.
- [Bokser 92]** H. Boulard, C. Wellekens: " Links between hidden Markov models and multiplayer perceptrons". *IEEE Transactions on pattern analysis and machines intelligence*, vol.12, pp-1197-1178.
- [Bunke95]** H. Bunke, M. Roth, E. G. Schukat-Talamazzini, O-line cursive handwriting recognition using hidden Markov models, *Pattern Recognition* 28 (1995), pp 1399-1413
- [Burges98]** C. J. C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery* 2(2) (1998), pp 955-974
- [Broyden67]** C.G. Broyden, Quasi-Newton methods and their application to function minimization, *Math. Comput* 21 (1967), pp 368-
- [Cas96]** R. Caesy, E.Lecolinet : "Strategies in character segmentation : a survey". *IEEE Proc.3<sup>rd</sup> International conference on document analysis and recognition (ICDAR' 95)*, p.1028-1033, Monreal, Canada, 1996.
- [Cybenko89]** G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of Controls, Signals, and Systems* 2 (1989), pp 303-314
- [Chen94]** Mou-Yen Chen, Amlan Kundu, Jian Zhou, O-line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(5) (1994), pp 481-496
- [Dempster77]** A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum-Likelihood from incomplete data via the EM algorithm, *Journal of Royal Statistical Society B* 39 (1977), pp 1-38
- [Davidon59]** C. W. Davidon, Variable metric method for minimization, A.E.C. Research and Development Report, ANL-5990 (1959)
- [Driancourt96]** X. Driancourt, Optimisation par descente de gradient stochastique de systèmes modulaires combinant réseaux de neurones et programmation dynamique, Application à la reconnaissance de la parole, Thèse de l'Université de Paris Sud, Centre d'Orsay (1996)
- [DIVIJER 82]** T.M.Cover,: "An algorithm for maximizing expected log investment return ". *IEEE Transactions on inform. Theory* IT-30, pp.369-373, 1982.
- [DUN 92]** S.S. El-Dabi, R. Ramsis, A.Kamel : « Arabic Character recognition system : a statistical approach for recognizing cursive typewritten text". *Pattern recognition*, vol.23, n°5, pp.485-495, 1992.

