



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



UNIVERSTE MOULOUD MAMMARI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire

*En vue de l'obtention du diplôme Master académique
en Informatique
Option : Système Informatique*

Thème

*Reconnaissance automatique des noms
arabes manuscrits anciens*

Proposé par :

M^r: Soualah Md Ou Rabah

Réalisé par :

M^{elle} : Ben belkacem Farida

M^{elle} : Amalou Kenza

Promotion: 2010/2011

Remerciements

Tout d'abord, nous remercierons notre promoteur M^r M. SOUALAH pour nous avoir proposé ce thème, nous lui sont très reconnaissantes pour ses remarques et conseils, ses contributions et orientations ont toujours été d'un grand apport au fur et à mesure que nous avançons dans notre travail. On espère d'avoir être à la hauteur de la confiance qu'il nous a toujours prêtée. C'était un honneur pour nous de travailler avec lui.

Nous aimerions également exprimer notre gratitude aux membres du jury, devant qui nous avons l'honneur de présenter notre travail et qui ont pris la peine de le lire avec ce soin ce mémoire pour juger son contenu.

C'est avec un immense plaisir que je dédie ce

Travail,

A ma chère mère qui est la perle de mon cœur.

*A mon cher père pour tous les sacrifices qu'il a
consenti pour moi.*

A mes frères Yahia et Nabile

A toutes mes sœurs

A tous mes amis et amies, qu'ils trouvent ici

L'expression de mon estime.

A ma famille et à tous ceux que j'aime et qui ont

Souhaité ma réussite.

Farida

Je dédicace ce modeste travail :

A mes parents, ce travail vous doit beaucoup, qu'il soit pour vous le témoignage de mon infinie reconnaissance pour ces années de compréhension et d'efforts communs.

A ma très chère grand-mère

A mes frères Nouredine et M'hand,

A ma sœur Ouiza, sa fille Lina et son fils Toufik

A mon fiancé Lyes qui m'a poussée et motivée dans mes études.

A toute ma famille et mes amies.

Ce mémoire représente l'aboutissement du soutien et des encouragements qu'ils m'ont prodigués tout au long de ma scolarité.

Qu'ils en soient remerciés par cette modeste dédicace.

Kenza.

Sommaire

Introduction générale.....	1
Chapitre I : état de l'art	
I. Introduction	3
II. Définition de système de reconnaissance de caractère (SRC)	3
III. Généralité sur les SRC.....	3
III.1. La reconnaissance En-Ligne (on-line)	4
III.2. La reconnaissance hors-ligne (off-line).....	5
III.3. Un système Mono-scripteur, Multi-scripteur et Omni-scripteur	5
III.4. Approches de reconnaissance	5
III.4.1. L'approche globale	5
III.4.2. L'approche analytique	6
IV. Architecture générale de SRC manuscrit.....	6
IV.1. Phase d'acquisition.....	8
IV.2. Prétraitements	9
IV.3. Phase d'analyse.....	11
IV.3.1. La segmentation.	11
IV.3.1.1. La segmentation explicite.....	11
IV.3.1.2. La segmentation implicite	11
IV.3.1.3. La méthode sans segmentation	12
IV.3.2. L'extraction des primitives	12
IV.3.2.1. Caractéristiques structurelles.....	12
IV.3.2.2. Les caractéristiques statistiques	13
IV.3.2.3. Les transformations globales.....	13
IV.3.2.4. Superposition des modèles et corrélation.....	14
IV.4. La classification.....	14
IV.4.1. L'apprentissage.....	14
IV.4.2. Reconnaissance et décision.....	15
IV.4.2.1. Approche statistique.....	15
IV.4.2.2. Approche structurelle	16
IV.4.2.3. Approche stochastique	16
IV.5. Post-traitement.....	17
V. Méthode de reconnaissance de caractères	17

Sommaire

V.1. Le modèle de Markov caché (MMC).....	18
V.1.1. Définition de MMC	18
V.1.2. Types de modèle de Markov caché	20
V.2. Réseaux Bayésiens	21
V.2.1. Définition	21
V.2.2. Apprentissage	22
V.2.3. Classification avec les réseaux bayésiens	22
V.3. Réseau de neurone	23
V.3.1. Type de réseaux de neurones	25
V.4. Machines à Vecteurs de Support (SVM).....	26
VI. Les systèmes existants de la reconnaissance d'écriture arabes	28
VI.1. Caractéristiques générales de l'écriture arabe	28
VI.2. Les manuscrits arabes.....	30
VI.2.1. Définition de manuscrit	30
VI.2.2. Caractéristiques des manuscrits arabes.....	31
VI.3. Quelques systèmes existants de reconnaissance d'écriture arabe	32
VI I. Conclusion.....	38

Chapitre II : Conception

I. Introduction.....	39
II. Mise en œuvre de la méthode de Markov caché (HMM)	40
III. Architecture de système proposé	41
III.1. Description générale.....	41
III.2. Description détaillée de notre système.....	44
III.2.1. Sous système d'apprentissage.....	44
III.2.1.1. Prétraitement.....	44
1. Niveau de gris	44
2. Binarisation.....	45
3. L'encadrement	47
4. Normalisation.....	51
III.2.1.2. L'extraction des primitives.....	53
III.2.1.3. Apprentissage.....	54
III.2.1.3.1. Algorithme de Baum-Welch	55

Sommaire

III.2.1.4. Base de données du notre système.....	56
III.2.2. Le sous-système de reconnaissance.....	57
III.2.2.1. Prétraitement.....	58
1. La correction de l'inclinaison des lignes.....	59
2. La correction de l'inclinaison des caractères.....	59
III.2.2.2. La segmentation.....	59
III.2.2.2.1. La segmentation en lignes.....	60
III.2.2.2.2. La segmentation en caractères.....	60
III.2.2.3. Normalisation et extraction des caractéristiques.....	68
III.2.2.4. La reconnaissance par Viterbi.....	69
III.2.2.5. Regroupement des lettres et la reconnaissance du nom.....	70
IV. Conclusion.....	71

Chapitre III : Réalisation

I. Introduction.....	72
II. Les outils de développement.....	72
II.1. Système d'exploitations.....	72
II.2. Langage de programmation.....	72
II.3. Le langage SQL (Structured Query Language).....	73
II.4. IDE Eclipse.....	73
II.5. Le SGBD MYSQL.....	74
III. Présentation de notre système.....	74
III.1. Les interfaces de notre système.....	75
III.1.1. Interface d'accueil.....	75
III.1.2. Interface de prétraitement.....	76
III.1.3. Interface d'apprentissage.....	76
III.1.4. Interface reconnaissance de la lettre.....	77
III.1.5. Interface de segmentation en lignes.....	78
III.1.6. Interface de reconnaissance de nom.....	79
III.2. Le fonctionnement de système.....	80
III.2.1. Sous système d'apprentissage.....	80
III.2.2. Sous système de reconnaissance.....	82
IV. Résultats Expérimentaux.....	86

Sommaire

IV.1. Tache de la reconnaissance des lettres arabes manuscrites.....	86
IV.2. Tache de la segmentation.....	87
IV.3. Tache de reconnaissance des noms.....	87
V. Conclusion.....	88
Conclusion générale.....	89
Glossaire	
Bibliographie	

Table de figure

Chapitre I : Etat de l'art

Figure 1 : Systèmes d'acquisition en ligne	4
Figure 2: Schéma général d'un système de reconnaissance de caractères.....	7
Figure 3: Différents niveaux de résolution	8
Figure 4: effets des opérations de prétraitement	10
Figure 5: Segmentation implicite et explicite	11
Figure 6 : Illustrations de primitive Zoning.....	13
Figure 7: Définition des paramètres d'un Modèle de Markov Caché.....	19
Figure 8 : Différente représentations markoviennes du mot	20
Figure 9: Neurone artificiel	24
Figure 10 : Tableau des fonctions de transfert.....	24
Figure 11 : Réseau de neurone à une couche cachée	26
Figure 12 : hyperplan qui sépare les deux ensembles de points	27
Figure 13 : Les lettres arabes	28
Figure 14 : Exemple de mot arabe présentant une ligature verticale et un chevauchement.....	29
Figure 15: Voyelles en arabe	30
Figure 16: Tableau récapitulatif précisant les caractéristiques et les performances de certains systèmes RC	35
Figure 17: Tableau récapitulatif précisant les caractéristiques et les performances de certains systèmes RC.....	37

Table de figure

Chapitre II : Conception

Figure 1: Système de reconnaissance des noms arabes manuscrits	42
Figure 2: (a) : Image originale, (b) : Image en niveau de gris, (c) : Image binarisée	47
Figure 3 : Exemple d'encadrement d'une lettre dans une image	51
Figure 4 : Représentation de la lettre arabe (ba) sous forme d'une matrice	53
Figure 5 : Schémas représentatif de données d'apprentissage.....	55
Figure 6 : Correction de l'inclinaison des lignes	59
Figure 7: exemple de segmentation en lignes	60
Figure 8: Représentation des éléments d'épaisseur uniforme	61
Figure 9: Image d'entrée résultante de la segmentation en ligne.....	61
Figure 10: Illustration de l'étape 1	65
Figure 11: Illustration de l'étape 3.....	67
Figure 12: illustre le résultat de l'algorithme de la segmentation.....	68
Figure 13 : Illustration de principe de Viterbi	69

Chapitre III : Réalisation

Figure 1 : Interface JAVA Eclipse	74
Figure2 : interface d'accueil	75
Figure 3 : Interface de prétraitement	76
Figure 4: interface d'apprentissage	77
Figure 5 : interface de reconnaissance	78

Table de figure

Figure 6 : interface de segmentation en lignes	79
Figure 7 : interface de reconnaissance de nom.....	79
Figure 8 : chargement de l'image de la lettre « alif »	80
Figure 9: Les résultats de prétraitement et l'extraction de primitives.....	81
Figure 10: les résultats de l'apprentissage.....	82
Figure 11: image de nom « محمد بن مصطفى ».....	82
Figure 12: la segmentation en lignes.....	83
Figure 13: résultat de l'extraction de caractères	84
Figure 14 : Reconnaissance de la lettre « ma ».....	85
Figure 15 : Reconnaissance de nom.....	86

Chapitre I

Eta de l'art

Introduction

Depuis son invention il y a plus de 5300 ans [9], l'écriture reste un moyen de communication privilégié entre les êtres humains. Bien que l'imprimerie créée il y a plus de 550 ans puis l'informatique aient permis son automatisation, l'écriture manuscrite est loin d'avoir disparu de notre société et les individus émettent et reçoivent une grande quantité de documents manuscrits.

Le traitement de masse de ces documents apparaît alors incontournable. Le traitement se donne pour objectif l'archivage, la recherche, la modification, la réutilisation et la transmission de l'information contenue dans ces documents. Une grande variété de documents manuscrits sont candidats à de tels traitement, nous citons entre autre les manuscrits arabes anciens.

Les manuscrits arabes anciens représentent un patrimoine culturel très précieux. Témoins d'une civilisation lointaine, ce sont de véritables trésors pour l'humanité. Ces documents se trouvent, souvent, soumis à une dégradation progressive et irréversible à cause de la corrosion. Ils sont menacés d'un réel danger de disparition et d'effritement à cause de leur structure physique (papier ancien, présence de bactéries, ...etc.) et du manque de moyens des lieux de conservation.

Par conséquent, la numérisation et la mise en ligne de ce type de document deviennent une démarche incontournable. Cette dernière permettra d'accéder à ces ressources sans aucune contrainte d'espace et de temps.

L'accès aux manuscrits numérisés ne saurait se faire sans un système d'indexation adéquat. Pour ce faire, le catalogage automatique des manuscrits s'avère indispensable dont le but d'extraire certaines informations clés. Parmi les informations les plus pertinentes du catalogue, le nom de l'auteur ou du copiste se trouvent en bonne posture.

Il est donc vivement recherché la reconnaissance des divers noms existants dans le manuscrit numérisé, ce qui nécessite la mise en œuvre d'un système de reconnaissance des noms arabes.

L'objectif de notre travail est la réalisation de système de reconnaissance de noms arabes manuscrit. Pour ce faire, nous proposons un système basé sur une méthode analytique en utilisant le modèle de Markov caché (MMC).

Pour mener à bien notre travail, nous avons organisé ce dernier en trois chapitres comme suit :

Introduction

Dans le premier chapitre, nous présentons les généralités et l'architecture des systèmes de reconnaissance de caractères, puis nous décrivons les différentes méthodes de reconnaissance de caractères (Markov Caché, réseau bayésien, réseau de neurone, machine à vecteur de support) et enfin, nous présentons quelques systèmes existants de reconnaissance de l'écriture arabe.

Dans le deuxième chapitre, nous présentons l'architecture de notre système de reconnaissance des noms arabes manuscrits, la description générale et détaillée de ses diverses parties ainsi que leur fonctionnement et les différents algorithmes utilisés pour réaliser notre système.

Dans le troisième chapitre, nous décrivons l'implémentation de notre système de reconnaissance de noms et les essais expérimentaux pour évaluer la performance du système réalisé.

Enfin, nous présentons nos perspectives après avoir conclu le mémoire et une glossaire pour définir et expliquer quelques concepts du domaine.

Chapitre I

Eta de l'art

I. Introduction

La reconnaissance de l'écriture manuscrite remonte à plus d'une quarantaine d'années. Les premiers systèmes de reconnaissance de caractères ont fait leur apparition dans les années 1950 et les premiers systèmes de lecture d'adresses postales utilisés pour le tri du courrier sont apparus en 1965. La lecture automatique de l'écriture manuscrite dans les formulaires a fait son apparition dans les années 1980 grâce au développement de moyens de calculs toujours plus puissants [9]. Ces dernières années, des progrès considérables ont été aussi réalisés dans la mise en œuvre de systèmes pour la reconnaissance de l'écriture manuscrite arabes vue son besoin dans divers domaine.

Nous présenterons dans ce chapitre les différents aspects d'un système de reconnaissance de caractères, ensuite nous présenterons le processus de reconnaissance de l'écriture manuscrite ainsi que ces différentes approches. Enfin nous présenterons quelques travaux de recherches sur l'écriture manuscrite arabe.

II. Définition de système de reconnaissance de caractère

La reconnaissance de caractères est le traitement machine de formes d'entrée de texte en vue de produire certaines données de sortie compréhensibles. De fait, les systèmes de reconnaissance de caractères constituent un sous-ensemble des systèmes de reconnaissance des formes. [1]

La reconnaissance des formes est la réduction méthodique d'information. A partir d'une donnée très riche, par exemple une image numérisée, on veut obtenir une information pertinente qui tient en quelques bits, par exemple l'indication que l'image contient une forme circulaire ou rectangulaire.

Parmi les domaines d'utilisation des SRC, le tri automatique du courrier, le traitement automatique de dossiers administratifs, des formulaires d'enquêtes, ou encore l'enregistrement des chèques bancaires et postaux.

III. Généralité sur les SRC

Il n'existe pas de système universel de reconnaissance de caractère (SRC) qui permet de reconnaître n'importe quel caractère dans n'importe quelle fonte.

Tout dépend du type de données traitées et bien évidemment de l'application visée [14]. Il existe plusieurs modes de classification des systèmes SRC parmi lesquels nous pouvons citer:

Les systèmes de reconnaissance de caractère qualifiés «en-ligne» ou «hors-ligne» suivant le mode d'acquisition. Un système de reconnaissance de caractère Mono-scripteur, Multi-scripteur ou Omni-scripteur selon le nombre de scripteurs et un système de reconnaissance de caractère globale ou analytique selon que l'analyse s'opère sur la totalité du mot ou par segmentation en caractères.

III.1. La reconnaissance En-Ligne (on-line)

La reconnaissance En-Ligne s'opère en temps réel (pendant l'écriture). Les symboles sont reconnus au fur et à mesure qu'ils sont écrits à la main.

L'acquisition de l'écrit est généralement assurée par un appareil en ligne (voir la figure 1) fonctionne avec un stylo et comprend des affichages-tablettes et des tablettes graphiques. Ces tablettes peuvent fournir l'ordre temporel des points qui constituent les lignes de texte. Certaines tablettes fournissent d'autres renseignements, notamment la vitesse (de l'écriture) et la pression (exercée par l'utilisateur) [1].

Le mode de reconnaissance en-ligne est réservé généralement à l'écriture manuscrite, il présente un avantage majeur qui est la possibilité de correction et de modification de l'écriture de manière interactive, vue la réponse en continu du système.



Figure 1 : Systèmes d'acquisition en ligne [10].

III.2. La reconnaissance hors-ligne (off-line):

La reconnaissance hors-ligne démarre après l'acquisition de l'information, elle convient aux documents imprimés et les manuscrits déjà rédigés. Ce mode peut être considéré comme le cas le plus général de la reconnaissance de l'écriture [1] [2].

III.3. Un système Mono-scripteur, Multi-scripteur et Omni-scripteur

La difficulté de reconnaissance croît avec le nombre de scripteur, divisant l'échelle en trois : mono, multi et omni-scripteurs [2].

Un système est dit mono-scripteur (propres au scripteur) si le système ne peut reconnaître qu'une seule écriture.

Un système est dit multi-scripteur (propres à l'écriture manuscrite) si le système peut identifier et reconnaître l'écriture pour un certain nombre de scripteurs.

Un système est dit omni-scripteur (propres à n'importe quelle écriture manuscrite) si le système doit être capable de généraliser son apprentissage (voir la section IV.4.1) à n'importe quel type d'écriture.

III.4. Approches de reconnaissance

Il y a deux façons d'aborder la problématique de reconnaissance des mots. Soit le système reconnaît le mot comme une entité entière et indivisible, il s'agit d'une approche globale ou holistique. Soit il reconnaît le mot à partir de ses caractères préalablement segmentés, il s'agit d'une approche analytique :

III.4.1. L'approche globale

Elle se base sur une description unique de l'image du mot, vue comme une entité indivisible. Disposant de beaucoup d'informations, en effet, la discrimination de mots proches est très difficile, et la réalisation des modèles pour la reconnaissance de caractère nécessite une grande quantité d'échantillon qui est souvent difficile à réunir [3].

III.4.2. L'approche analytique

Elle consiste à décomposer le mot en une séquence de caractères intermédiaires, qui font partie d'un alphabet prédéfini. La reconnaissance du mot complet sera obtenue par la combinaison des reconnaissances de ces caractères intermédiaires. Il est donc nécessaire de découper le mot à reconnaître en une séquence de symboles, ce qui n'est pas toujours trivial.

Ces symboles intermédiaires seront éventuellement recomposés pour former des caractères. Cette suite de caractères forme le mot à reconnaître [3].

Chacune de ces deux approches a ses avantages et ses inconvénients. L'avantage majeur de l'approche analytique se trouve dans la possibilité de reconnaître n'importe quel mot, cependant, sa grande faiblesse est dans le processus de segmentation qui n'est pas toujours trivial et demande beaucoup de temps de calcul. Quant à l'approche globale, elle est parfaitement envisageable pour la reconnaissance, moyennant un lexique limité, même pour des mots dégradés. Néanmoins, l'augmentation de la taille du lexique risque d'augmenter exponentiellement le taux de confusion [15].

IV. Architecture générale de SRC manuscrit

Un SRC peut être effectivement morcelé en plusieurs composants, la figure 2 nous montre un schéma général d'un système de reconnaissance de caractères :

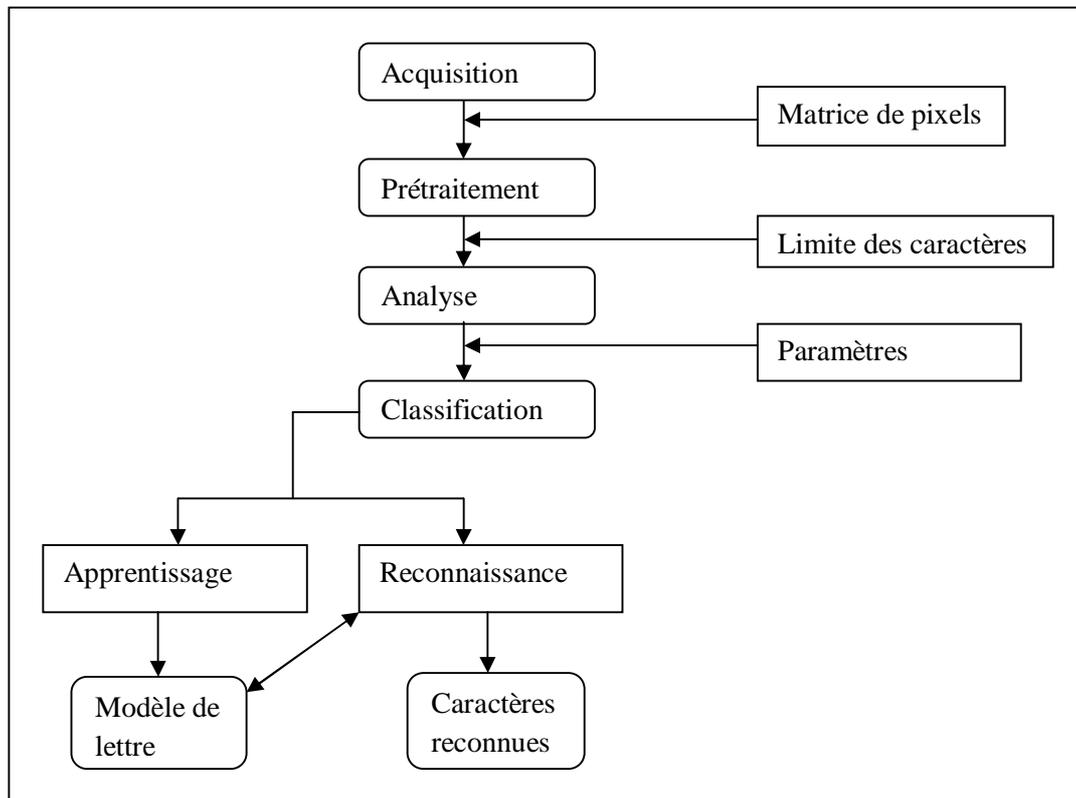


Figure 2: Schéma général d'un système de reconnaissance de caractères [2].

Un système de reconnaissance fait appel généralement, aux étapes suivantes : après l'acquisition de données, l'un de ces composants se charge des fonctions de « prétraitement », comme la normalisation. Une fois que la forme d'entrée a été prétraitée, un autre composant l'accepte et en extrait les attributs caractéristiques (boucle, présence et nombre de point diacritique).

Les caractéristiques ainsi extraites sont utilisées par un composant de « classification » pour attribuer une étiquette à la forme. Toutes les fonctions menées après la classification (initiale) font partie du « post-traitement ».

Chacun des composants fonctionnels d'un SRC est décrit avec plus de précision dans ce qui suit :

IV.1. Phase d'acquisition

L'acquisition permet la conversion du document papier (forme analogique) sous la forme d'une image numérique. Cette étape est importante car elle se préoccupe de la préparation des documents à saisir, du choix et du paramétrage du matériel de saisie (scanner), ainsi que du format de stockage des images [2].

La numérisation du document est opérée par balayage optique. Le résultat est rangé dans un fichier de points, appelés pixels, dont la taille dépend de la résolution (voir l' glossaire). La figure 3 montre différents niveaux de résolution utilisés pour le même document. On peut remarquer la dégradation occasionnée par 75 ppp (point per pitch), l'insuffisance des 300 ppp pour le graphique, et l'inutilité des 1200 ppp pour l'ensemble.

La technicité des matériels d'acquisition (scanner) a fait beaucoup de progrès ces dernières années. On trouve aujourd'hui des scanners pour des documents de différents types (feuilles, revues, livres, photos, etc.). La résolution par défaut est de l'ordre de 300 à 1200 ppp selon les modèles.

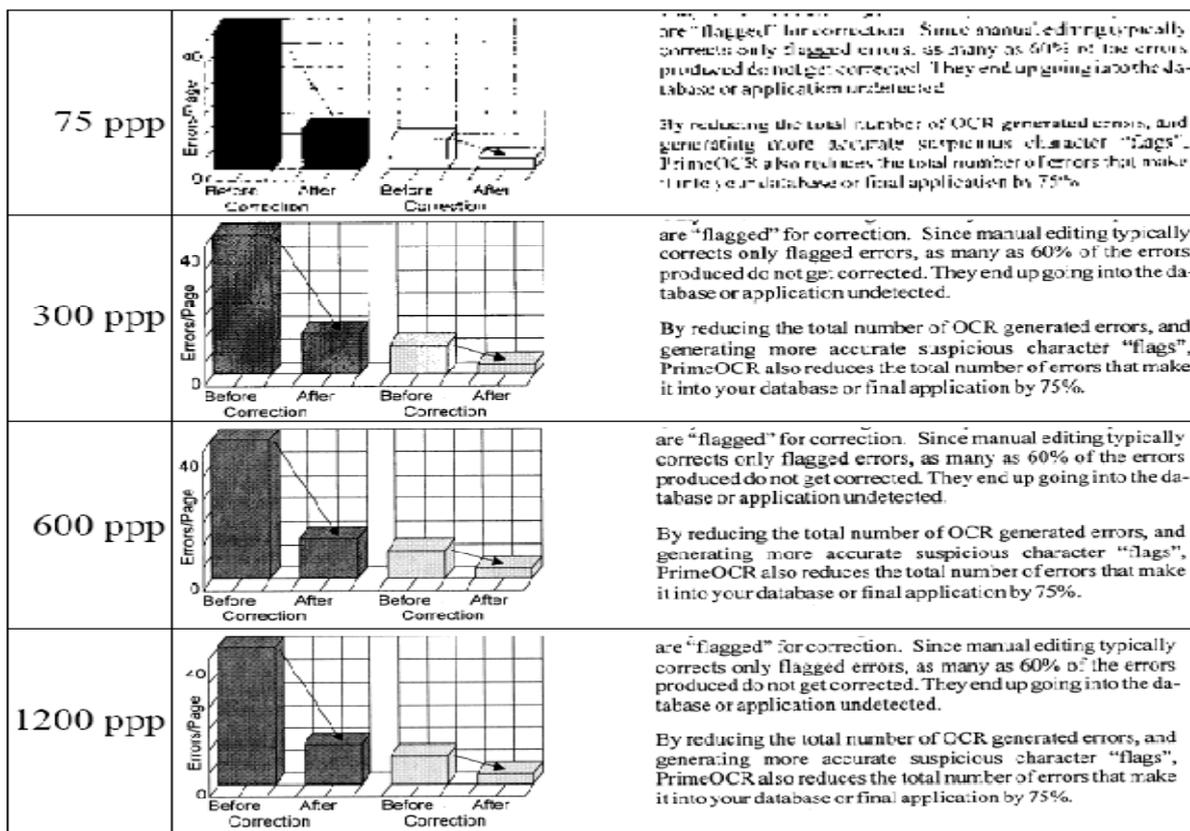


Figure 3: Différents niveaux de résolution [5].

IV.2. Prétraitements

Avant de procéder à la reconnaissance de mots, il est nécessaire d'effectuer une série de prétraitements sur l'image de celui-ci en vue d'éliminer (ou du moins réduire) le bruit qui l'encrasse, et simplifier la procédure d'extraction de caractéristiques et le processus de segmentation. La phase de prétraitement est une étape facultative. Suivant les systèmes on trouve plus ou moins de prétraitements parmi lesquels nous pouvons citer :

- a. **La binarisation** : elle permet de passer d'une image de niveaux de gris (voir l' glossaire) à une image binaire (voir l' glossaire) composée de deux valeurs 0 et 1 [5] .
- b. **Lissage**: L'image des caractères peut être entachée de bruits dus aux artefacts de l'acquisition et à la qualité du document, conduisant soit à une absence de points ou à une surcharge de points. Les techniques de lissage permettent de résoudre ces problèmes par des opérations locales qu'on appelle opérations de bouchage et de nettoyage [2]. L'opération de nettoyage permet de supprimer les petites tâches et les excroissances de la forme. Pour le bouchage il s'agit d'égaliser les contours et de boucher les trous internes à la forme du caractère en lui ajoutant des points noirs [14].
- c. **La squelettisation** : Elle est une opération qui permet de passer d'une image à sa représentation en "fil de fer". Le squelette a un pixel d'épaisseur. C'est une manière de représenter l'information indépendamment de l'épaisseur initiale de l'écriture [3].

Il permet d'extraire des caractéristiques importantes, comme les intersections et le nombre de tracés, leurs positions relatives. Il est également possible de renormaliser l'épaisseur de l'écriture à partir du squelette.

Il n'existe pas de définition unique du squelette. Le squelette doit seulement remplir trois conditions :

- Il doit être aussi fin que possible (typiquement, 1 pixel d'épaisseur),
- Il doit respecter la connexité,
- Il doit être centré dans la forme qu'il représente.

A chaque composante connexe de la forme, le squelette correspondant ne doit être composé que d'une seule composante connexe d'un pixel d'épaisseur incluse dans la première.

d. Normalisation des mots: La normalisation tend à réduire les variations entre les styles, tailles et orientations d'écriture. La tendance est à réduire l'importance de cette phase pour prendre en compte cette variabilité au niveau des modèles. Parmi les traitements de normalisation, nous pouvons citer [6] :

- Normalisation de la taille : consiste à transformer la taille de l'image et l'adapter à une dimension fixe.
- Mise à l'horizontale de la ligne d'écriture : un défaut d'orientation du support pendant l'acquisition, ou une écriture imprécise peuvent conduire à une inclinaison et une déformation de la ligne de base. Des traitements sont appliqués pour la rendre horizontale. Cette ligne de base du texte (sur laquelle reposent les caractères) est utilisée ensuite pour l'extraction des primitives.
- de l'inclinaison des caractères : on remarque que très souvent, les caractères sont inclinés : leur direction générale forme un angle avec la verticale variable selon les scripteurs.

Les systèmes de reconnaissance corrigent toujours cette inclinaison pour diminuer la variabilité intra-classe et pour rendre la segmentation en caractères plus facile. Les techniques de correction sont en général fondées sur la détection des traits quasi verticaux. On calcule leur inclinaison moyenne par rapport à la verticale et on corrige l'inclinaison de tous les mots avec ce même angle moyen. La figure 4 résume les différentes étapes de prétraitements :

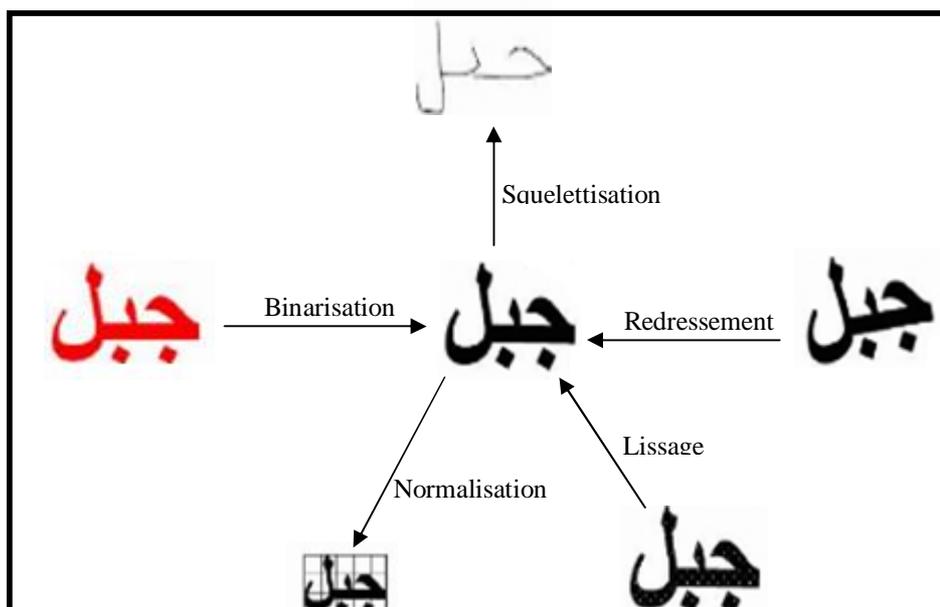


Figure 4: effets des opérations de prétraitement.

IV.3. Phase d'analyse

La phase d'analyse regroupe la segmentation et l'extraction des primitives :

IV.3.1. La segmentation

L'une des difficultés de la reconnaissance de l'écriture est sa segmentation en sous séquences stationnaires, pour leur reconnaissance. On distingue trois types de segmentations : segmentation explicite, implicite et sans segmentation.

IV.3.1.1. La segmentation explicite

Les méthodes dites avec segmentation externe (explicite) ou "OUTSEG", s'appuient sur un découpage de l'image en sous-unités qui ne sont pas directement des lettres mais en graphème grâce à des règles morphologiques (voir l' glossaire) ou topologiques (voir l' glossaire) [8].

IV.3.1.2. La segmentation implicite

Elle fournit des modèles de lettres. Ainsi elle évite, comme dans le premier cas, de faire des erreurs de segmentation, la méthode est bien plus sûre pour reconnaître par la suite les caractères. Un problème se pose lors de l'établissement de la frontière entre les caractères. Nous n'avons fourni que des modèles de lettre, la position de la coupure peut varier sensiblement d'un passage sur un mot à un autre [7].

La figure 5 illustre les deux types de segmentation :

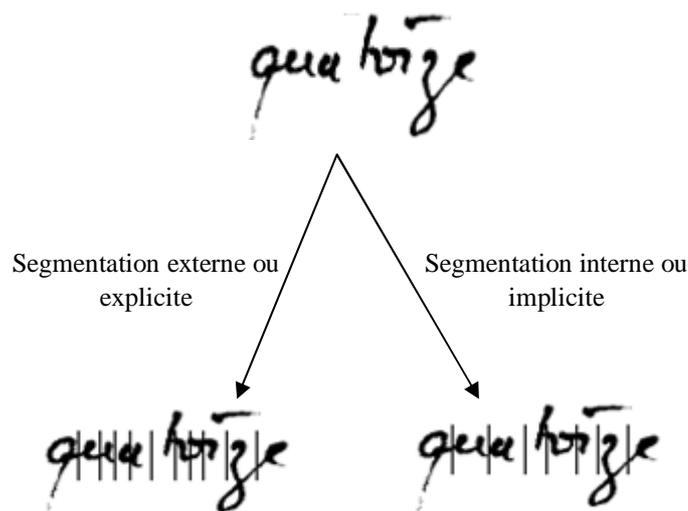


Figure 5: Segmentation implicite et explicite.

IV.3.1. 3. La méthode sans segmentation

La dernière méthode, et vraisemblablement la plus efficace, est de ne pas segmenter en lettre. La reconnaissance de caractères va s'effectuer à partir du mot. Ceci permet d'éviter les erreurs dues à la segmentation, mais reporte le problème sur l'objet qui analyse les caractères. Cependant celui-ci ne pourra pas être contraint à étudier des morceaux de caractères en essayant de les associer à un caractère qu'il croit reconnaître. L'apprentissage doit donc être fait sur des mots, ce qui va compliquer davantage son algorithme et son apprentissage [7].

IV.3.2. L'extraction des primitives

L'extraction de primitives consiste à transformer une image (caractère, graphème, bande verticale, ...) en un vecteur de primitives de taille fixe. Cette transformation revient à changer l'espace de représentation des données, du plan de l'image vers un espace à N dimensions.

Le choix des primitives est critique, et influence nettement le résultat de reconnaissance. Ces primitives doivent avoir deux propriétés :

- être discriminantes : permettre une bonne différenciation entre les classes de symboles à reconnaître.
- maintenir un nombre de dimensions limité, afin d'éviter le phénomène de malédiction de la dimensionnalité.

Les types de caractéristiques peuvent être classés en quatre groupes principaux : caractéristiques structurelles, caractéristiques statistiques, transformations globales, et superposition des modèles et corrélation.

IV.3.2.1. Caractéristiques structurelles

Les caractéristiques structurelles décrivent une forme en termes de sa topologie et de sa géométrie en donnant ses propriétés globales et locales. Parmi ces caractéristiques on peut citer: les traits et les anses dans les différentes directions ainsi que leurs tailles, les points terminaux, les points d'intersections, les boucles, le nombre de points diacritiques et leur position par rapport à la ligne de base, les voyellations et les zigzags (hamza), la hauteur et la largeur du caractère, la catégorie de la forme (partie primaire ou point diacritique, etc.) [9].

Plusieurs autres caractéristiques peuvent être tirées, suivant qu'elles soient extraites d'une courbe, d'un trait ou d'un segment de contour.

IV.3.2.2. Les caractéristiques statistiques

Les caractéristiques statistiques décrivent une forme en termes d'un ensemble de mesures extraites à partir de cette forme. Les caractéristiques utilisées pour la reconnaissance de textes arabes sont : le zonage (zoning) et les caractéristiques de lieu géométrique (Loci)[2].

a. Le zonage consiste à superposer une grille $n \times m$ sur l'image du caractère et pour chacune des régions résultantes, calculer la moyenne ou le pourcentage de points en niveaux de gris, donnant ainsi un vecteur de taille $n \times m$ de caractéristiques [9]. Voir la figure 6 :

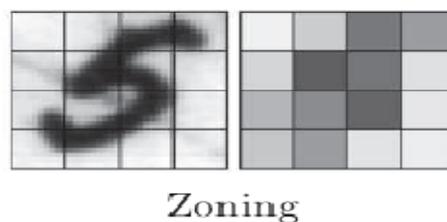


Figure 6 : Illustrations de primitive Zoning [9]

b. La méthode Loci est basée sur le calcul du nombre de segments blancs et de segments noirs le long d'une ligne verticale traversant la forme, ainsi que leurs longueurs [2].

IV.3.2.3. Les transformations globales

La transformation consiste à convertir la représentation en pixels en une représentation plus abstraite pour réduire la dimension des caractères, tout en conservant le maximum d'informations sur la forme à reconnaître [9].

Une des transformations les plus simples est celle qui représente le squelette ou le contour d'un caractère sous forme d'une chaîne de codes de directions. La chaîne de code obtenue est souvent simplifiée pour réduire les redondances et les changements brusques de direction.

IV.3.2.4. Superposition des modèles (template matching) et corrélation

La méthode de Template Matching appliquée à une image binaire (en niveaux de gris ou squelettes), consiste à utiliser l'image de la forme comme vecteur de caractéristiques pour être comparé à un modèle (template) pixel par pixel dans la phase de reconnaissance et une mesure de similarité est calculée [40].

IV.4. La classification

Le rôle du classifieur est de déterminer l'appartenance d'une forme à une classe à partir des vecteurs de primitives extraits de cette forme. La classification dans un SRC regroupe deux tâches : d'une part, l'apprentissage et d'autre part, la reconnaissance et décision.

IV.4.1. L'apprentissage

Il s'agit, lors de cette étape, d'apprendre au système les propriétés pertinentes du vocabulaire utilisé et de l'organiser en modèles de références.

L'idéal serait d'apprendre au système autant d'échantillons que de formes d'écritures différentes, mais cela est impossible à cause de la grande variabilité de l'écriture qui conduirait à une explosion combinatoire de modèles de représentation. La tendance consiste alors à remplacer le nombre par une meilleure qualité des traits caractéristiques. L'apprentissage consiste en deux concepts différents : **l'entraînement** et **l'adaptation**.

L'entraînement consiste à enseigner au système la description des caractères tandis que l'adaptation sert à améliorer les performances du système en profitant des expériences précédentes. Certains systèmes permettent à l'utilisateur d'identifier un caractère lorsqu'ils échouent à le reconnaître et ils utilisent l'entrée de l'utilisateur à chaque fois que le caractère est rencontrée [2].

Les procédés d'apprentissage sont différents selon qu'il s'agisse de reconnaissance de caractères imprimés ou manuscrits. D'une manière générale, on distingue deux types de techniques d'apprentissage : **supervisé** et **non supervisé**.

- L'apprentissage est dit supervisé s'il est guidé par un superviseur appelé professeur. Il est réalisé lors d'une étape préliminaire de reconnaissance en introduisant un grand nombre

d'échantillons de référence. Le professeur indique dans ce cas le nom de chaque échantillon. Le choix des caractères de référence est fait à la main en fonction de l'application. Le nombre d'échantillons peut varier de quelques unités à quelques dizaines, voir même quelques centaines par caractère.

- L'apprentissage non supervisé ou sans professeur consiste à doter le système d'un mécanisme automatique qui s'appuie sur des règles précises de regroupement pour trouver les classes de référence avec une assistance minimale. Dans ce cas les échantillons sont introduits en un grand nombre par l'utilisateur sans indiquer leur classe.

IV.4.2. Reconnaissance et décision

La décision est l'ultime étape de reconnaissance. A partir de la description en paramètres du caractère traité, le module de reconnaissance cherche parmi les modèles de référence en présence, ceux qui lui sont les plus proches.