- [ELD 90]** El-Dabi S. S., Ramsis R., Kamel A., "Arabic character recognition system: a statistical approach for recognizing cursive typewritten text", *Pattern Recognition*, Vol. 23, No 5, pp 485-495, 1990.
- [ELK 90]** El-Khaly F., Sid Ahmed M. A., "Machine recognition of optically captured machine printed Arabic text", *Pattern Recognition*, Vol. 23, No. 11, pp.1207-1214, 1990.
- [ELS 1 88]** El-Sheikh T. S., Guindi R. M., "Automatic recognition of isolated Arabic cursive script", *Signal processing*, Vol. 14, pp 177-184, 1988.
- [ELS 2 88]** El-Sheikh T. S., Guindi R. M., "computer recognition of Arabic cursive script", *Pattern Recognition*, Vol. 21 No 4, pp 293-302, 1988.
- [ELW 89]** El-Wakil M., Shoukry R., "On-line recognition of handwritten Arabic characters", *Pattern Recognition*, Vol. 22, No. 2, pp. 97-105, 1989.
- [Eickeler98]** S. Eickeler, S. Müller, G. Rigoll, Improved Face Recognition Using Pseudo-2D Hidden Markov Models, Workshop on Advances in Facial Image Analysis and Recognition Technology in conjunction with 5th European Conference on Computer Vision (ECCV'98), Freiburg, Germany (1998).
- [Fletcher93]** R. Fletcher, An overview of Unconstrained Optimization, Numerical Analysis Report NA/149 (1993).
- [FOR73]** G.D. Forney : "The Viterbi algorithm ". *Proc. IEEE*, vol. 61n°3, pp.268-278, 1973.
- [FUG 92]** .Fujisawa, Y. Nakano, K. Kurino : "Segmentation methods for character recognition : from segmentation to document structure analysis". *Proc.IEEE*, vol.80,n°7, 1992.
- [Gader 92]** F.Ghorbel : "Stability of invariant Fourier descriptors and its inference in the shape classification". *Proc.IEEE 11<sup>th</sup> International conference on pattern recognition (ICPR'93)*, 1992.
- [GRA 72]** G.Gaillat : "Méthodes statistiques de reconnaissance de formes « . Edité par le Centre d'édition et de documentation de l'Ecole Nationale Supérieure de Techniques Avancées, 1972.
- [GOR 92]** Goraine H., Al Emami S., "Off-line arabic recognition", *Computer*, pp. 71-74, July 1992.
- [GOW 90]** Gowely K., El Dessouki O., "Multi-phase recognition of multifont photscript Arabic text", 10<sup>th</sup> International Conference on Pattern Recognition, Vol. 1, pp.700-702, June 1990.
- [GOV 90]** G.Gaillat : "Méthodes statistiques de reconnaissance de formes « . Edité par le Centre d'édition et de documentation de l'Ecole Nationale Supérieure de Techniques Avancées, 1990.
- [Heutte 93]** J.P. Hugues, P. Guttorp, W.F. Krajewski, K.P.Georgegakakos:" Incorporating spatial dependence and atmosphere data in a model of precipitation: hydrological and meteorological aspect of rainfall measurement and predictability ". *Journal of applied meteorology* : 1988, vol 33,n°12, pp.1503-1515, 1993.
- [KELLER 85]** B.M.Kim, S.B.Park: " A fast k nearest neighbor finding algorithm based on the ordered partition " . *IEEE transactions on pattern analysis and machine intelligence*, PAMI-6(6), pp.761-766, 1986.
- [Knerr00]** S. Knerr, P. Lallican, C. Viard-Gaudin, From O-Line to On-Line handwriting Recognition, International Workshop on Frontiers in Handwriting Recognition, IWFHR'2000, Amsterdam, The Netherlands (2000), pp 303-312
- [Knerr01]** S. Knerr, Y. Tay, P. Lallican, M. Khalid, C. Viard-Gaudin, An o\_line cursive handwritten word recognition system, In *Proceedings of IEEE Region 10 Conference (2001)*
- [Koerich02b]** A. L. Koerich, R. Sabourin, Y. Leydier, C. Y. Suen, A Hybrid Large Vocabulary Handwritten Word Recognition System using Neural Networks with Hidden Markov Models, 8th International Workshop on Frontiers of Handwriting Recognition (IWFHR'8) (2002), pp 99-104
- [Khorsheed03]** M. S. Khorsheed, Recognising handwritten Arabic manuscripts using a single hidden Markov model, *Pattern Recognition Letters* 24 (2003), pp 2235-2242
- [Kuo94]** S. Kuo, E. Agazzi, Keyword Spotting in Poorly Printed Documents Using Pseudo 2-D Hidden Markov Models, *IEEE Transactions on Pami* 16(8) (1994), pp 842-848.
- [Lecolinet96]** E. Lecolinet, A Survey of Methods and Strategies in Character Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 18(7) (1996), pp 690-706

- [LeCun85]** Y. Le Cun, Une procédure d'apprentissage pour réseaux à seuil asymétrique, *Cognita* (1985), pp 599-604
- [LeCun98]** Y. Lecun, L. Bottou, Y. Bengio, P. Haner, Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE* 86 (11) (1998), pp 2278-2324
- [Levinson83]** S. E. Levinson, L. R. Rabiner, N. M. Sondhi, An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition, *Bell System Technical Journal* 52 (4) (1983), pp 257-285.
- [Lorette 92]** G.Lorette, Y.Lecourtier : "Reconnaissance et interpretation de textes manuscrits hors-ligne : un problème d'analyse de scène ? ». *Bigre n°80, Actes du 2<sup>ème</sup> Colloque national sur l'écrit et le document (CNED'92)*, pp.109-135, Nancy, 1992.
- [LIA 94]** L.C. Liu, D. Chiou, H.C. Wang: "A speech recognition method based on feature distributions". *Pattern recognition*, vol.24,n°8, pp.717-722, 1994
- [LU 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.
- [MAH 94]** Mahmoudi S., "Arabic Character recognition using Fourier descriptors and character contour encoding", *Pattern Recognition*, Vol . 27, No. 6, pp. 815-824, 1994.
- [MIL 97]** Miled H., "Stratégies de résolution en reconnaissance de l'écriture semi-cursive: application aux mots manuscrits arabes", thèse de doctorat, université de Rouen, 1997.
- [Moré77]** J. J. Moré, The Levenberg-Marquardt algorithm : Implementation and theory, *Proceedings of the 1977 Dundee Conference on Numerical Analysis*, G. A.Watson, ed., *Lecture Notes in Mathematics*, vol. 630, Springer-Verlag, Berlin (1977), pp 105-116;
- [MOR 92]** S. Mori, C.Y.Suen, K.Yamamoto, : « Historical review of OCR reseach and developement ». *Proc. IEEE*, Vol.80, n°7, pp.1029-1057, 1992.
- [OLI 96]** C.Olivier, H. Miled, K. Remeo, Y.Lecourtier : "Segmentation and coding of arabic handwritten words". *IEEE Proc. 13<sup>th</sup> International conference on pattern recognition (ICPR'96)*, pp.264-268, Vienne, Autriche, 1996.
- [OLI 99]** Olivier D., « 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 92]** Paquet T., « Segmentation et cassification de mots en reconnaissance optique de textes manuscrits » Thèse de Doctorat, Université de Rouen 1992.
- [PAV 96]** Pavlidis T., « Algorithms for Graphic and image processing », Rockville, MD: Computer science press, 1996.
- [PETIER 91]** B.Parhami, M.Taraghi : »Automatic recognition of prined farsi texts". *Pattern recognition*, vol.14, n°1, pp.1- 6, 1991.
- [Rabiner86]** L. R. Rabiner, B. H. Juang, An introduction to hidden Markov models, *IEEE ASSP Magazine* January (1986), pp 4-15.
- [Rabiner 89]** L.Rabiner, : "A tutorial on hidden Markov models and selected applications in speech recognition". *Proc.IEEE*, vol.77, 1989.
- [Rumelhart86]** D. E. Rumelhart, G. E. Hinton, R. J. Williams, *Learning internal representations by error propagation*, *Parallel distributed processing : explorations in the microstructures of cohnnyonn* MIT Press, Cambridge (1986).
- [Saon97]** G. Saon, A. Belaid, High Performance Unconstrained Word Recognition System Combining HMMs and Markov Random Field, *International Journal on Pattern Recognition and Arti\_cial Intelligence (IJPRAI) Special Issue on Automatic Bankcheck Processing*, S. Impedovo Ed. vol. 11, n. 5 (1997), pp 771-788.
- [Saporta90]** Gilbert Saporta, *Probabilités, analyse des données et statistique*, Editions Technip (1990).
- [Sayre73]** K. Sayre, Machine recognition of handwritten words: a project report, *Pattern Recognition* 5(3) (1973), pp 213-228.
- [SEL 98]** Sellami M., Souici L., « Un système hybride de reconnaissance de l'arabe manuscrit renforcé par un analyseur morpho-lexical », *Journées d'Analyse de Données Textuelles, JADT'98*, Nice, France, Février 1998.
- [Senior94]** A. Senior, O\_-line Cursive Handwriting Recognition using Recurrent Neural Network, *PHD Thesis in Trinity Hall, Canmbridge*, England (1994).

**[Senior98]** A. Senior, An o\_line cursive handwriting recognition system, IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3) (1998), pp 309-321.

**[SOU 1 96]** Souici L., « Système connexionniste pour la reconnaissance des caractères arabes manuscrits », Mémoire de Magister, Département d'informatique, Université Badji-Mokhtar, Annaba 1996.

**[SOU 2 96]** Souici L., Farah N., Sellami M., « Contribution to the recognition of hand-written arabic text by a neural network in cooperation with an expert system », current trends in computer science and information systems, CTCSIS'96, pp. 13-23, Philadelphia University, Jordan, July 1996.

**[SOU 97]** Souici L., Zemirli Z., Sellami M., « Système connexionniste pour la reconnaissance de l'arabe manuscrits », Journée Scientifiques et Techniques : JST'97-Francil, Avignon, France, Avril 1997.

**[SOU 98]** Souici L., Sari T., Zemirli Z., Sellami M., « Prototypé de reconnaissance de caractères arabes manuscrits a base de sous réseaux neuronaux », Revue Synthèse (Publication de l'université d'Annaba, Algérie), No. 3, pp. 5-11, Juin 1998.

**[TOL 90]** Tolba M.F., Shaddad E., "On the automatic reading of printed Arabic characters", IEEE International Conference on Systems, Man and Cybernetics Conference Proceedings, pp. 496-498, 1990.

**[Sue 93]** C.Y. Suen : "Réflexions sur la reconnaissance d'écriture cursive". Proc.1<sup>er</sup> Colloque International francophone sur l'écrit et le document (CIFED'93), pp.1-8, Québec, Canada, 1993.