La reconnaissance peut conduire à un succès si la réponse est unique (un seul modèle répond à la description de la forme du caractère). Elle peut conduire à une confusion si la réponse est multiple (plusieurs modèles correspondent à la description). Enfin elle peut conduire à un rejet de la forme si aucun modèle ne correspond à sa description. Dans les deux premiers cas, la décision peut être accompagnée d'une mesure de vraisemblance, appelée aussi score ou taux de reconnaissance.

Les approches de reconnaissance peuvent être regroupées en trois groupes principaux: l'approche statistique, l'approche structurelle, et l'approche stochastique.

IV.4.2.1. Approche statistique

La classification statistique est une manière de différencier divers objets en utilisant des lois de probabilité et de statistiques. Elle peut être utilisée pour déterminer à quelle classe appartient un caractère que l'on a isolé précédemment [7].

Les approches statistiques bénéficient des méthodes d'apprentissage automatique qui s'appuient sur des bases théoriques fondées, telles que la théorie de la décision bayésienne, les méthodes de classification non supervisées, etc.

Nous pouvons citer trois méthodes statistiques parmi celles les plus couramment utilisées : L'approche bayésienne, les réseaux de neurones, Machines à Vecteurs de Support (SVM) (voir la section V).

IV.4.2.2. Approche structurelle

Les méthodes structurelles reposent sur la structure physique des caractères. Elles cherchent à trouver des éléments simples ou primitifs, et à décrire leurs relations. Les primitives sont de type topologiques telles que : une boucle, un arc... et une relation peut être la position relative d'une primitive par rapport à une autre [2]. Parmi les méthodes structurelles nous pouvons citer :

- a. Les méthodes de tests :** Elles consistent à appliquer sur chaque caractère traité des tests de plus en plus fins sur la présence ou l'absence de primitives, de manière à répartir les échantillons en classes. Le processus le plus habituel consiste à diviser à chaque test l'ensemble des choix en deux jusqu'à n'obtenir qu'une seule forme correspondante au caractère entré. Ce choix dichotomique est très rapide et très simple à mettre en œuvre, mais il est très sensible aux variations du tracé.
- b. La comparaison de chaînes :** Les caractères sont représentés par des chaînes de primitives. La comparaison du caractère traité avec le modèle de référence, consiste à mesurer la ressemblance entre les deux chaînes et à se prononcer sur celui-ci. La mesure de ressemblance peut se faire par calcul de distance ou par examen de l'inclusion de toute ou une partie d'une chaîne dans l'autre [2].

IV.4.2.3. Approche stochastique

Contrairement aux méthodes précédemment décrites, l'approche stochastique utilise un modèle pour la reconnaissance, prenant en compte la grande variabilité de la forme. La distance communément utilisée dans les techniques de « comparaison » est remplacée par des probabilités calculées de manière plus fine par apprentissage. La forme est considérée comme un signal continu observable dans le temps à différents endroits constituant des états « d'observations ». Le modèle décrit ces états à l'aide de probabilités de transitions d'états et de probabilités d'observation par état.

La comparaison consiste à chercher dans ce graphe d'états, le chemin de probabilité forte correspondant à une suite d'éléments observés dans la chaîne d'entrée [2]. Les méthodes les plus répandues dans cette approche sont les méthodes utilisant les modèles de Markov cachés (H.M.M) (voir la section V).

IV.5. Post-traitement

Le post-traitement comprend la vérification, l'exécution de l'action et l'adaptation. L'objectif de la vérification est d'accroître le niveau de confiance dans la classification effectuée; une telle vérification peut être effectuée de diverses façons. L'une de ces façons consiste à utiliser une base de données comportant des combinaisons de 2 ou 3 lettres pour vérifier si la séquence des lettres reconnues ne comprend pas de combinaisons impossibles. Une autre possibilité est d'utiliser un dictionnaire pour vérifier si une certaine séquence de caractères constitue un mot valide. En règle générale, cette méthode est moins fiable puisque les mots exacts qui ne sont pas consignés au dictionnaire sont rejetés.

En plus des dictionnaires contenant des lettres et/ou des mots, un modèle grammatical formel de niveau plus élevé peut être utilisé pour vérifier l'exactitude d'expressions ou de phrases entières [1].

Remarque : Il est bon de noter [1]:

- qu'il n'y a pas nécessairement de trouver toutes les parties fonctionnelles (prétraitement, classification, apprentissage, ...) dans tous les SRC. Un SRC peut prendre charge la fonction de classification sans avoir auparavant explicitement extrait les caractéristiques.
- que des parties fonctionnelles ne sont pas toujours mises en place à titre de composants incompatibles. Ainsi, un objet logiciel peut extraire des caractéristiques et les classifier simultanément.
- que des parties fonctionnelles n'interviennent pas nécessairement en séquence. De fait, dans de nombreuses applications, un nombre important d'extractions de caractéristiques sont effectuées avant la segmentation.

V. Méthode de reconnaissance de caractères

Dans la suite de cette section, nous présenterons un ensemble de reconnaisseurs couramment utilisés en reconnaissance de l'écriture, et nous discuterons de leurs principes et leurs

fonctionnements. Nous passerons en revue le modèle de Markov caché, les réseaux de neurones, les réseaux bayésiens et les machines à vecteur support.

V.1. Le modèle de Markov caché (MMC)

Les modèles de Markov cachés sont introduits par Baum et All dans les années 60 [16]. Ils sont des modèles stochastiques, largement utilisés en reconnaissance de la parole et plus tardivement en reconnaissance de l'écrit. Les MMC permettent de calculer la probabilité d'appartenance d'une forme à une classe en fonction de degré de distorsion subi par cette forme.

V.1.1. Définition de MMC

Un modèle de Markov caché (Hidden Markov Model : HMM) est un automate probabiliste doublement stochastique. Il est caractérisé par les matrices de probabilités de transition d'états A , de probabilités des symboles d'observation B et le vecteur de probabilités d'état initial Π [15].

Les notations suivantes sont formellement définies pour un HMM:

N : nombre d'états;

M : nombre de symboles d'observation;

T : la longueur de la séquence d'observation;

$S = \{s_t\}$ l'ensemble d'états, $s_t \in \{1, 2, \dots, N\}$, $1 \leq t \leq T$,

$O = \{o_k\}$: l'ensemble discret des symboles d'observations possibles, $1 \leq k \leq M$

$A = \{a_{ij}\}$: la distribution de probabilité de transition d'état, où $a_{ij} = P(s_{t+1} = j | s_t = i)$, $1 \leq i, j \leq N$,

$B = \{b_j(o_k)\}$: la distribution de probabilité de symbole d'observation où $b_j(o_k) = p(o_k \text{ à } t | s_t = j)$, $1 \leq j \leq N$, $1 \leq k \leq M$,

$\Pi = \{\Pi_i\}$: la distribution de probabilité d'état initial où $\Pi_i = p(s_1 = i)$, $1 \leq i \leq N$,

Nous pouvons, d'une façon plus compacte désigner l'ensemble des paramètres d'un HMM par $\lambda = (A, B, \Pi)$, Par conséquent, un HMM est complètement spécifié par λ .

Les MMC permettent de modéliser par des états différents chaque sous-partie statistiquement stable de la séquence d'observations. C'est-à-dire que toutes les observations modélisées par

chaque état sont représentées par des vecteurs de primitives (voir la figure 7) [8]. La suite d'états est conditionnée par des probabilités de transitions. A chaque état est associée une probabilité d'émission de symboles observables [10].

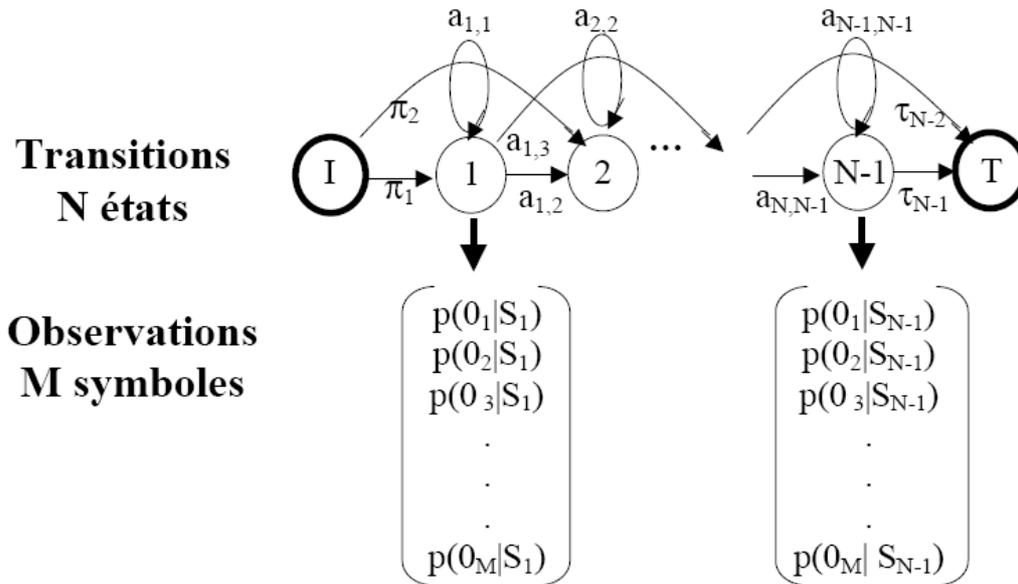


Figure 7: Définition des paramètres d'un Modèle de Markov Caché [8]

Chaque état S émet selon sa propre distribution de probabilités $p(O_i|S_j)$, des observations O , qui constituent la séquence effectivement observée de longueur T . La séquence d'observations est donc le résultat de deux processus stochastiques : les transitions entre états et les émissions des observations.

L'une des difficultés de la reconnaissance de l'écriture est sa segmentation en sous séquences stationnaires, pour leur reconnaissance.

Or, selon la contradiction bien connue, il n'est pas possible de segmenter sans avoir reconnu les parties, ni de les reconnaître sans avoir segmenté la séquence [3]. Les MMC permettent d'appliquer le paradigme de segmentation-reconnaissance simultanées. Après reconnaissance complète, il est possible de faire une segmentation globalement optimale du signal en zones stationnaires modélisées (phonèmes ou graphèmes, mots . . .).

V.1.2. Types de modèle de Markov caché

On distingue trois types de modèle de Markov correspondants aux trois principales stratégies de segmentation (Voir la figure 8).

1. **Modèle de mot à partir de ses constituants lettres ou graphèmes** (segmentation explicite). L'étape préalable à l'utilisation de ce modèle est la segmentation du mot en lettres ou en graphèmes. Le modèle est construit essentiellement sur les bi-grammes ou les trigrammes (deux ou trois lettres consécutives dans un mot).
2. **Modèle de lettre** (segmentation implicite). Il y a un modèle de Markov caché par mot du lexique. Ce modèle de mot est généralement construit par concaténation des modèles de lettres.
3. **Modèle de mot à partir de ses primitives** (sans segmentation) : Un mot peut être représenté par une suite d'éléments orientée gauche droite tel que des primitives, ou des graphèmes voire des lettres. Pour reconnaître le mot on cherche à retrouver dans le lexique le mot qui lui correspond. Pour trouver le mot le plus vraisemblable, on calcule à partir des données une probabilité d'observations de chaque élément dans chaque état et on probabilise les transitions [10].

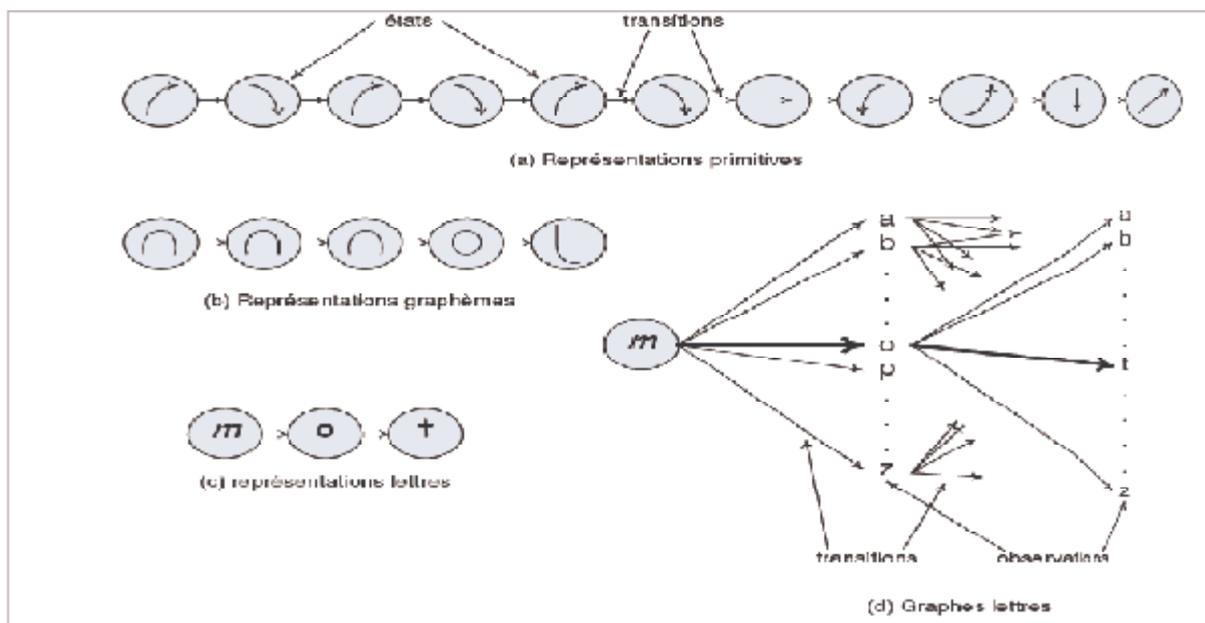


Figure 8 : Différentes représentations markoviennes du mot [10].

L'utilisation des HMMs à des fins de reconnaissance est conditionnée par la résolution des trois problèmes qui leur sont associés :

1. **Problème d'évaluation** : soit un modèle λ et une séquence d'observations $O = O_1, O_2, \dots, O_T$, comment peut-on calculer la probabilité $P(O/\lambda)$? La solution à ce problème est donnée par l'algorithme de Forward-Backward (voir glossaire).

2. **Problème de décodage** : soit une séquence d'observations $O = O_1, O_2, \dots, O_T$ et un modèle λ , comment peut-on déterminer la suite d'états dans λ qui a généré O ? La solution à ce problème est donnée par l'algorithme de Viterbi (voir chapitre II section III.2.2.4).

3. **Problème d'apprentissage** : soit un ensemble de séquences d'observation et un modèle initial λ , comment peut-on réestimer les paramètres de λ pour pouvoir maximiser la vraisemblance de génération de cet ensemble de séquences ? La solution à ce problème est donnée par l'algorithme de Baum-Welch (voir chapitre II la section III.2.1.3.1) [17].

V.2. Réseaux Bayésiens

Initiés par Judea Pearl dans les années 1980 [16], Les réseaux bayésiens se révèlent être des outils très pratiques pour représenter et manipuler les connaissances. Ils ont été historiquement développés dans le domaine de la prise en compte de l'incertain dans les systèmes de décision et les systèmes experts. Le domaine de la classification de documents en particulier est un domaine pour lequel les classificateurs bayésiens sont souvent utilisés avec succès. Ces classificateurs utilisent des méthodes basées sur le théorème de Bayes afin de déterminer les probabilités d'associer certaines classes à certaines instances.

V.2.1. Définition

Un réseau bayésien est un système représentant la connaissance et permettant de calculer des probabilités conditionnelles (voir glossaire) apportant des solutions à différentes sortes de problématiques [16].

V.2.2. Apprentissage

La phase dite d'apprentissage consiste à trouver la structure du réseau Bayésien (apprentissage de structure) et l'estimation des probabilités conditionnelles (apprentissage des paramètres) [16].

Apprentissage des paramètres

Elle peut être à partir de données complètes ou à partir de données incomplètes :

- A partir de données complètes : Dans le cas où toutes les variables sont observées, la méthode la plus simple et la plus utilisée est l'estimation statistique qui consiste à estimer la probabilité d'un événement par la fréquence d'apparition de l'événement dans la base de données.
- A Partir de données incomplètes : Dans les applications pratiques, les bases de données sont très souvent incomplètes. Certaines variables ne sont observées que partiellement ou même jamais. De nombreuses méthodes tentent d'estimer la densité de probabilité des données manquantes.

Apprentissage de la structure

Généralement, la structure est définie par des experts. Lorsque ce n'est pas le cas, il est possible d'utiliser des algorithmes pour construire le réseau. Parmi ces algorithmes il y a ceux qui consistent à rechercher les différentes relations causales qui existent entre les variables, et d'autres qui essaient de quantifier l'adéquation d'un réseau bayésien au problème à résoudre, c'est-à-dire : associer un score à chaque réseau bayésien. Puis elles recherchent la structure qui donnera le meilleur score.

V.2.3. Classification avec les réseaux bayésiens

Les classificateurs bayésiens utilisent des méthodes basées sur le Théorème de Bayes afin de déterminer les probabilités d'associer certaines classes à certaines instances selon les données d'entraînement.

Théorème de bayes: [16]

$$P(A/B) = \frac{P(B/A)*P(A)}{P(B)}$$

P (A) : Probabilité **a priori**,

P (A/B): Probabilité **a posteriori**.

V.3. Réseau de neurone

Bien que leur fonctionnement en « boîte-noire » ne permette pas d'interpréter les valeurs de leurs paramètres internes, leur facilité de mise en œuvre et leurs bonnes performances ont fait des réseaux de neurones l'un des outils les plus utilisés en apprentissage automatique au cours des 20 dernières années.

Un neurone est l'unité élémentaire de traitement d'un réseau de neurones. Il est connecté à des sources d'information en entrée (d'autres neurones par exemple) et renvoie une information en sortie. [18]

Les entrées d'un neurone correspondent à la description d'une observation x , décrite par un vecteur de n attributs (x_1, \dots, x_n) numériques. Chacune de ces composantes x_i est pondérée par un poids synaptique w_i . Il y a toutes fois un poids supplémentaire w_0 supposant lié à une information $x_0 = -1$.

La sortie z du neurone est obtenue en deux phases, la première consiste à calculer la somme pondérée des entrées. Nous définirons une nouvelle donnée in , par : [19]

$$in = \sum_{i=0}^k w_i * x_i = \left(\sum_{i=1}^k w_i * x_i \right) - w_0$$

Cette valeur (in) est passée à la fonction de transfert (appelée également fonction d'activation, ou fonction de seuil) qui calcule la valeur de l'état du neurone.

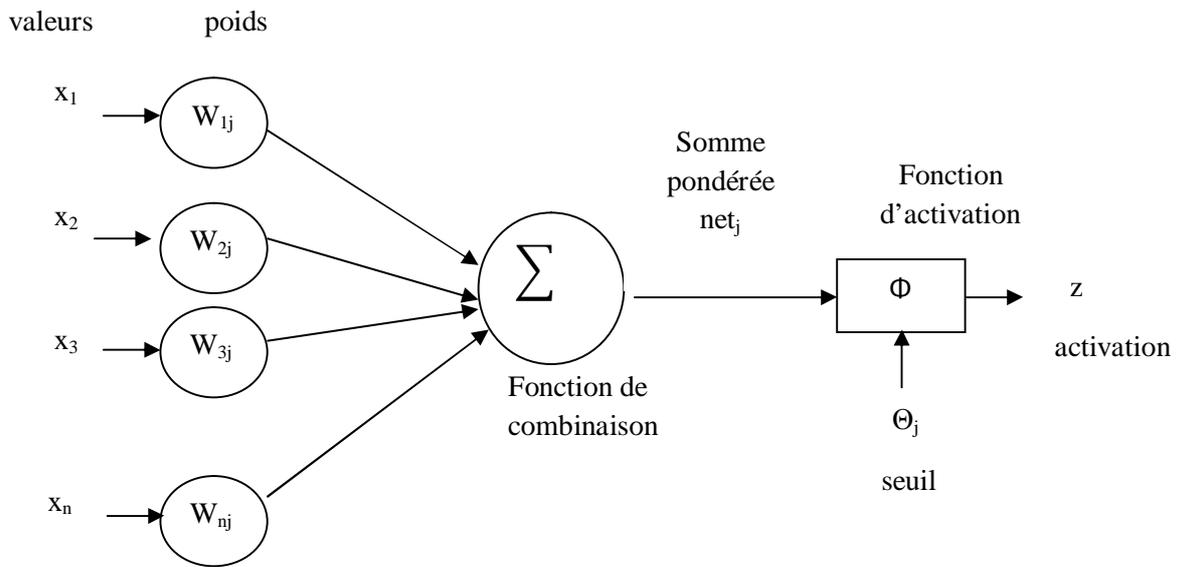


Figure 9: Neurone artificiel [7]

En pratique, les neurones sont disposés en couches (voir figure 11), et tous les neurones d'une couche donnée ont la même fonction d'activation (les fonctions d'activation les plus courantes sont données dans le tableau suivant).

Fonction linéaire	$\sum_i w_i \cdot x_i + w_0$
Fonction sigmoïde	$\frac{1}{1 + e^{-\lambda y}}$
Fonction tanh	$\frac{e^{2y} - 1}{e^{2y} + 1}$
Fonction softmax	$\frac{e^y}{\sum_i e^{y_i}}$
Fonction à base de radiale de centre x_c	$\exp\left(-\frac{\ x - x_c\ ^2}{\sigma^2}\right)$

Figure 10 : Tableau des fonctions de transfert

Le plus souvent, l'apprentissage d'un réseau de neurones se fait de façon itérative, par rétro-propagation (voir glossaire) du gradient d'erreur. Les fonctions d'activation des neurones doivent être dérivables par rapport aux entrées (les entrées d'une couche étant les sorties de la couche précédente) de manière à pouvoir rétro-propager le gradient d'erreur de couche en couche.

De plus, si la fonction de transfert est également dérivable par rapport aux paramètres du système (les w_i , qui sont les poids synaptiques), il est possible d'optimiser ces poids de manière itérative, par descente de gradient.

La réestimation itérative d'un coefficient $w_i \in W$ par l'algorithme de descente de gradient s'écrit: [3]

$$W_i(t+1) = w_i(t) - \lambda \frac{\partial \text{Err}}{\partial w_i} (W)$$

λ est le coefficient d'apprentissage. Ce poids est important : s'il est trop grand, il risque de faire diverger l'apprentissage, s'il est trop faible l'erreur risque de converger vers un minimum local.

V.3.1. Type de réseaux de neurones

Il existe une grande variété de réseaux de neurones. Ils diffèrent par leurs architectures et leurs modes de fonctionnement. Les plus connus: perceptron monocouche et perceptron multicouches.

Perceptron monocouche

Le perceptron a été inventé en 1957 par Rosenblatt. Il est vu comme le type de réseau de neurones le plus simple.

Il est caractérisé de la manière suivante :

- Il possède n informations en entrée ;
- Il est composé de p neurones, que l'on représente généralement alignés verticalement. Chacun peut en théorie avoir une fonction d'activation différente. En pratique, ce n'est généralement pas le cas ;
- Chacun des p neurones est connecté aux n informations d'entrée.

Le réseau de neurones possède ainsi n informations en entrée et p sorties, chaque neurone renvoyant sa sortie.

Une utilisation courante est que chaque neurone de la couche représente une classe. Pour un exemple \mathbf{X} donné, on obtient la classe de cet exemple en prenant la plus grande des p sorties.

Perceptron multicouches

Le perceptron multicouches ou PMC représentent une amélioration du perceptron monocouche. Il comprend une ou plusieurs couches intermédiaires dites couches cachées. En effet le PMC est donc un réseau dont les neurones sont disposés en plusieurs couches successives et où chaque neurone d'une couche cachée est connecté à tous les neurones de la couche précédente et à chaque neurone de la couche suivante. Dans un réseau PMC, Il n'y a pas de connexions entre neurones d'une même couche, et le sens de parcourir des informations est de la couche d'entrée vers la couche de sortie sans retour en arrière. Les fonctions d'activation utilisées dans ce type de réseaux sont principalement les fonctions à seuil ou sigmoïdes. Le PMC suit un apprentissage supervisé par l'algorithme de rétro propagation du gradient.

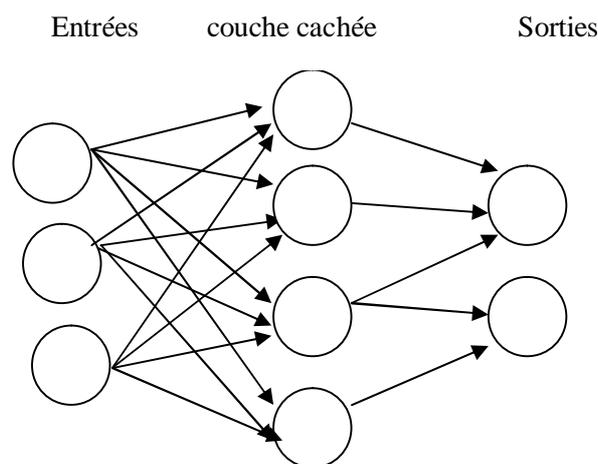


Figure 11 : Réseau de neurone à une couche cachée

V.4. Machines à Vecteurs de Support (SVM)

SVM est inspiré de la théorie statistique de l'apprentissage. Il est une méthode de classification binaire par apprentissage supervisé, elle fut introduite en 1995 [13].

Cette méthode est donc une alternative récente pour la classification. Elle repose sur l'existence d'un classificateur linéaire dans un espace approprié. Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonctions dites noyau (kernel) qui permettent une séparation optimale des données [13].

Dans la présentation des principes de fonctionnements, nous schématiserons les données par des « points » dans un plan.

Pour deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan.

Dans la figure 10, on détermine un hyperplan (H) qui sépare les deux ensembles de points.

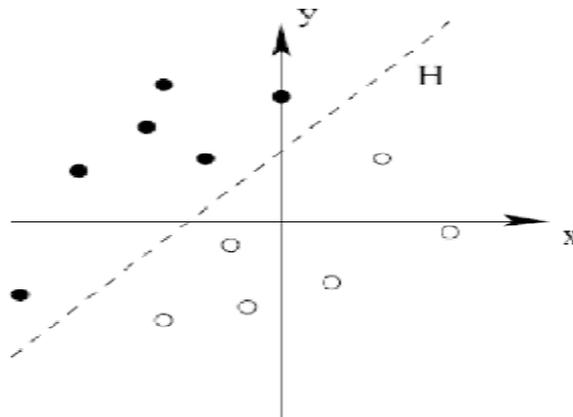


Figure 12 : hyperplan qui sépare les deux ensembles de points. [13]

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support [13].

Les SVM travaillent avec des données en dimension fixe, ils sont donc plus utilisés dans des applications restreintes à la reconnaissance de caractères ou de mots basée sur une segmentation explicite. L'inconvénient majeur de cette méthode est le coût important de mémorisation des vecteurs qui sont très nombreux pour absorber la variabilité de l'écriture [10].

VI. Les systèmes existants de la reconnaissance d'écriture arabes

Dans cette partie nous décrivons les caractéristiques de l'écriture arabes et les principaux systèmes existants pour la reconnaissance de caractères arabes.

VI.1. Caractéristiques générales de l'écriture arabe

L'arabe s'écrit de la droite vers la gauche. Elle est semi-cursive soit sous forme imprimée ou manuscrite. Chaque caractère a un point de connexion droite et/ou gauche lié sur la ligne de base. Le concept de majuscule et minuscule en écriture arabe n'existe pas [11].

Alphabet

L'alphabet arabe comporte 28 lettres (voir le tableau suivant). La forme des lettres dépend de leur position dans le mot. Certaines lettres prennent jusqu'à quatre formes différentes : par exemple le (ع ع ع ع)

Triquettes des lettres	isolé	début	milieu	fin	Triquettes des lettres	isolé	début	milieu	fin
Alif		ا		آ	Daal	ض	ضد	ضد	ضو
Baa	ب	ب	ب	ب	Thaa	ظ	ظ	ظ	ظ
Taa	ت	ت	ت	ت	Taa	ظ	ظ	ظ	ظ
Thaa	ث	ث	ث	ث	Ayn	ع	ع	ع	ع
Jim	ج	ج	ج	ج	Ghayn	غ	غ	غ	غ
Haa	ح	ح	ح	ح	Faa	ف	ف	ف	ف
Xaa	خ	خ	خ	خ	Qaaf	ق	ق	ق	ق
Daal		د		د	Kaaf	ك	ك	ك	ك
Thaal		ذ		ذ	Laam	ل	ل	ل	ل
Raa		ر		ر	Mim	م	م	م	م
Zayy		ز		ز	Nun	ن	ن	ن	ن
Seen	س	س	س	س	Hee	ه	ه	ه	ه
Shiin	ش	ش	ش	ش	Waa	و		و	
Saad	ص	ص	ص	ص	Yaa	ي	ي	ي	ي

Figure 13 : Les lettres arabes

Mais pour la plupart des lettres, les formes début/milieu et fin/isolé sont identiques à la ligature prés. La présence d'une ligature avec la lettre précédente ou avec la lettre suivante ne modifie pas la forme de la lettre de manière significative (pas plus que dans l'écriture manuscrite cursive latine) [11].

Les caractères d'un mot arabe peuvent être ligaturés horizontalement ou verticalement. (voir figure).

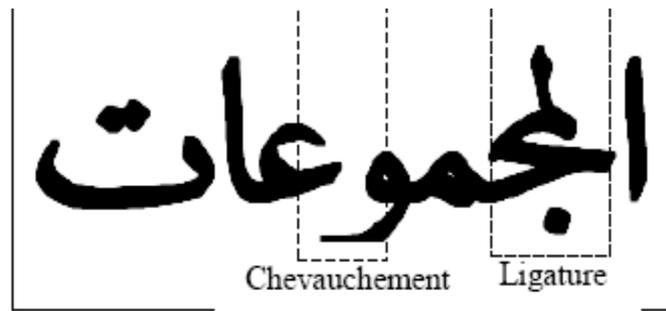


Figure 14 : Exemple de mot arabe présentant une ligature verticale et un chevauchement

Un mot arabe peut être composé d'une ou de plusieurs composantes connexes (pseudo-mot). Dans certaines fontes, on peut aller jusqu'à quatre caractères ligaturés verticalement.

Ceci rend la segmentation à priori en caractères quasi-impossible. Finalement, les chevauchements verticaux peuvent se produire par l'intersection des composantes connexes (pseudo-mots) ou des mots pour quelques combinaisons de caractères. Les chevauchements et ligatures dépendent de la fonte utilisée.

Signes diacritiques

Le terme « signe diacritique » peut porter à confusion : dans certains travaux, seules les voyelles arabes sont appelées diacritiques. Dans d'autres travaux, en revanche, tous les signes secondaires sont appelés diacritiques, qu'il s'agisse des voyelles, des points ou des autres signes (chadda, madda, hamza, ...).

Des points nécessaires pour différencier les lettres

Dans l'alphabet arabe, 15 lettres parmi les 28 possèdent un ou plusieurs points. Ces signes diacritiques sont situés soit au-dessus, soit en dessous de la forme à laquelle ils sont associés, mais jamais les deux à la fois.

Un groupe de deux points peut s'écrire sous forme d'une seule, ou de deux composantes connexes. Un groupe de trois points peut donner lieu à une, deux ou trois composantes connexes, en fonction du style d'écriture.

Les voyelles

En arabe, les voyelles ne sont pas des lettres, mais des signes diacritiques associés aux lettres sur lesquelles ils s'appliquent.

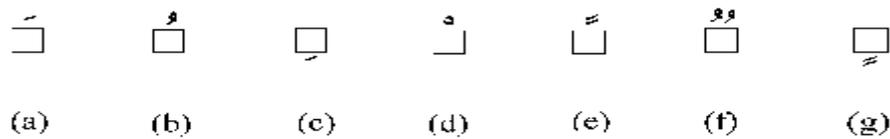


Figure 15: Voyelles en arabe

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

L'absence de voyelles peut toutefois être source de confusions. Un mot peut avoir plusieurs voyellations possibles et par conséquent plusieurs catégories grammaticales. Dans certains cas, une phrase peut donc avoir deux voyellations différentes, ce qui nous donne deux structures syntaxiques possibles.

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

VI.2. Les manuscrits arabes

VI.2.1. Définition d'un manuscrit

Un manuscrit (du latin *manus scriptus*) est un texte écrit à la main, il est d'abord un livre qui contient une ou plusieurs œuvres intéressantes par leur contenu. Leur copie, leur diffusion, leur appartenance à une bibliothèque ou à une institution, témoigne aussi de l'état des connaissances et de la circulation du savoir à une époque et en un lieu donnés. En outre, le manuscrit est un document archéologique, un objet qui peut être étudié sous l'angle de sa production artisanale, de ses caractéristiques matérielles et de son décor. Ceci ne concerne pas les seuls manuscrits arabes, mais tous les manuscrits quelle que soit l'aire culturelle à laquelle ils appartiennent [36].

VI.2.2. Les caractéristiques des manuscrits arabes

Les manuscrits arabes, comme les manuscrits de différentes langues, ont des caractéristiques communes mais chacun de ces manuscrits a sa propre identité. Parfois le même élément qui est en commun avec les autres diffère dans son contenu. Les manuscrits arabes diffèrent selon leurs régions (manuscrits orientaux ou maghrébins), l'appartenance religieuse de la communauté qui l'on a produits et la période de leur achèvement (création).

Les caractéristiques d'un manuscrit arabe peuvent se résumer dans ce qui suit:

- La présentation : tous les manuscrits arabes commencent au verso du premier feuillet, tandis que les rectos sont normalement réservés au « frontispice », à l'inscription du titre et du nom de l'auteur, et parfois au cachet, au médaillon ou au nom du commanditaire du livre, etc.
- Le début du texte peut être accompagné d'un décor particulier à la première page ou aux deux premières pages
- Le début du texte et aussi le début de chaque chapitre ou section du texte ainsi que le début de chaque sourate dans le Coran, est toujours précédé par le "*Basmala*" formule préliminaire conventionnelle (Au nom de Dieu, le Clément, le Miséricordieux).
- Le texte est normalement écrit avec de longues lignes, sauf dans le cas du texte poétique où on peut trouver deux colonnes dans la même page.
- L'encre utilisée : plusieurs couleurs sont utilisées dans l'écriture des manuscrits. Selon Paola Orsatti, « il existe des règles à suivre dans l'usage des encres de couleur ; le rouge par exemple est conseillé pour l'écriture des noms propres, des nombres, des citations, des termes techniques et pour le texte commenté dans les ouvrages d'exégèse ». Mais on trouve aussi des copies du Coran écrites avec de l'encre argentée et où le titre est doré. Les lignes sont parfois tracées à l'aide d'un instrument en couleur marron différente de l'encre du texte.
- Le texte encadré : en ce qui concerne le manuscrit coranique, le texte encadré apparaît au 8ème siècle, tandis que pour les textes non coraniques, l'apparition du cadre ne se trouve que plus tard. Le texte se présente normalement entouré par des cadres de différents styles soit ornementé, soit normal.

VI.3. Quelques systèmes existants de reconnaissance d'écriture arabe

Plusieurs chercheurs ont mené plusieurs travaux afin de proposer des systèmes de reconnaissance de caractères arabes. Dans ce qui suit nous présentons par ordre chronologique un récapitulatif précisant les caractéristiques et les performances de certains systèmes existants.

ICDAR 1997 [41] (Adnan Amin) : Il analyse différentes approches pour la reconnaissance automatique de l'écriture arabe. Il présente les approches On-Line/OffLine, ainsi que les approches Globale/Analytique. Il pose quelques unes des problématiques fondamentales de la reconnaissance de l'écriture arabe (alphabet, signes diacritiques, styles d'écriture, codage de l'écriture, pseudo-mots, ...).

CIFED 2000 [14] : Ils s'intéressent aux approches markoviennes qui ont été mises en œuvre pour résoudre le problème de la reconnaissance de l'écriture arabe. Concernant l'écriture arabe manuscrite, ils passent notamment en revue les HMM à une dimension, les HMM-planaires (PHMM).

Selon Ben Amara et al, les HMM-1D donnent des résultats encourageants, tout comme dans le cas de l'écriture latine. Les HMMs sont indépendants de l'alphabet considéré. Mais Ben Amara et al. soulignent également les limitations classiques d'une modélisation linéaire, qui là encore, ne sont pas propres à l'écriture arabe. Pour pallier ce problème, ils introduisent les PHMM, qui prennent en compte des variations bidirectionnelles de l'écriture. Les auteurs justifient en particulier ce choix sur l'arabe en raison du degré de difficulté supplémentaire que représente la présence de ligatures verticales. Ce dernier argument concernant la présence de ligatures verticales, n'est pas le plus pertinent.

Les symboles qui correspondent aux ligatures verticales peuvent en effet être reconnus comme tels dans leur globalité, sans qu'il soit nécessaire de les segmenter.

PAMI 2006 [42] : Le système de Liana M. Lorigo et Venu Govindaraju est le plus complet sur la reconnaissance de l'écriture arabe à l'heure actuelle. Ils passent en revue la plupart des problèmes spécifiques à la reconnaissance de l'écriture arabe et ils présentent quelques bases de données.

Ils découpent la chaîne de traitement d'un système de reconnaissance en fonctionnalités : représentation, segmentation, extraction de primitives, moteur de reconnaissance, où ils distinguent : réseau de Neurones, modèles de Markov Cachés et systèmes hybrides.

En conclusion, les auteurs expliquent que des systèmes de reconnaissance de l'écriture arabe donnent des performances satisfaisantes sur des applications contraintes (petite taille de lexique, forme des mots relativement contrainte...). L'avenir de la discipline se situe dans la capacité des systèmes à traiter de l'écriture libre, comme des courriers manuscrits. De telles applications nécessitent des modèles de langage développés, qui sont pour l'instant des voies largement inexplorées en langue arabe. L'intégration de contraintes morphologiques (analyse de la formation des mots à partir de suffixes et racines) permettrait également de reconnaître des mots hors-vocabulaire.

ISSPA 2007 [43]: Mohamed Cheriet fait une synthèse des différentes approches, avec la volonté d'ouvrir la réflexion sur de futures applications industrielles. Il part du constat suivant: les systèmes de reconnaissance de l'écriture manuscrite arabe, même s'ils obtiennent des performances encourageantes, n'ont pour l'instant fait leurs preuves que sur des données académiques, dans le cadre d'applications à l'environnement contraint. Il n'y a pas de système commercial de reconnaissance de l'écriture arabe manuscrite.

Il pose un certain nombre de questions ouvertes : Nécessité de procéder à une segmentation ? Quel paradigme de reconnaissance utiliser ? Quelle technique de reconnaissance utiliser ? Faire un post-traitement, ou essayer d'intégrer les contraintes au sein de la chaîne de reconnaissance, Redéfinir de nouvelles architectures, ou s'appuyer sur des techniques existantes ?

Il insiste également sur l'importance de l'analyse morphologique. Il s'interroge sur le meilleur endroit pour intégrer l'analyseur morphologique : au sein de la chaîne, ou à la fin en tant que post-traitement ?

Il s'interroge, également, sur les moyens de constituer un lexique pour guider un système de reconnaissance à vocabulaire ouvert, et conclut par l'intérêt d'utiliser le Coran en tant que corpus, pour en dériver un lexique de manière automatique ou semi-automatique.

Pour terminer, M. Cheriet conclut sur l'intérêt de travaux de recherche qui viseraient à mieux intégrer le traitement de la langue naturelle dans une chaîne de reconnaissance de l'écriture arabe.