**[Simon92]** J. C. Simon, O. Baret, Cursive words recognition, From Pixel to Features III : Frontiers in Handwriting Recognition, S. Impedovo and J.C. Simon (eds.) Elsevier Science Publishers B.V. (1992), pp 241-260

**[Vapnik79]** V. N. Vapnik, Estimation of dependances based on empirical data, (en russe), Nauka, Moscow, 1979, traduction anglaise : Springer-Verlag, New-York (1982) (1979)

**[Vinciarelli02]** A. Vinciarelli, A survey on o\_line Cursive Word Recognition, Pattern Recognition 35 (2002), pp 1433-1446

**[TAP 90]** C.C. Tappert, C.Y. Suen, T. Wakahara : "The state of the art in on-line handwriting recognition ". IEEE Transactions on pattern analysis and machine intelligence, vol.12, n°8, pp.787-808, 1990.

**[TOU 04]** Touj S., Essoukri Ben Amara N., « Reconnaissance de l'écriture arabe imprimée par transformée de hough généralisée », CIFED'2004, Colloque International Francophone sur l'Écrit et le document, pp. 267-272 La Rochelle, France, Juin 2004.

**[TRE 95]** C.C. Tappert, C.Y. Suen, T. Wakahara : "The state of the art in on-line handwriting recognition ". IEEE Transactions on pattern analysis and machine intelligence, vol.12, n°8, pp.787-808, 1995.

**[TRIER 95]** J.P. Trincklin : "Conception d'un système d'analyse de documents, étude et réalisation d'un module d'extraction de la structure physique de documents à support visuel ». Thèse Université de France- Compté, Besançon, 1995.

**[Takahachie 91]** F. takahachie R. Plamondo : "On-line recognition of handprinted characters : "survey and beta tests". Pattern recognition, vol.23, n°9, pp.1037-1044, 1990.

**[Tri 96]** J.P. Trincklin : "Conception d'un système d'analyse de documents, étude et réalisation d'un module d'extraction de la structure physique de documents à support visuel ». Thèse Université de France- Compté, Besançon, 1996.

**[XU 92]** Xu L., Krzyzak A., Suen C. Y., « Methods of combining multiple classifiers and their application to handwriting recognition », IEEE Transactions on systems, man and cybernetics, Vol 22, No.3, pp.418-435, May/June 1992.

**[ZAH 90]** A.Zahour, B. Taconet, A. Faure : " Méthode structurelle de reconnaissance de l'écriture arabe manuscrite". Bigre n°68, Actes du 1<sup>er</sup> Colloque national sur l'écrit et le document (CNED'90), pp.148-159, Nancy, 1990.

**[ZAH 98]** A. Zahour, A. Djematene, S. Kebairi., "Contribution à la reconnaissance de l'écriture manuscrite arabe", Actes CIFED'98, 1 ère Colloque International Francophone sur l'Écrit et le Document, pp. 218-227, Québec, Canada, Mai 1998.

**[Zahn 72]** K.H. Zahn, F. Fallside : »Dynamic programming in the recognition of connected handwritten script ». IEEE 1972.

**[ZERMI 00]** N.Zermi, M.Ramdani, M.Bedda, N.Doghmane « Un modèle neuro-markovien pour la reconnaissance des mots manuscrits arabes ». Journées de mathématiques appliquée (JMA2000), Blida 13 et 14 Novembre 2000, Blida, Alger.

**UNIVERSITE BADJI-MOKHTAR ANNABA  
FACULTES DES SCIENCES DE L'INGENIEUR  
DEPARTEMENT D'ELECTRONIQUE**

**Pr : Mouldi BEDDA**

**PROPOSITION DE JURY  
DE SOUTENANCE**

Vu que le travail de recherche, dans le cadre de la préparation de la thèse de doctorat, de Mme ZERMI NARIMA dont l'intitulé est :

***Reconnaissance de mots manuscrits Arabe par les modèles de Markov cachée  
et les réseaux de neuronne***

Est achevé. Je propose pour la soutenance le jury suivant:

|              |    |              |          |
|--------------|----|--------------|----------|
| Président    | Pr | H Tebbikh    | U GUELMA |
| Rapporteur   | Pr | M Bedda      | U ANNABA |
| Corapporteur | Pr | N Doghmane   | U ANNABA |
| Examineurs   | Pr | Dj Chikouche | U SETIF  |
|              | Mc | L Zouaoui    | U SETIF  |
| Invité       |    | M Ramdani    | U ANNABA |

الأستاذ الدكتور  
المولدي بده



**UNIVERSITE BADJI-MOKHTAR ANNABA  
FACULTES DES SCIENCES DE L'INGENIEUR  
DEPARTEMENT D'ELECTRONIQUE**

**Pr : Mouldi BEDDA**

**Rapport de thèse de doctorat**

Rapport établi par **Pr Mouldi BEDDA**  
sur la thèse de Mme **ZERMI NARIMA**

Depuis quelques années, Mme **ZERMI** s'intéresse à l'étude des systèmes intelligents et plus particulièrement aux algorithmes intelligents permettant l'analyse des documents manuscrits en arabe. Ce travail a été initié lors de la préparation de son magister où elle s'est intéressée à la reconnaissance de l'écriture manuscrite arabe par l'approche hybride, elle continue sur le même thème mais avec plus d'approfondissement.