Le tableau suivant présente un récapitulatif précisant les caractéristiques et les performances de certains systèmes de reconnaissance de caractère :

Référence	système	approche	segmentation	primitives	classification	performance
[Abdelazin 89]	Hors-ligne, imprimé	analytique	explicite	Structurelles/statistiques	structurelles/statistiques/ arbre de décision	RC 99%
[Abouhiba 93]	Hors-ligne, manuscrit	analytique	explicite	Structurelles	Transformation on-line/off-line	-
[Al-imami 90]	Hors-ligne, manuscrit	analytique	explicite	Structurelles	Arbre de décision	RM86-100%
[Aissaoui 94]	Hors-ligne, manuscrit	analytique	explicite	Structurelles	Structurelle	-
[Almouali m 87]	Hors-ligne, manuscrit	analytique	explicite	statistique	Syntaxique distance	RC 91%
[Algamal 01]	Hors-ligne, imprimé	analytique	explicite	-	Grammaire régulière	RC 93,4%
[El-Sheikh 88]	Hors-ligne, imprimé	analytique	explicite	Description de Fourier	Classifieur topologique	RC 99%
[Fehri 98]	Hors-ligne, manuscrit	analytique	explicite	Structurelles	Réseau de neurone/HMM	-

Etat de l'art

[Ghelies 99]	Hors-ligne, imprimé	analytique	explicite	Structurelles	Réseau de neurone	RC 89-93,1%
[Haj-Hassan 91]	Hors-ligne, imprimé	-	explicite	Structurelles	syntaxique	RC 99%
[Fehri 94]	Hors-ligne, imprimé	analytique	implicite	Structurelles	Réseau de neurone	RC 99%
[Hassibi 99]	Hors-ligne, imprimé	analytique	implicite	Structurelles	Réseau de neurone	RC 99%
[Jambi 93]	Hors-ligne, manuscrit	analytique	explicite	Structurelles	Dictionnaire	SC 95%
[Kurdy 93]	Hors-ligne, imprimé	analytique	explicite	Structurelles	morphologie	RC 98%
[Alaa 01]	Hors-ligne, manuscrit	analytique	explicite	Structurelles	Réseau de neurone	RC 69,72%
[Fahmy 01]	Hors-ligne, manuscrit	analytique	explicite	géométrique	Réseau de neurone	RC 69,7%
[Kandi 04]	Hors-ligne, imprimé	analytique	explicite	-	-	-

Figure 16 : Tableau récapitulatif précisant les caractéristiques et les performances de certains systèmes RC [14]

Etat de l'art

Référence	Système	Approche	Segmentation	Primitive	Classification	Performance
[Al- youseli 92]	Hors-ligne, manuscrit	Analytique	Explicite	Moment	Classifieur bayésien	RC 99 ,5%
[Ameur 93]	Hors-ligne, manuscrit	Analytique	Explicite	Structurelles	Arbre de décision	SC 98 ,9%
[Ameur 94]	Hors-ligne, manuscrit	Globale	-	Structurelles	Dictionnaire	-
[Amine 96]	Hors-ligne, caractère	-	-	Structurelles	Réseau de neurones	RC 90 ,92 %
[Amine 97]	Hors-ligne, mot	Globale	-	Structurelles	Réseau de neurones	RC 98 %
[Ben amara 95]	Hors-ligne	Analytique	Explicite	Géométrique	-	SC 93 ,98 %
[Ben amara 97]	Hors-ligne, caractère	-	-	Structurelles/ variables	Logique floue	RC 100 %
[El-dabi 90]	Hors-ligne, imprimé	Analytique	implicite	moment	Table de correspondance	RC 94%

Etat de l'art

Référence	Système	Approche	Segmentation	Primitive	Classification	Performance
[Mahjub 96]	Hors-ligne, caractère	-	-	Statistiques	HMMs	RC 98,1%
[Miled 96]	Hors-ligne, manuscrit	Analytique	Explicite	Structurelles	-	SC 98,52%
[Miled 98]	Hors-ligne, manuscrit	Analytique	Explicite	topologiques/ Statistiques	HMMs	RC 79,5%
[Mitiche 97]	Hors-ligne, imprimé	Analytique	Explicite	Structurelles	Distance	RC 90%
[Motawa 97]	Hors-ligne, manuscrit	Analytique	Explicite	Structurelles	Morphologies mathématique	SC 81,88%
[Souisi 97]	Hors-ligne, manuscrit	Analytique	Explicite	Statistiques/ Structurelles	Réseau de neurones	RC 76,17%
[Zahour 91]	Hors-ligne, manuscrit	Analytique	Explicite	Structurelles	Dictionnaire	RC 86%
[Zahour 98]	Hors-ligne, manuscrit	Analytique	Explicite	Structurelles	Mise en correspondance	RC 87%

Figure 17: Tableau récapitulatif précisant les caractéristiques et les performances de certains systèmes RC [14].

VI I. Conclusion

Dans ce chapitre, nous avons présenté les généralités et la description de l'architecture générale des systèmes de reconnaissance de caractères, puis nous avons décrit les différentes méthodes de reconnaissance de caractères (Markov Caché, réseau bayésien, réseau de neurone, machine à vecteur de support) et enfin, nous avons passé en revue les principaux travaux réalisés en reconnaissance de l'écriture arabe.

Dans le chapitre qui suit, nous allons décrire notre propre système de reconnaissance de noms arabes manuscrits.

Chapitre II
Conception

I. Introduction

Les manuscrits arabes anciens représentent un patrimoine culturel très précieux. Témoin de la civilisation arabe ce sont de véritables trésors pour les générations futures. Généralement, conservés dans des lieux vétustes, Ces documents se trouvent confrontés à une dégradation progressive, ils sont menacés d'un réel danger de disparition et d'effritement à cause du manque de moyens des lieux de conservation. La manipulation de ces œuvres constitue un danger supplémentaire de détérioration.

Par conséquent, la numérisation des manuscrits arabes anciens devient une démarche incontournable.

La mise en ligne des manuscrits numérisés permet d'accéder à ces ressources à partir de divers points. L'accès aux manuscrits numérisés ne saurait se faire sans un système d'indexation adéquat. Pour ce faire, le catalogage des manuscrits s'avère indispensable.

Le catalogage des manuscrits est une étape lente et fastidieuse à cause de son caractère manuel. Il faut, en moyenne, un à deux jours pour chaque manuscrit et parfois plus. Ainsi, l'automatisation de cette phase constitue une solution avantageuse. Elle permet d'extraire certaines informations clés, qui serviront de métadonnées (voir glossaire) utiles au catalogage. Parmi les métadonnées les plus pertinentes du catalogue, le nom de l'auteur ou du copiste se trouvent en bonne posture. Il est donc vivement recherché la reconnaissance des divers noms existant dans le manuscrit numérisé, ce qui nécessite la mise en œuvre d'un système de reconnaissance des noms arabes qui est l'objectif de notre travail.

Afin de d'atteindre cet objectif, nous proposons un système de reconnaissance des noms arabes manuscrits hors ligne, basé sur la méthode analytique et le modèle de Markov caché.

Dans ce présent chapitre, nous allons présenter l'architecture de notre système, la description générale et détaillée de ses diverses parties ainsi que leur fonctionnement et les différents algorithmes utilisés pour réaliser notre système.

Notre système consiste à la reconnaissance des noms des auteurs extraient à partir des manuscrits arabes, ceci nécessite l'utilisation d'une méthode plus performante pour la classification telle que la méthode de Markov caché.

II. Mise en œuvre de la méthode de Markov caché (HMM)

HMM est une méthode dominante dans le domaine de reconnaissance de l'écriture manuscrite. Son utilisation en reconnaissance des caractères arabes, signale des résultats encourageants à l'instar d'autres écritures (en l'occurrence latine) et ceci malgré la simplicité des architectures utilisées [22] [23].

Les différents travaux réalisés reposent pour une grande part sur l'expérience accumulée dans le domaine de la reconnaissance de l'écriture où les HMMs sont fréquemment utilisés. Comparés à d'autres approches de reconnaissance (structurelle, comparaison de chaînes,...), les HMMs se distinguent par leurs capacité de modéliser efficacement différentes sources de connaissance. En effet, d'une part ils offrent une intégration cohérente de différents niveaux de modélisation (morphologique, lexicale et syntaxique) et d'autre part, il existe des algorithmes puissants permettant de déterminer la valeur optimale des paramètres fournissant la meilleure adéquation entre le modèle et la base de données (connue) qualifiée d'apprentissage [14].

Les principaux avantages des HMMs sont attribués à leur cadre probabilistique, qui s'accorde bien avec la nature des signaux bruités comme le cas de la parole et de l'écriture manuscrite, et à leurs fondements théoriques qui ont permis la mise en œuvre d'algorithmes efficaces pour ajuster automatiquement et de façon itérative les paramètres du modèle [24][25][26][27].

Motivés par ces avantages, nous avons proposé une méthode analytique à base de HMMs pour la reconnaissance de noms arabes manuscrits dans un vocabulaire limité (voir glossaire). Les HMMs utilisés à cet effet, sont des HMMs discrets où les symboles des observations sont des symboles discrets dans un alphabet fini.

III. Architecture de système proposé

Dans ce qui suit nous présentons l'architecture ainsi la description générale et détaillée de notre système de reconnaissance des noms manuscrits arabes.

III.1. Description générale

Le système de reconnaissance de noms arabes manuscrits que nous avons développé se base sur la segmentation des noms en caractères, et sur le modèle de markov caché qui participe à l'apprentissage et la reconnaissance des caractères arabes manuscrits afin de reconnaître tous le nom.

En effet notre système fonctionne en deux phases : l'apprentissage et la reconnaissance. La vue globale de système est présentée sur la figure suivante.

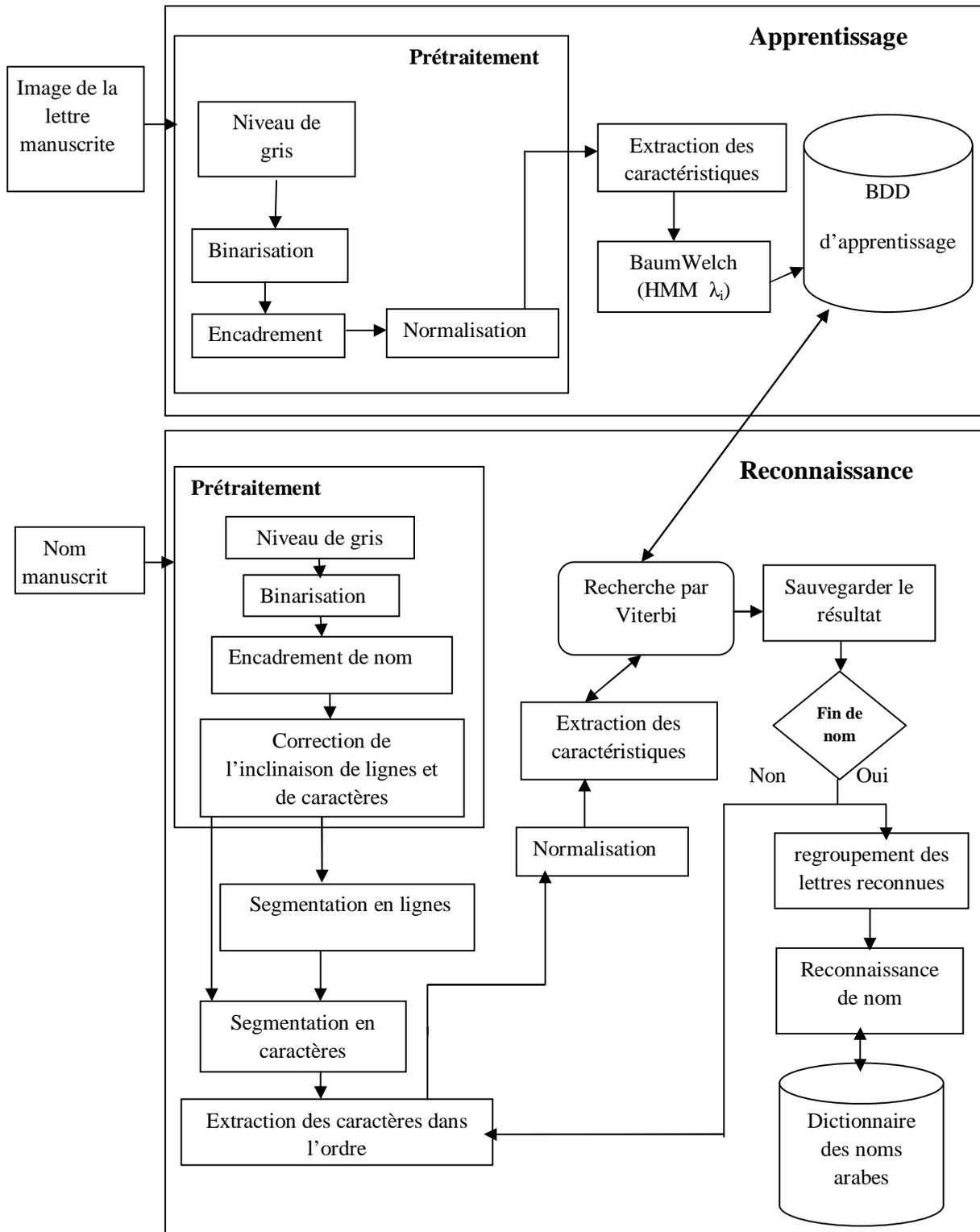


Figure 1: Système de reconnaissance des noms arabes manuscrits

Conception

L'objectif de notre travail est la reconnaissance des noms extraient à partir des manuscrits arabes. Ceci nécessite d'abord l'apprentissage des lettres.

Nous avons donc en entrée une image d'une lettre manuscrite qui correspond à une lettre de l'alphabet arabe, cette dernière est d'abord préparée en utilisant le processus de prétraitement qui comporte quatre sous modules : un sous module de niveaux de gris et de binarisation qui permet de passer d'une image en niveaux de gris à une image binaire composée de deux valeurs 0 et 1. La sortie de ce sous module sera l'entrée de sous module encadrement pour localiser la lettre et enfin nous allons passer l'image de la lettre à un sous module de normalisation afin d'obtenir une lettre adaptée à une dimension fixée par le système.

Toutefois, l'image de la lettre manuscrite prétraitée va passer à un sous système d'apprentissage pour qu'elle puisse être traitée, ce dernier se charge d'extraction des caractéristiques sous forme d'un vecteur de description, qui sera destiné à être traité par le modèle de Markov caché (Baum Welch). Les résultats obtenus seront sauvegardés dans une base de données d'apprentissage. Chaque lettre est représentée sur la base de données d'apprentissage sous quatre formes différentes (début, milieu, fin, isolée).

En ce qui concerne, le sous système de reconnaissance son but est la reconnaissance des noms manuscrits arabes :

En entrée une image de nom de l'auteur du manuscrit va passer au module de prétraitement comportant les mêmes modules cités ci-dessus ainsi un module de correction de l'inclinaison de lignes et de caractères, puis, nous effectuons la localisation de nom à partir de l'image résultante en utilisant l'algorithme d'encadrement. Une fois que le nom est localisé, il sera segmenté en lignes s'il se porte sur plusieurs lignes puis en caractères.

Nous traitons un caractère par itération de début jusqu'à la fin de nom. Chaque caractère sera normalisé pour l'adapter à une dimension précise, par la suite, nous allons construire son vecteur d'observation qui sera traité par l'algorithme de **Viterbi**.

Si le caractère est reconnu, il sera sauvegarder, sinon il sera rejeté. Dans les deux cas, si se n'est pas la fin du nom, le système passe au traitement de caractère suivant. Sinon, En fonction des lettres reconnues, le système sélectionne le nom correspondant dans un dictionnaire de noms arabes.

Nous allons dans ce qui suit, présenter en détail chaque composant et son processus de fonctionnement.

III.2. Description détaillée de notre système

Notre système prend en entrée les images des lettres manuscrites numérisées pour la phase d'apprentissage et les images des noms manuscrits numérisées concernant la phase de reconnaissance.

III.2.1. Sous système d'apprentissage

Le sous système d'apprentissage est la première phase dans notre système, qui a comme objectif la réalisation de la base d'apprentissage des différentes formes d'écriture des lettres manuscrites arabes afin de reconnaître les noms.

III.2.1.1. Prétraitement

Les prétraitements se donnent pour objectif la simplification de la procédure d'extraction de caractéristiques. Les prétraitements qui ont été appliqués sur les images numérisée sont: Niveau de gris, Binarisation, l'encadrement et la normalisation.

1. Niveau de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires.

La représentation des images en niveaux de gris, fait en attribuant à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise entre 0 et 255 [20].

Pour ce faire on peut utiliser la formule suivante :

$$\text{Gris} = \frac{\text{Rouge} + \text{Vert} + \text{Bleu}}{3} \quad (1)$$

Algorithme de niveau de gris

Ouvrir l'image I

Créer une image résultat de même taille que l'image I.

Pour chaque colonne faire

Pour chaque ligne faire

Extraire les canaux Rouge, Vert et Bleu du pixel de l'image I.

Calculer la valeur de niveau de gris correspondante (1).

Ecrire la valeur de l'entier RGB dans le pixel de l'image résultat.

Fin pour

Fin pour

2. Binarisation

La binarisation est effectuée dans le but de simuler le processus d'impression à l'encre noir sur du papier blanc. Elle est une opération qui produit deux classes de pixels, en général, ils sont représentés par des pixels noirs et des pixels blancs.

Ainsi les pixels correspondant à des points élevés doivent être binarisés en noire (valeur=0) et ceux dans les creux doivent être binarisés en blanc (valeur=1).

Il existe plusieurs algorithmes de binarisation. Nous avons utilisé l'algorithme de binarisation Seuillage globale fixe [28] :

C'est la technique la plus simple, elle consiste à comparer le niveau de gris de chaque pixel x_i de l'image avec un seuil global fixe T (par exemple 127). On note b_i la nouvelle valeur du pixel, le seuillage est donné par l'expression suivante :

$$b_i = 255 \text{ si } x_i \geq T \text{ et } b_i = 0 \text{ si } x_i < T.$$

Algorithme de binarisation

image1 : image d'entrée, Image2 : image de sortie

Pour i de 1 à largeur image1 Faire

Pour j de 1 à hauteur image1 Faire

pixel = Image1.pixel (i, j)

SI pixel < seuil ALORS

val=0

sinon

val=255

fin si

Image2.MettrePixel (i, j, val)

Fin pour

Fin pour

Fin.

Le niveau de gris de chaque pixel de l'image initiale est comparé à un seuil. S'il est inférieur à ce seuil, on met en noir le pixel qui lui correspond dans l'image résultat (noir et blanc), sinon il est mis en blanc.

La figure suivante montre la transformation d'une image originale à une image en niveau de gris puis en image binarisée.



Figure 2: (a) : Image originale, (b) : Image en niveau de gris, (c) : Image binarisée

3. L'encadrement

L'encadrement c'est le processus de localisation de la lettre, c'est de définir les coordonnées de la lettre dans l'image. Pour cela nous avons réalisé des algorithmes qui permettent de donner les propriétés suivantes : haut, bas, gauche, droite, afin de passer à l'encadrement de la lettre.

Conception

Nous présentons ci-dessous les algorithmes utilisés :

Initialisation haut = -1, gauche = -1; droit = -1; bas = -1;

Entier indexed, indiceb;

Image img;

```
// A gauche
```

```
Pour x = 0 à x < img.largeur faire
```

```
    Pour y = haut à y < img.hauteur faire
```

```
        // recuperer la couleur de chaque pixel
```

```
        Si ((img.getRGB(x, y) != blanc) et (img.getRGB(x, y) != -1)) Alors
```

```
            gauche=x;
```

```
            sortir de la boucle ;
```

```
        FinSi
```

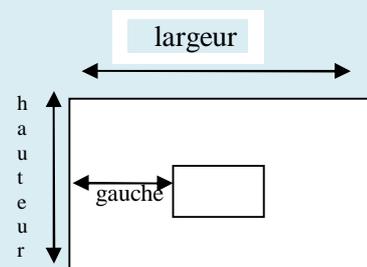
```
    Fin pour
```

```
    Si gauche!=-1 Alors
```

```
        Sortir de la boucle
```

```
    FinSi
```

```
Fin pour
```



// En hauteur

Pour $j = 0$ à $j < \text{img.hauteur}$ faire

 Pour $i = 0$ à $i < \text{img.largeur}$ faire

 // recuperer couleur de chaque pixel

 Si $(\text{img.getRGB}(i, j) \neq \text{blanc})$ et $(\text{img.getRGB}(i, j) \neq -1)$ Alors

haut = j ;

 Sortir de la boucle;

 Fin Si

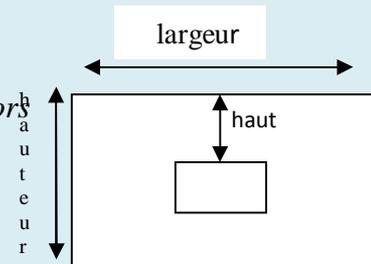
 Fin Pour

Si *haut* $\neq -1$ Alors

 Sortir de la boucle

FinSi

FinPour



// A Droite

Pour $x = \text{img.largeur} - 1$; $x \geq \text{gauche}$, $x++$ faire

 Pour $y = \text{img.getHeight}() - 1$; $y \geq \text{haut}$; $y--$ faire

 // recuperer la couleur de chaque pixel

 Si $(\text{img.getRGB}(x, y) \neq \text{blanc})$ et $(\text{img.getRGB}(x, y) \neq -1)$ Alors

Indiced = x ;

droit = $\text{img.largeur} - x$;

 Sortir de la boucle

 Fin Si

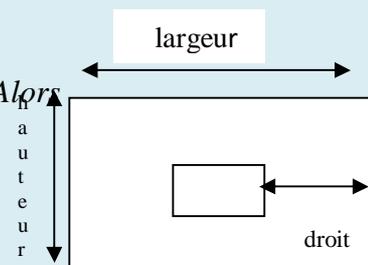
 Fin Pour

Si (*droit* $\neq -1$) Alors

 Sortir de la boucle

Fin Si

Fin Pour.



//En bas

Pour $y = \text{img.hauteur} - 1$ à $y > \text{haut}$ Faire

 Pour $x = 0$ à $x < \text{img.largeur}$ Faire

 // recuperer couleur de chaque pixel

 Si $(\text{img.getRGB}(x, y) \neq \text{blanc})$ et $(\text{img.getRGB}(x, y) \neq -1)$

 Indiceb=y ;

 bas=img.hauteur-y;

 sortir de la boucle;

 Fin Si

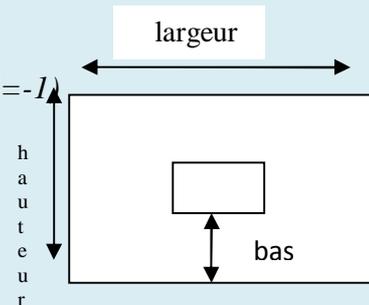
 Fin Pour

 Si bas!= -1 Alors

 Sortir de la boucle

 Fin Si

Fin Pour



//L'encadrement

Entiers $w=0, h=0$;

Pour col = gauche à col < indiced faire

 Pour row = haut à row < indiceb faire

 // Recopie de l'image 1 dans l' image résultant

 Incrémentation de h;

 Fin Pour

 Incrémentation de w;

 h=0;

Fin Pour.

Ces algorithmes permettent donc de localiser les pixels noirs dans une image contenant seulement une lettre ou un nom en parcourant toute l'image. Donc cette phase est vraiment la plus intéressante dans notre sous système de prétraitement, puisqu'elle nous offre un gain de temps que ce soit dans l'apprentissage ou dans la reconnaissance du nom et ceci par le traitement de l'image encadrée résultante.

La figure suivante montre un exemple de l'encadrement d'une lettre dans une image.

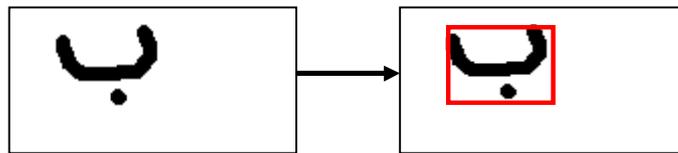


Figure 3 : Exemple d'encadrement d'une lettre dans une image

4. Normalisation

La normalisation consiste à transformer la taille de l'image et l'adapter à une dimension fixée a priori par le réalisateur de système. Pour cela nous avons proposés une procédure qui permet de normaliser l'image encadrée dans une dimension de 64x64 pixels. Cette procédure copie le contenu d'une première image pixel par pixel et la copie dans une seconde image rétrécie si l'image dépasse 64x64, sinon elle sera agrandie.

Nous avons utilisé une échelle qui sera calculée automatiquement en fonction de la dimension de la lettre encadrée dans l'image par rapport à la dimension 64*64 pixels.

Alors, on note les différentes échelles : Echelle < 1 : Pour agrandir,

Echelle > 1 : Pour rétrécir,

Echelle = 1 : Pour garder la même taille,

L'algorithme de normalisation

Procédure Normalisation (Image1 : Bitmap, Image2 : Bitmap, Echelle : Réel): Bitmap

Variable X, Y: Réel ;

c : entier ;

i, j : Réel ;

Hauteur, Largeur : Réel;

Début

Hauteur = bas - haut

Largeur = droit - gauche

EchelleH = Hauteur / Image2 . hauteur

EchelleW = Largeur / Image2 . largeur

j = 0 ;

Pour y de haut à bas (pas de EchelleH) Faire

i = 0

Pour x de droit à gauche (pas de EchelleW) Faire

c = Image1. Point (x, y)

Image2 . PSet (i, j) , c

i = i + 1

Fin Pour

j = j + 1 ;

Fin Pour

Fi

III.2.1.2. L'extraction des primitives

Après les prétraitements sur l'image de la lettre manuscrite, nous nous intéressons à l'étape de sa caractérisation (extraction des primitives), qui a pour objectif de déterminer le vecteur représentatif de cette image. Ce vecteur servira à représenter de manière compacte la séquence des formes (la lettre) contenues dans l'image.

Dans notre système, nous extrayons plutôt des primitives de type statistiques qui conviennent pour le modèle de Markov caché (MMC). A partir de l'image normalisée, nous tirons des primitives du type **zoning** [35] [38] [39].

La méthode zoning consiste à partitionner une image en M sous-images de mêmes tailles appelées zones (8x8). Nous calculons pour chaque zone le pourcentage de pixels noir en obtenant une matrice de pourcentage qui représente le vecteur de primitives, celui-ci sera utilisé pour l'apprentissage et la reconnaissance.

La figure suivante illustre le principe de zoning pour la lettre arabe (ba) :

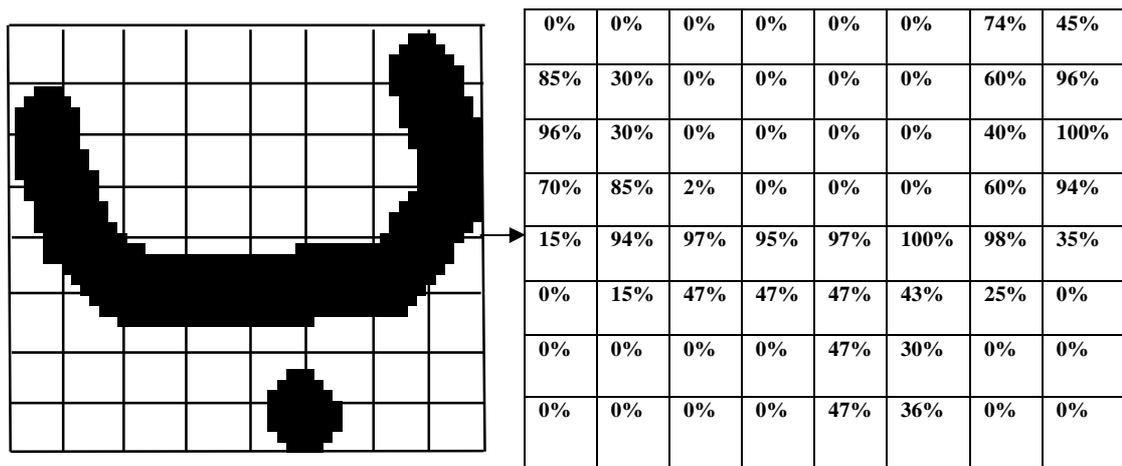


Figure 4 : Représentation de la lettre arabe (ba) sous forme d'une matrice

III.2.1.3. Apprentissage

Nous avons élaboré un MMC (Π, A, B) pour chaque lettre de l'alphabet arabe à l'aide de la matrice résultante de la phase d'extraction de primitive.

Nous avons fixé le nombre d'états à $N=4$, selon les travaux réalisés [44] [12] [45].

Formellement, la base d'apprentissage (BDD) est un doublet (L, λ) où :

- ü L présente la lettre arabe à modéliser par un MMC,
- ü λ présente l'ensemble des paramètres d'un MMC (A, B, Π).

Où :

A , présente la matrice de distribution de probabilité de transition d'état .

B , présente la matrice de distribution de probabilité de symbole d'observation .

Π , présente la distribution de probabilité d'état initiale .

Chaque lettre arabe peut s'écrire de 2 à 4 formes différentes : au *début du mot*, au *milieu du mot*, à la *fin du mot mais liée à une lettre*, à la *fin du mot sans être liée à la lettre*. Selon la position de la lettre dans le mot, nous modélisons chaque lettre par un MMC différent.

Il existe des algorithmes itératifs pour réaliser cette opération d'apprentissage tel que :

- ü Les algorithmes de sélection de modèles qui convergent vers une structure optimale à partir d'une structure initiale par une suite de fusion d'état [30].
- ü L'algorithme de Baum Welch qui évalue les probabilités des états en fonction des paramètres courants (Π, A, B) et la séquence de vecteurs d'observation. Il permet la maximisation qui consiste à réestimer les paramètres (Π, A, B) et produit en sortie un HMM (Π, A, B) optimisé. Voir la figure suivante

Dans notre système nous avons utilisé l'algorithme de Baum Welch face a sa simplicité, sa moins complexité et il garantit la vraisemblance des données [30].



Figure 5 : Schémas représentatif de données d'apprentissage

III.2.1.3.1. Algorithme de Baum-Welch

Étant donné une suite d'observations $Y_{1:T}$, comment définir un MMC (au travers de ses paramètres) maximisant la probabilité d'observation de $Y_{1:T}$ (i.e., $p(Y_{1:T} | \lambda)$).

L'algorithme de Baum-Welch est défini par [37] :

1. Initialisation à λ_0 des paramètres du MMC.
2. Calculer un nouveau jeu de paramètres λ à partir de λ_0 .
3. Si $\log(p(Y_{1:T} | \lambda)) - \log(p(Y_{1:T} | \lambda_0)) < \sigma$, arrêt des itérations.
4. Sinon, le nouveau jeu de paramètres λ prend la place de λ_0 et on recommence à l'étape 2.

Afin d'effectuer cette procédure, il est nécessaire de définir le moyen de réestimer, à partir d'un jeu de paramètres λ_0 , les paramètres λ du MMC [31].

Pour cela, nous définissons tout d'abord $\epsilon_t(i, j)$ comme la probabilité de se trouver dans l'état i au temps t et dans l'état j au temps $t+1$ conditionnellement aux observations.

Il apparaît immédiat, à partir des définitions des variables Forward $\alpha_t(i)$ et Backward $\beta_{t+1}(j)$ (voir glossaire), que $\epsilon_t(i, j)$ peut se calculer, comme suit:

$$\begin{aligned} \epsilon_t(i, j) &= \frac{\alpha_t(i) a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)}{p(y_{1:T} | \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^M \sum_{j=1}^M \alpha_t(i) a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)} \end{aligned}$$

On utilise ensuite, $\gamma_t(i)$, la possibilité d'être à l'état i au temps t , donné par :

$$\gamma_t(i) = \sum_{j=1}^M \epsilon_t(i, j).$$

Tel que M présente le nombre de symboles d'observation;

$\sum_{t=1}^T \gamma_t(i)$: Correspond au nombre de fois où le processus est dans l'état i , et

$\sum_{t=1}^{T-1} \epsilon_t(i, j)$: Est le nombre de transitions de l'état i à l'état j au travers des T périodes.

On peut alors définir les règles de réestimation des paramètres du $\bar{\lambda}=(\bar{A}, \bar{B}, \bar{\Pi})$ MMC par :

$$\begin{aligned} - \bar{\pi}_i &= \gamma_1(i), \\ - \bar{a}_{ij} &= \frac{\sum_{t=1}^{T-1} \epsilon_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \\ - \bar{b}_j(k) &= \bar{b}_{jk} = \frac{\sum_{t, s_t=k} \gamma_t(j)}{\sum_t \gamma_t(j)}. \end{aligned}$$

III.2.1.4. Base de données du notre système

La base de données utilisée est construite à partir des lettres arabes imprimées ainsi que ses paramètres de HMM (Π, A, B). Chaque lettre doit être représentée dans la base sous ces différentes formes (début, milieu, fin, isolée) comme dans le tableau suivant :

Conception

lettres	MMC λ						
أ	λ_1	ذ	λ_{26}	ط	λ_{51}	ك	λ_{75}
آ	λ_2	ذ	λ_{27}	ظ	λ_{52}	ل	λ_{76}
ب	λ_3	ر	λ_{28}	ظ	λ_{53}	ل	λ_{77}
ب	λ_4	ر	λ_{29}	ت	λ_{54}	ل	λ_{78}
ب	λ_5	ز	λ_{30}	ظ	λ_{55}	ل	λ_{79}
ب	λ_6	ز	λ_{31}	ع	λ_{56}	م	λ_{80}
ت	λ_7	س	λ_{32}	ع	λ_{57}	م	λ_{81}
ت	λ_8	س	λ_{33}	ع	λ_{58}	م	λ_{82}
ث	λ_9	س	λ_{34}	ع	λ_{59}	م	λ_{83}
ث	λ_{10}	س	λ_{35}	غ	λ_{60}	ن	λ_{84}
ج	λ_{11}	ش	λ_{36}	غ	λ_{61}	ن	λ_{85}
ج	λ_{12}	ش	λ_{37}	غ	λ_{62}	ن	λ_{86}
ج	λ_{13}	ش	λ_{38}	غ	λ_{63}	ه	λ_{87}
ج	λ_{14}	ش	λ_{39}	ف	λ_{64}	ه	λ_{88}
ح	λ_{15}	ص	λ_{40}	ف	λ_{65}	ه	λ_{89}
ح	λ_{16}	ص	λ_{41}	ف	λ_{66}	ه	λ_{90}
ح	λ_{17}	ص	λ_{42}	ف	λ_{67}	و	λ_{91}
ح	λ_{18}	ص	λ_{43}	ق	λ_{68}	و	λ_{92}
خ	λ_{19}	ض	λ_{44}	ق	λ_{69}	ي	λ_{93}
خ	λ_{20}	ض	λ_{45}	ق	λ_{70}	ي	λ_{94}
خ	λ_{21}	ض	λ_{46}	ق	λ_{71}	ي	λ_{95}
خ	λ_{22}	ض	λ_{47}	ك	λ_{72}	ي	λ_{96}
د	λ_{23}	ط	λ_{48}	ك	λ_{73}	ن	λ_{97}
د	λ_{24}	ط	λ_{49}	ك	λ_{74}	ن	λ_{98}
ث	λ_{25}	ث	λ_{50}				

III.2.2. Le sous-système de reconnaissance

Le sous système de reconnaissance du nom manuscrit consiste à reconnaître chaque lettre de nom qui est représentée dans une matrice d'observation. Cette reconnaissance se fait par référence aux modèles de la base de données d'apprentissage par le biais de **Viterbi**.

La reconnaissance de toutes les lettres aboutie à la reconnaissance du nom lui-même. Toutes les différentes parties de la reconnaissance seront détaillées dans ce qui suit.

III.2.2.1. Prétraitement

Le module de prétraitement dans la phase de reconnaissance des noms contient en plus de celui de la phase d'apprentissage, la correction de l'inclinaison des lignes et des caractères.

Ces opérations sont nécessaires puisque l'inclinaison des lignes et l'inclinaison des caractères sont des sources d'erreur classique, relativement gênantes pour la phase de segmentation donc pour la reconnaissance.

1. La correction de l'inclinaison des lignes

La correction de l'inclinaison des lignes de texte, également appelée correction de "skew", consiste à redresser horizontalement les lignes d'écriture obliques. Plusieurs méthodes sont disponibles. Les deux plus populaires sont la transformée de Hough (appliquée sur les centres de gravité des composantes connexes), et les histogrammes (voir glossaire) [35] de projection.

Nous avons utilisé la méthode des histogrammes qui consiste à parcourir l'image selon des directions d proches de l'horizontale, et à compter le nombre de pixels noirs selon ces directions pour chaque ligne. La qualité de l'histogramme, ou sa pertinence, est estimée par son entropie. La direction la plus probable est celle qui maximise l'entropie. L'histogramme d'entropie maximale est celui dont les extrema sont les plus marqués, voir figure qui suit [3].

L'angle du document (ou de la ligne), θ , est celui qui correspond à l'histogramme d'entropie maximale.

Pour corriger cette inclinaison, il suffit d'appliquer une rotation de l'image d'angle θ :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

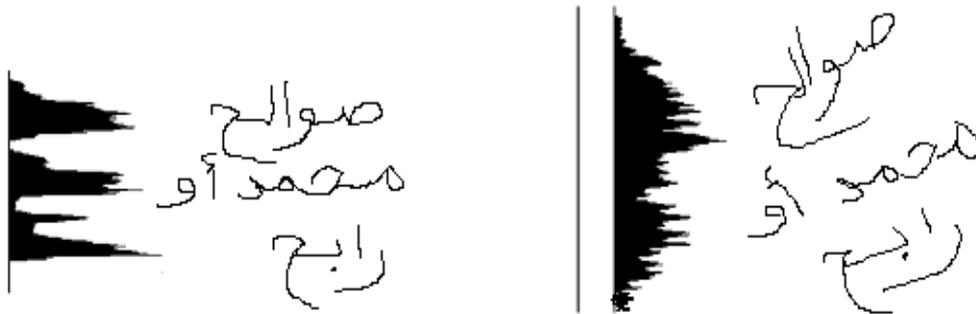


Figure 6 : Correction de l'inclinaison des lignes

2. La correction de l'inclinaison des caractères

La correction de l'inclinaison des caractères de texte, également appelée correction de "slant", consiste à redresser verticalement l'écriture obliques (inclinées vers la droite ou vers la gauche). Plusieurs méthodes sont disponibles pour estimer l'inclinaison des caractères.

Nous avons utilisé des histogrammes de projection, de la même manière que pour la correction de l'inclinaison des lignes, mais cette fois dans des directions proches de la verticale. La encore, l'histogramme d'entropie maximale est celui dont les extrema sont les plus marqués. [33]

III.2.2.2. La segmentation

La segmentation permet à partir d'une image acquise (l'image du nom) l'extraction des lignes (segmentation en lignes), ensuite à partir de ces lignes sont extraits les caractères. Cette étape est indispensable dans notre système, puisqu'elle produit en sortie des caractères qui sont la base pour la reconnaissance des noms.

III.2.2.2.1. La segmentation en lignes

La segmentation en ligne consiste à extraire les lignes à partir d'une image du nom acquise. Cette phase est nécessaire dans le cas où le nom est sur plusieurs lignes.

Le principe de notre algorithme est d'utiliser les lignes blanches (les espaces inter-lignes) pour distinguer et extraire les régions (les lignes) contenant des pixels noir (les parties de nom).

Notre algorithme est applicable dans le cas où les lignes ne sont pas chevauchées.

La figure suivante montre un exemple de segmentation en lignes :

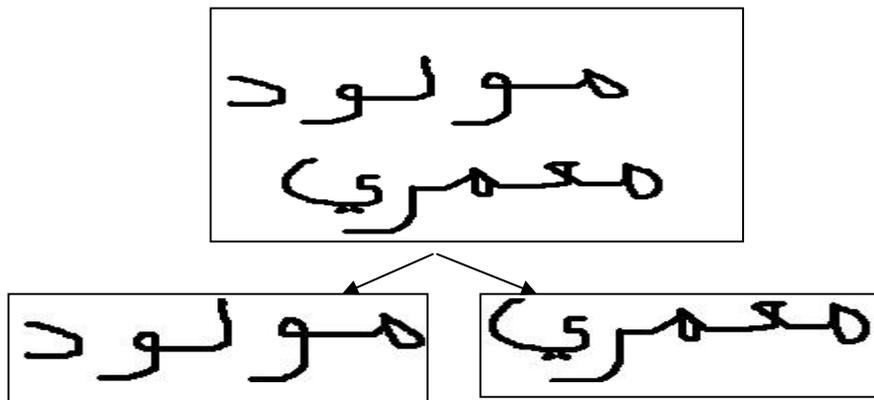


Figure 7: exemple de segmentation en lignes

III.2.2.2.2. La segmentation en caractères

Le but de la segmentation en caractères est de déterminer (approximativement) où débute et où prend fin chaque caractère dans un nom.

Vu que les caractères des noms arabes sont généralement liés par des éléments d'épaisseur uniforme (voir la figure suivante), nous avons proposé un algorithme de segmentation qui permet la séparation des caractères en éliminant ces éléments.