Elle commence par un exposé de l'état de l'art sur la reconnaissance de l'écriture manuscrite hors ligne, puis les outils utilisés pour l'analyse et la reconnaissance qui consistent à réaliser une extraction des primitives en faisant une segmentation logique qui débute par la segmentation du texte en ligne puis la ligne en mots et la phase finale la segmentation du mot en lettres ou en graphèmes selon l'approche adoptée. Les outils utilisés pour la classification et la reconnaissance sont présentés d'une manière succincte.

Un chapitre a été consacré à la recherche développée dans le domaine de la reconnaissance de l'écriture arabe en faisant apparaître ses particularités telles que: écriture de droite à gauche, ligature, cursivité et points et leurs dispositions.

Dans le dernier Chapitre Mme Zermi a présenté d'une manière explicite la méthodologie adoptée pour l'extraction des paramètres de l'écriture et comme l'analyse est réalisée selon l'approche globale où on s'intéresse au mot complet, à partir de cette analyse un regroupement des graphèmes est réalisé.

Les outils utilisés pour la reconnaissance sont: la chaîne de Markov cachée, le réseau de neurone et les fonctions d'appartenance. Les tests des algorithmes développés par la candidate sur une base de données conçue au laboratoire d'automatique et signaux de l'université Badji Mokhtar montrent bien la robustesse des algorithmes et l'efficacité de la démarche adoptée.

En conclusion, l'ensemble des travaux réalisés par Mme Zermi ont tout le mérite d'être soutenus en thèse de doctorat.

الأستاذ الدكتور  
المولدي بده



## TRAVAUX DE L'AUTEUR

### PUBLICATIONS :

1 – **N. Zermi**, M. Ramdani, M.Bedda, « Arabic Handwriting Word Recognition Based on a Hybrid HMM/ANN Approach », International Journal of Soft Computing , IJSC, vol 1, N°1, pp. 05-10, Janvier 2007.

### COMMUNICATIONS :

1-**N. Zermi**, M. Ramdani, M.Bedda, « Un modèle neuro-markovien pour la reconnaissance de l'écriture arabe », Journée des Mathématiques Appliquées, JMA2000, Université de Blida, Alger, le 13-14 Novembre 2000.

2- **N. Zermi**, M. Ramdani, M.Bedda, « reconnaissance des montants littéraux arabes », Conférence Maghrébine en Génie Electrique, CMGE'01, Université de Constantine, Alger , le 05-06 Novembre 2001.

3- **N. Zermi**, M. Ramdani, M.Bedda, « Approche Neuro-Markovienne pour la Reconnaissance des mots manuscrits arabes » Séminaire Nationale sur l'automatique et les signaux, SNAS'02, Université d'Annaba, Alger, le 27-28 Octobre 2002.

4- **N. Zermi**, M. Ramdani, M.Bedda,, « Neuro-Markovian hybrid system for handwritten Arabic word recognition », 10th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2003, December 14-17, 2003- Sharjah-United Arab Emirates.

5- **N. Zermi**, M.Bedda, « Reconnaissance de mots manuscrits arabe, application : lecture des adresses postales », Colloque National sur le traitement du signal et ses applications, CNTSA'05, Université de Guelma, Alger, 19-20 Septembre 2005.

6- **N. Zermi**, M. Ramdani, M.Bedda, «grapheme neural networks for Arabic Handwritten Recognitionl », Conférence Intenational sur le Contrôle, la modélisation et le Diagnostic, ICCMD'06, Université d'Annaba, Alger, 22-24 Mai 2006.

7- **N. Zermi**, M. Ramdani, M.Bedda, « Arabic Handwritten Recognition with grapheme neural networks”, 4<sup>th</sup> International Multi Conference on Computer Science and Information Technology,CSIT'2006, Applied Science University, Amman, Jordan from the 5<sup>th</sup> until the 7<sup>th</sup> of April 2006.