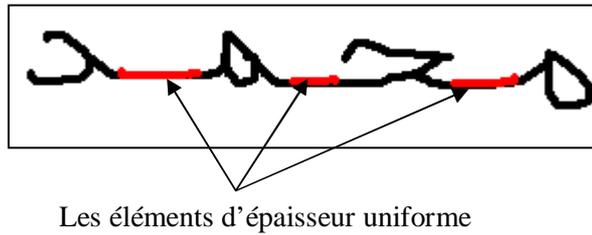


Figure 8: Représentation des éléments d'épaisseur uniforme.

1. Principe de fonctionnement de l'algorithme de segmentation en caractères

Nous avons en entrée une image contenant le nom ou une partie de nom, cette image sera parcourue de droite à gauche en détectant les colonnes blanches en utilisant un seuil. Telle que si le nombre de pixel blanc dans cette colonne est supérieur ou égale à ce seuil, elle sera considérée une colonne blanche. Ce seuil est obtenu empiriquement comme suit : hauteur de l'image * 0.95f.

Dans chaque itération, on calcul aussi le nombre de colonnes blanches successives. Si ce nombre est supérieur ou égale à un seuil, cette suite de colonne est considérée comme un bloque blanc, et ses indices de début et de fin seront sauvegarder dans une liste « al ». Ce seuil est obtenu empiriquement comme suit : maximum (1, hauteur * 0.05f).

A l'aide de cette liste, nous déterminons les caractères à extraire de l'image d'entrée en parcourant la liste par pas de deux (2).

Dans ce qui suit, nous présentons les détails de cet algorithme.

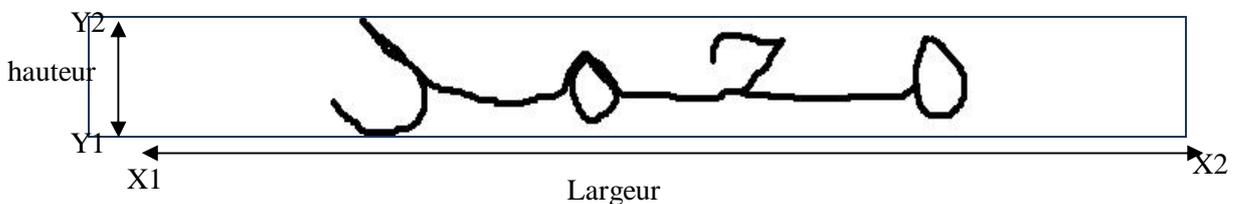


Figure 9: Image d'entrée résultante de la segmentation en ligne

Conception

Entrée : l'image contenant le nom ou une partie de nom,

Sortie : images des trames résultantes de la segmentation de nom,

// hauteur de l'image d'entrée

Entier : hauteur = $y_2 - y_1$,

//seuil déterminant si un bloque peut être négligé (blanc)

Entier : SeuilBlocBlanc = maximum (1, hauteur * 0.05f),

//seuil déterminant si une colonne d'un pixel d'épaisseur peut être négligée (blanche)

Entier : SeuilColonneBlanc = hauteur * 0.95f ,

//Liste résultante contenant les indices où se débute et où se termine chaque trame à extraire dans l'image d'entrée

Liste d'entier : al,

Booleen : Separateur = **Vrais**,

Entier : CharX1 = 0, prevCharX1 = -1,

Booleen: EspaceBlancLibre = **Faux**,

Entier : NumBlancConsecutive = 0,

Etape 1 : Le but de cette étape est de trouver les indices des bloques considérés blancs, puis les sauvegarder dans la liste al.

```
Pour x = 1 à x < (largeur - 1), pas de 1 Faire
| // Teste effectuer dans le cas il a trouvé un bloque de largeur supérieur à la hauteur et il n'a
|   pas marqué des colonnes blanches.
| Si EspaceBlancLibre = Faux et NumBlancConsecutive == 0 et x - charX1 >= hauteur alors
|   | x = charX1,
|   | EspaceBlancLibre = Vrais,
|   FinSi
|
| //Nombre de pixels blancs dans une colonne
| Entier : numPixelsBlancColumn = 0,
| Booléen : EspaceBlanc = Vrais,
|
| Pour y = y1 à y < y2, pas de 1 Faire
|   | //tester si le pixel de coordonnée x, y est blanc ou non
|   | Si pixels(x, y) >= 128
|   |   | //Incrémenter le nombre de pixel blanc d'une colonne x
|   |   | Incrimenter numPixelsBlancColumn,
|   |   FinSi
|   Sinon
|     | // le cas où le pixel de coordonnée x, y est noir (<128)
|     | Si EspaceBlancLibre = Faux Alors
|     |   | EspaceBlanc = Faux
|     |   | Sortir de la boucle pour,
|     |   FinSi
|     FinSinon,
|   Fin pour,
```

```
//tester si la colonne x est négligeable (> SeuilColonneBlanc) ou non (<SeuilColonneBlanc)
Si EspaceBlancLibre = Vrais et numPixelsBlancColumn < SeuilColonneBlanc Alors
    EspaceBlanc = Faux
FinSi

Si EspaceBlanc = Vrais Alors
    //incrémenter le nombre de colonne blanche successive
    Incrimenter NumBlancConsecutive,
    //test si le bloque (succession de colonnes) est négligeable (>= SeuilBlocBlanc) ou non
    (<SeuilBlocBlanc)
    Si NumBlancConsecutive >= SeuilBlocBlanc)
        Si Separateur= Faux Alors
            Separateur=Vrais,
            //sauvegarder dans la liste al les indices des bloques blancs à supprimer
            Ajouter dans al(CharX1), //début de bloque
            Ajouter dans al(x- (NumBlancConsecutive - 1)), // fin de bloque
        FinSi
    FinSi
FinSi

// Le cas où la colonne x n'est pas blanche
Sinon
    NumBlancConsecutive = 0,
    Si Separateur = Vrais Alors
        Separateur = Faux,
        prevCharX1 = charX1,
        charX1 = x,
        EspaceBlancLibre = Faux,
    FinSi
FinSinon
Fin Pour
```

//à la fin de nom, si le dernier bloque n'est pas un blanc alors il faut sauvegarder son indice pour le localiser

Si NumBlancConsecutive = 0 Alors

 Ajouter dans al(CharX1),

 Ajouter dans al (largeur),

FinSi

La figure suivante illustre le principe de l'étape 1 :

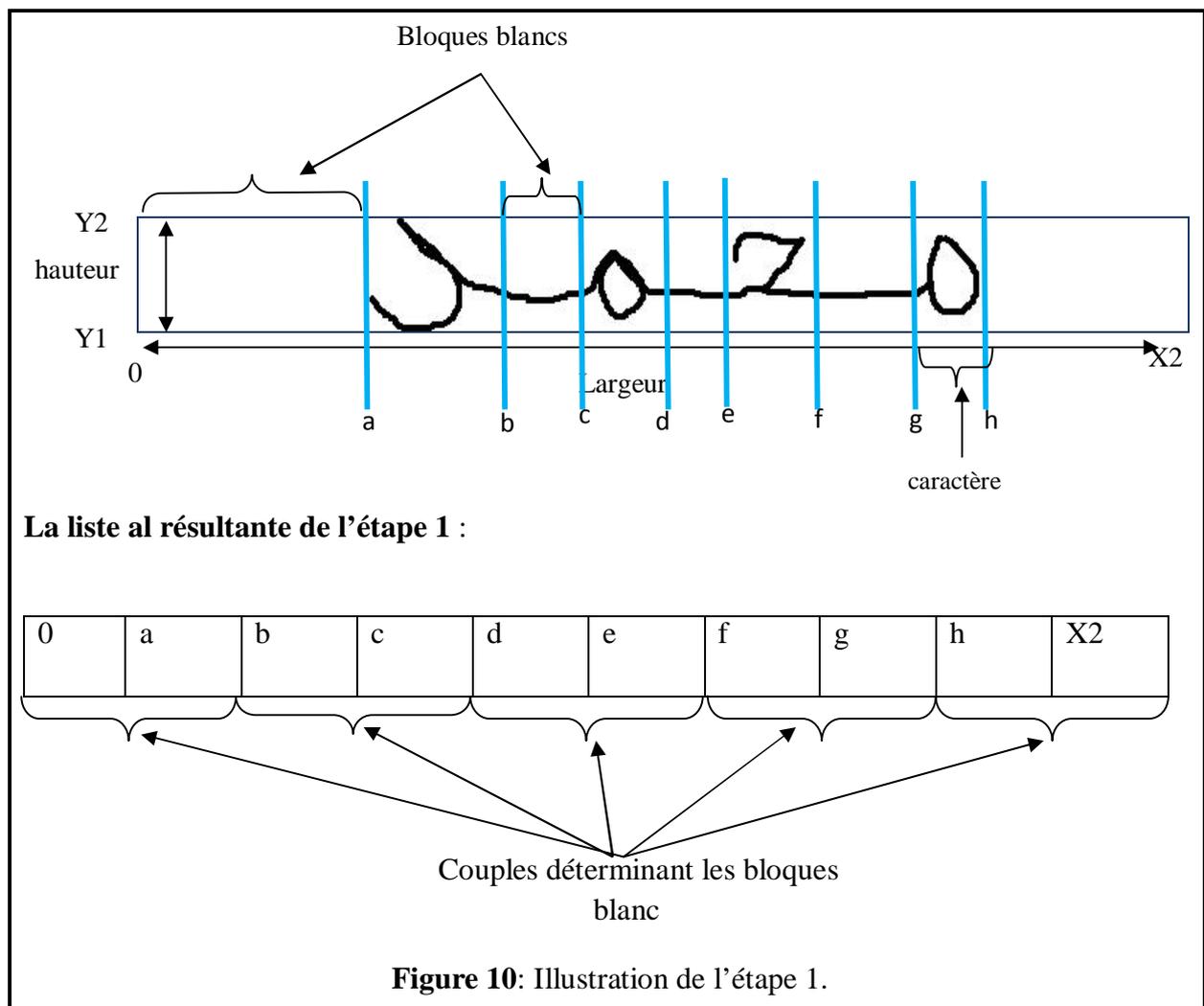


Figure 10: Illustration de l'étape 1.

Étape 2: l'objectif est de fusionner les blocs blanc successifs les plus proche en utilisant le seuil `MinlargeurChar`.

Entier : `MinlargeurChar = hauteur * 0.15f`,

Si `MinlargeurChar < 1` Alors

`MinlargeurChar=1`,

FinSi

//Test permettant de déterminer si deux blocs blanc successifs dans la liste peuvent être fusionnés

Pour `i = 0` à `(i + 4) < taille de la liste al`, pas de 2 Faire

 Entier `LargeurChar = récupérer de al l'élément (i + 2) - récupérer de al l'élément (i)`,

 Si `largeurChar < MinlargeurChar` ou `largeurChar < 2`)

 Supprimer de la liste al l'élément `(i + 2)`,

 Supprimer de la liste al l'élément `(i + 1)`;

 Décrémenter `i` de 2,

 FinSi

FinPour

Étape 3 : Après l'obtention de la liste `al` optimisée qui contient les indices des blocs blancs, elle sera utilisée pour l'extraction des caractères de l'image d'entrée. Cette liste est utilisée comme suit :

La liste sera parcourue de début jusqu'à la fin par pas de deux, dans chaque itération le contenu de la liste dont les indices sont `(i+1)` et `(i+2)` délimite où commence et où termine le caractère à extraire dans l'image d'entrée.

Si par exemple `i=2` alors le contenu des indices 3 et 4 délimite une trame à extraire.

Comme il est illustré dans la figure suivante :

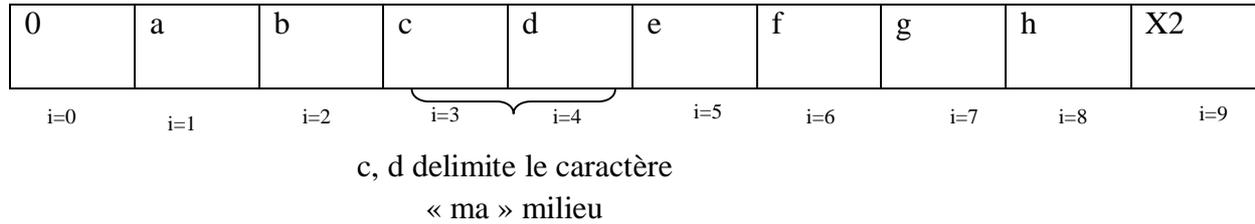


Figure 11: Illustration de l'étape 3

Pour $i = 0$ à $i < (\text{taille de la liste} - 2)$, pas de 2 Faire

Si $\text{al}(0) = 0$ Alors

Entier cx1 = récupérer de la liste al l'élément $(i + 1)$,

Entier cx2 = récupérer de la liste al l'élément $(i + 2)$,

Extraire de l'image d'entrée trame de largeur $(\text{cx1}, \text{cx2})$,

FinSi

Sinon

Si $\text{al}(0) \neq 0$ Alors

Entier cx1 = récupérer de la liste al l'élément (i) ,

Extraire de l'image d'entrée trame de largeur $(0, \text{cx1})$,

FinSi

Si $i \neq 0$ Alors

Entier cx1 = récupérer de la liste al l'élément $(i + 1)$,

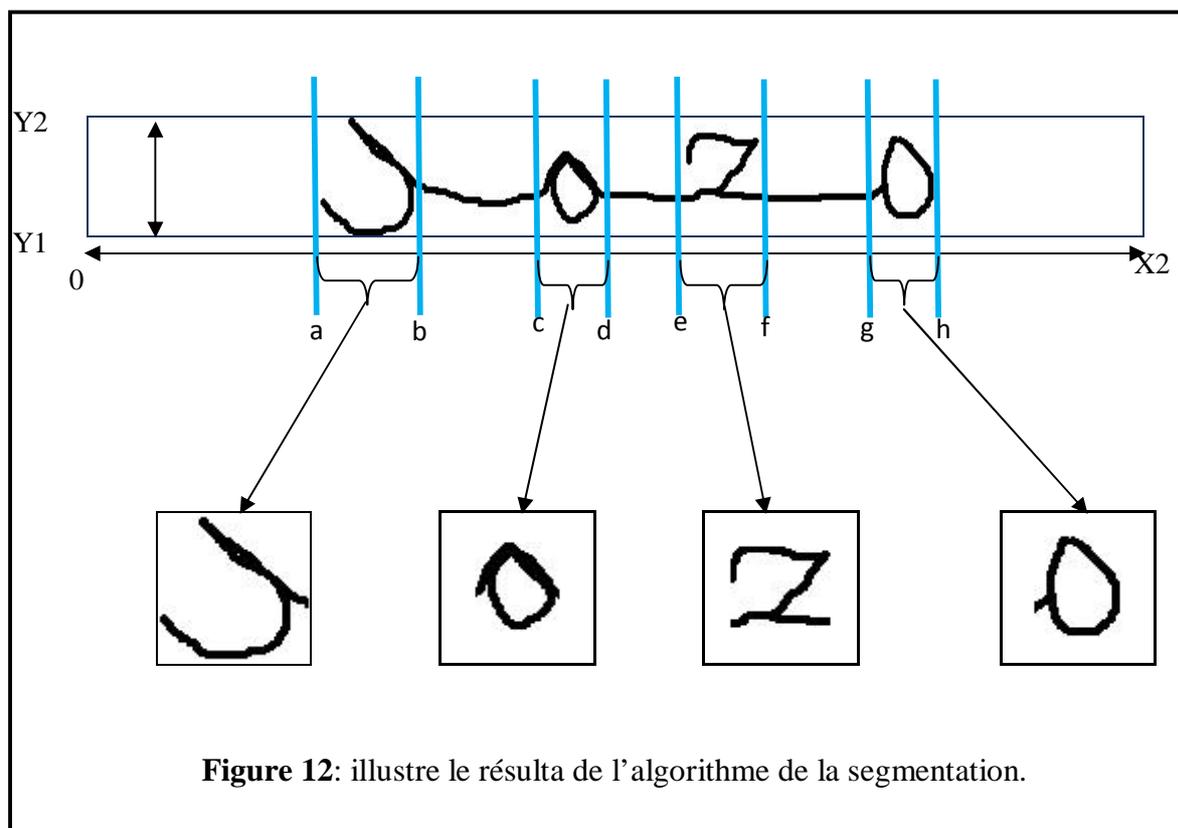
Entier cx2 = récupérer de la liste al l'élément $(i + 2)$,

Extraire de l'image d'entrée trame de largeur $(\text{cx1}, \text{cx2})$,

FinSi

FinSinon

Fin Pour



L'application de l'algorithme de segmentation en caractères sur l'écriture arabe manuscrite peut ne pas être fiable, vu les caractéristiques de l'écriture arabe manuscrites par exemples : les problèmes de discontinuité de l'écriture, les ligatures, de chevauchement et d'accolement de pseudo mots, de grandes variabilité inter et intra scripteur, de variation de dimension des pseudos mots ainsi que les signes diacritiques...

« Il n'est pas évident de juger de l'efficacité d'un algorithme de segmentation en lettres, le résultat peut être décevant » [14].

III.2.2.3. Normalisation et extraction des caractéristiques

Après la segmentation, nous faisons une opération de normalisation de chaque caractère pour l'adapter à une dimension de 64*64 pixels, puis nous allons construire son vecteur de primitives qui va être traité par Viterbi pour trouver la lettre correspondante dans la base de données d'apprentissage.

III.2.2.4. La reconnaissance par Viterbi

Après les phases de normalisation et l'extraction de vecteur d'observation de la lettre, ce dernier sera donné en entrée de l'algorithme de Viterbi pour être comparé aux modèles λ_k de la base de données réalisée dans la phase d'apprentissage, $k=1, n$. Le modèle retenu sera celui qui fournira la meilleure probabilité correspondante à l'évolution de sa suite de primitives, c'est-à-dire :

$$V = \arg \max [P (O / \lambda_k)], 1 \leq V \leq n \quad [29]$$

Où

O : présente l'observation qui correspond à la lettre.

λ_k : présente les paramètres de MMC (Π_k, A_k, B_k).

Le principe de Viterbi est présenté dans la figure qui suit :

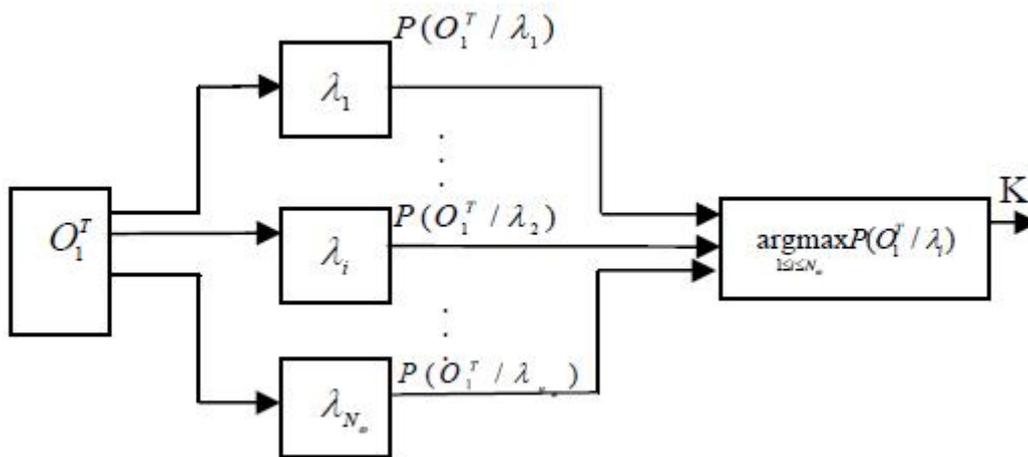


Figure 13 : Illustration de principe de Viterbi.

L'algorithme de Viterbi que nous avons utilisé pour la reconnaissance des noms arabe se trouve allégé, en effet il n'y a pas une suite d'états emprunté pour arriver à la probabilité maximale. La découverte du modèle donnant la plus forte probabilité $P (O / \lambda)$ est suffisante pour identifier la lettre.

De plus, sachant que les matrices de MMC A et B sont connues à l'avance, on peut optimiser l'algorithme de reconnaissance en les transformant (A, B) en valeur logarithmique.

L'algorithme de reconnaissance est le suivant [29]:

Initialisation :

$$\delta_1(i) = \log(\pi_i) + \log(b_i(O_1)) \quad 1 \leq i \leq N,$$

Induction :

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) + \log(a_{ij})] + \log(b_j(O_t)) \quad 1 \leq j \leq N, \quad 2 \leq t \leq T$$

Terminaison :

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

L'algorithme de Viterbi peut retourner deux résultats : soit la lettre est reconnue dans ce cas elle sera sauvegardée, soit la lettre est rejetée. Dans les deux cas, le système peut prendre l'une des décisions que nous définissons dans ce qui suit:

1. Si ce n'est pas la fin du nom : dans ce cas, le système recommence à partir de processus d'acquisition d'une nouvelle lettre tout en éliminant les lettres parcourues puisque nous avons abouti à une reconnaissance.
2. Si c'est la fin du nom : dans ce cas, le système regroupe tous les lettres reconnues pour aboutir à la reconnaissance du nom.

III.2.2.5. Regroupement des lettres et la reconnaissance du nom

Le système regroupe toutes les lettres reconnues, en fonction de ces lettres il sélectionne le nom correspondant dans le dictionnaire des noms arabes. La sélection s'effectue en choisissant le nom possédant le maximum des lettres reconnues.

IV. Conclusion

Ce chapitre nous a permis de présenter et de décrire le fonctionnement de notre système de reconnaissance des noms manuscrits. Nous avons d'abord introduit la description générale de système puis la description détaillée ainsi que les différents algorithmes appliqués dans chaque partie de ce système. Dans le chapitre qui suit nous allons présenter les différents outils qui nous a permis de réaliser notre système de reconnaissance et de décrire la réalisation du prototype que nous avons développé pour mettre en œuvre notre travail.

Chapitre III

Réalisation

I. Introduction

Après avoir défini l'architecture de système de reconnaissance de noms manuscrits arabe que nous avons développé, nous présentons dans ce chapitre sa réalisation. D'abord nous décrivons les outils de développements, puis nous décrivons les différentes interfaces et le fonctionnement de système ainsi que les testes effectués afin d'évaluer notre système.

II. Les outils de développement

II.1. Système d'exploitations

Notre application a été développée sous le système d'exploitation Windows XP, mais comme elle est développée en langage java, elle peut être intégrée dans n'importe quel autre système d'exploitation supportant la machine virtuelle java (Windows 98/00, Linux, ...).

II.2. Langage de programmation

JAVA est un langage purement orienté objet. Les concepts de JAVA sont hérités du C++ dont il est issu. Java possède des bibliothèques de classes qui fournissent les types de données de base, les possibilités d'entrées/sorties du système, la gestion des réseaux et autres fonctions utilitaires. Ces bibliothèques sont écrites en JAVA et de plus JAVA est doté de mécanismes puissants tels que :

- Ø **Le Garbage Collector** (GC) asynchrone (ramasse-miettes), Le GC libère toutes les instances en mémoire qui ne sont plus référencées par aucune variable.
- Ø **Un mécanisme d'exception** contraignant dans lequel une exception ne peut pas être ignorée. Une méthode provoquant une exception doit soit la traiter, soit indiquer explicitement qu'elle ne la traite pas et la fait remonter. Une exception non captée par aucune méthode provoque l'arrêt de la JAVA Virtual Machine (JVM).
- Ø **Le multi-Thread** : Toute instance peut être gérée dans un nouveau Thread (tache), Les accès concurrents sont gérés par des moniteurs transparents pour le programmeur.
- Ø La **sécurité** est assurée par trois couches :
 - a) Le "*verifier*" est intégré à la JVM et vérifie que le pseudo code qu'il reçoit est valide (il vérifie par exemple qu'un entier n'est pas utilisé comme une adresse mémoire).

b) Le "*classloader*" vérifie que le chargement des classes est valide (il vérifie par exemple qu'une classe ne se fait pas passer pour une autre ou qu'une classe possède une signature valide).

c) Le "*security manager*" contrôle l'accès d'un programme aux ressources (entrées/sorties, accès réseaux, chargement de bibliothèques binaires, etc.).

Ø La portabilité de JAVA :

La compilation JAVA ne prend pas en compte les caractéristiques systèmes de la machine sur laquelle elle est effectuée (on parle de pseudo code). L'interpréteur JAVA interprète alors le pseudo code et exécute les applications. Ceci fait de JAVA un langage totalement indépendant des plates-formes et donc portable d'un système à l'autre. L'inconvénient du pseudo code est sa faible vitesse d'exécution.

II.3. Le langage SQL (Structured Query Language)

C'est un langage standard (langage standardisé en 1992) qui permet communiquer avec la plupart des SGBD en utilisant la même syntaxe. Toutefois, chaque SGBD possède sa propre interface permettant d'établir une connexion.

II.4. IDE Eclipse

Eclipse est un projet open source fondé par SUN Microsystems, l'IDE Eclipse est un environnement de développement permettant d'écrire, compiler, déboguer et déployer des programmes. Il est écrit en java, et y'a un grand nombre de modules pour étendre l'IDE Eclipse.

Pour le développement de notre application, nous avons choisi Eclipse Galileo (Son interface principale est donnée dans la (figure qui suit). De part ces qualités en tant que langage évolué, Eclipse permet un traitement simple et efficace sur des fichiers images en fournissant des API spécialisées qui sont APIs standards et JAI (Java Advanced Imaging). Elles sont fournies dans des package comme java.awt et javax.imageio etc.

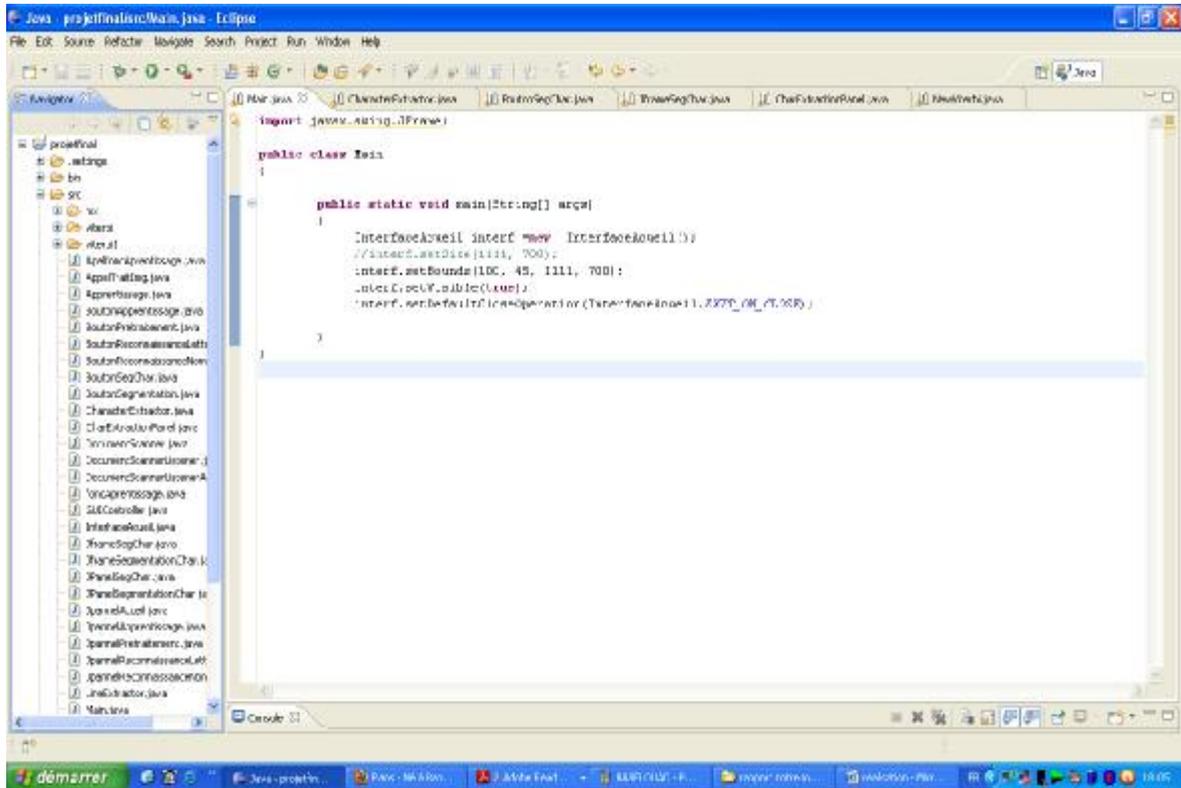


Figure 1 : Interface JAVA Eclipse.

II.5. Le SGBD MYSQL

MYSQL est un véritable serveur de base de donnée SQL (Structured Query Language) qui est un langage de requêtes vers les bases de données exploitant le modèle relationnel. Il en reprend la syntaxe mais n'en conserve pas toute la puissance puisque de nombreuses fonctionnalités de SQL n'apparaissent pas dans MYSQL (sélection imbriquées, clés étrangères...).

III. Présentation de notre système

Le système de reconnaissance des noms arabes manuscrits que nous avons développé est composé de deux sous systèmes, un sous système d'apprentissage et un sous système de reconnaissance.

Dans ce qui suit nous présentons les interfaces et le fonctionnement de notre système.

III.1. Les interfaces de notre système

III.1.1. Interface d'accueil

C'est la première page visualisée par l'utilisateur de notre système. Cette page contient six boutons essentiels :

- Ü Bouton prétraitement : permettant d'afficher l'interface de prétraitement.
- Ü Bouton apprentissage : permettant d'afficher l'interface d'apprentissage.
- Ü Bouton reconnaissance de la lettre: permettant d'afficher l'interface de reconnaissance de la lettre ;
- Ü Bouton segmentation en lignes : permettant d'afficher l'interface de segmentation en lignes ;
- Ü Bouton segmentation en caractères : permettant d'afficher l'interface de segmentation en caractères ;
- Ü Bouton reconnaissance de nom : permettant d'afficher l'interface de reconnaissance de nom.

La figure suivante représente l'interface d'accueil de notre système:



Figure2 : interface d'accueil

III.1. 2. Interface de prétraitement

Elle permet d'effectuer les prétraitements sur l'image de la lettre et l'extraction de primitives (matrice d'observations). Elle contient deux boutons :

- Ü Bouton sélection : il permet le chargement de l'image.
- Ü Bouton lancement de prétraitement : il permet de déclencher les opérations de prétraitement et l'extraction de primitives.

Le résultat de l'extraction de primitives sera affiché dans la zone de texte, et l'image prétraitée sera affichée sur écran.

La figure suivante présente l'interface de prétraitement :

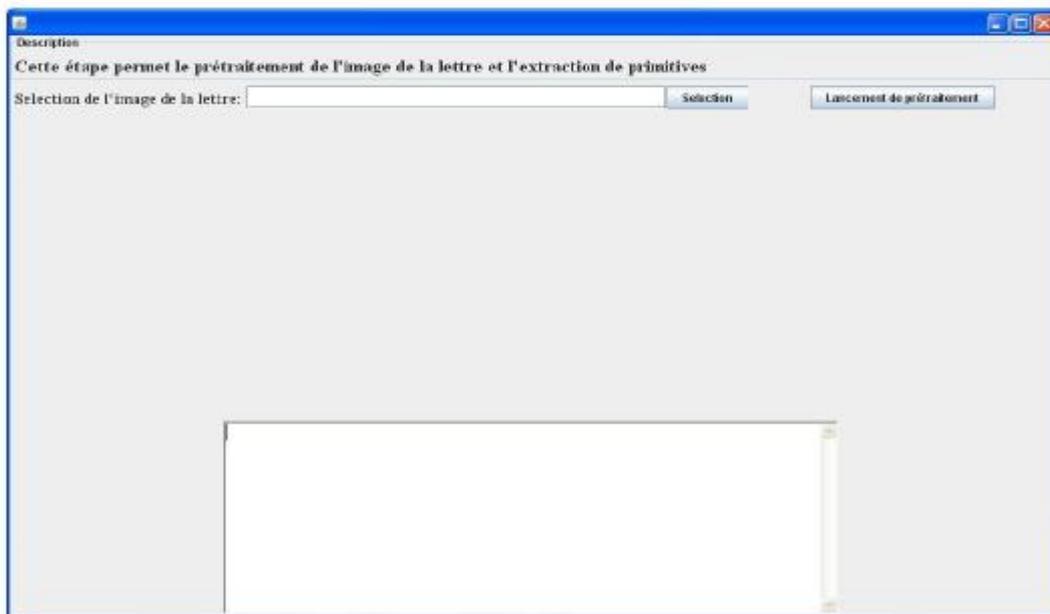


Figure 3 : Interface de prétraitement.

III.1. 3. Interface d'apprentissage

Cette interface permet l'apprentissage de l'image de la lettre prétraitée en appliquant l'algorithme de Baum Weltch. Elle contient :

- Ü Bouton sélection : il permet le chargement de l'image.
- Ü Bouton d'apprentissage: il permet de déclencher l'opération d'apprentissage.
- Ü Listes de choix: qui permet le choix de type de la lettre.

Les résultats de l'apprentissage seront affichés dans la zone de texte.

La figure suivante montre l'interface d'apprentissage :

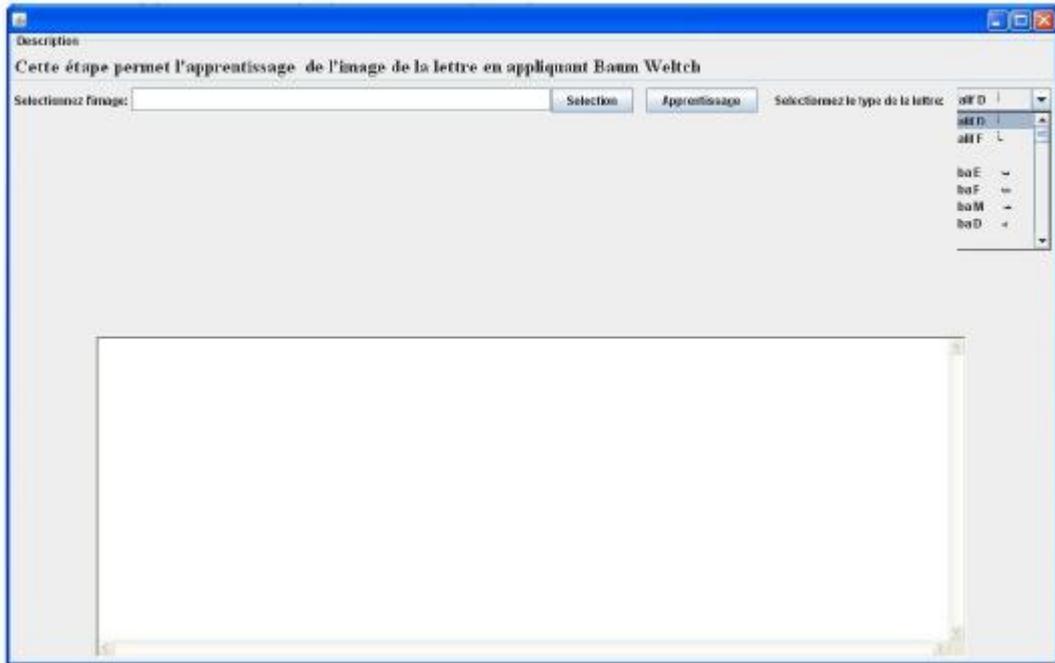


Figure 4: interface d'apprentissage

III.1.4. Interface reconnaissance de la lettre

Cette interface permet d'effectuer la reconnaissance de la lettre en appliquant l'algorithme de Viterbi. Elle contient :

- Bouton sélection : il permet le chargement de l'image de la lettre.
- Bouton reconnaissance : il permet de déclencher l'opération de reconnaissance de la lettre.

Les résultats de la reconnaissance de la lettre seront affichés dans la zone de texte.

La figure suivante présente l'interface de la reconnaissance de la lettre :

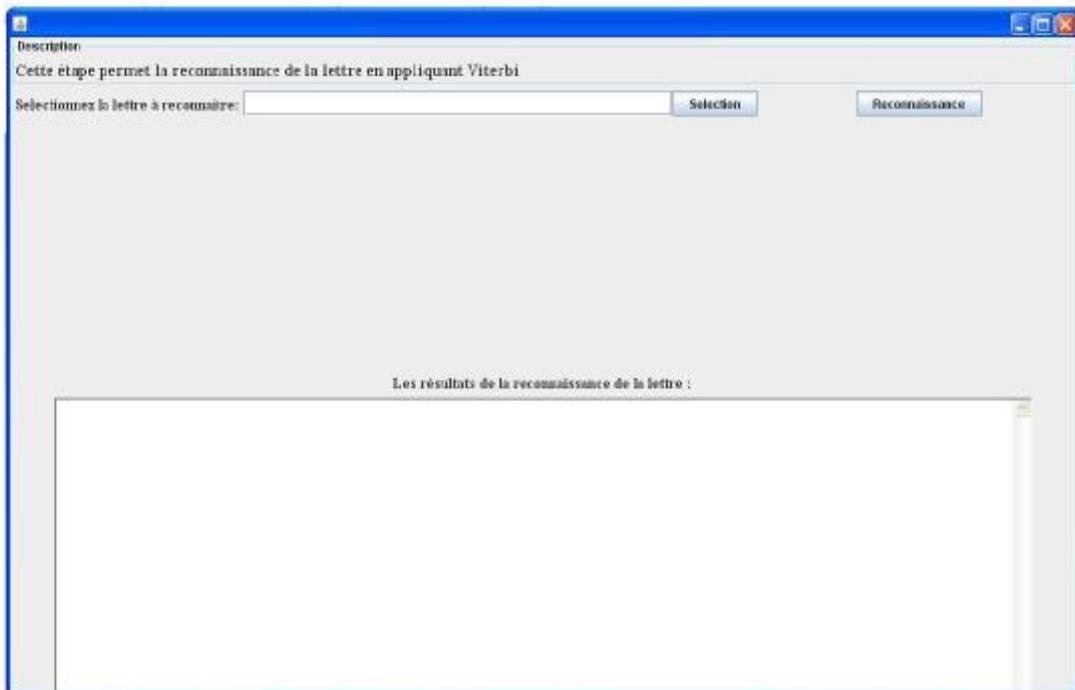


Figure 5 : interface de reconnaissance.

III.1.5. Interface de segmentation en lignes

Cette étape permet la segmentation en lignes de l'image de nom. Elle contient :

- Ü Bouton sélection de l'image d'entrée : il permet le chargement de l'image de nom à segmenter.
- Ü Bouton extraire les lignes : il permet de déclencher l'opération de segmentation en lignes.
- Ü Bouton sélection de répertoire de sortie : il permet de sélectionner le répertoire où les résultats de segmentation en lignes seront enregistrés.

La figure qui suit présente l'interface de segmentation en lignes :

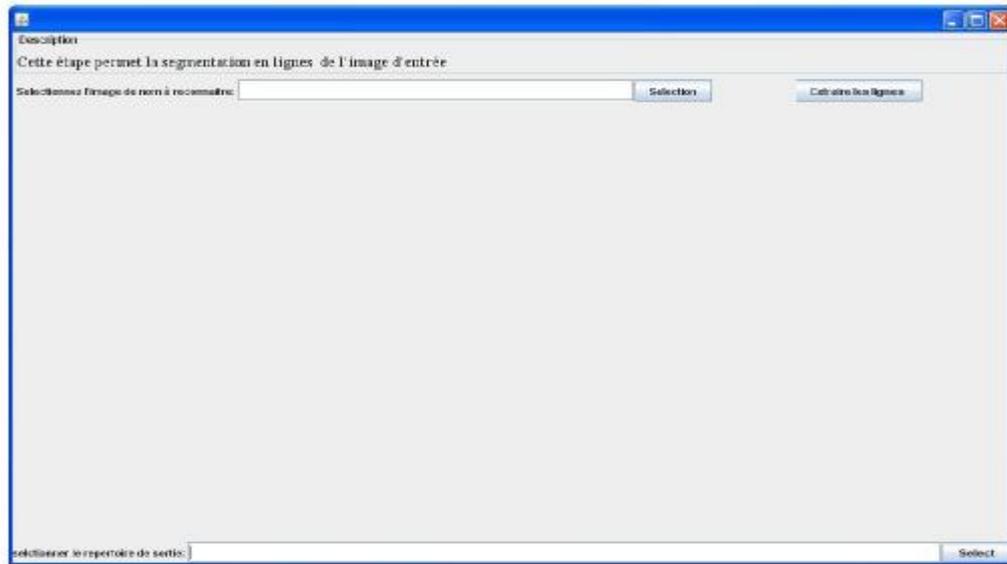


Figure 6 : interface de segmentation en lignes.

III.1.6. Interface de reconnaissance de nom

Cette interface permet d'effectuer la reconnaissance de nom. Elle contient :

- ü Bouton sélection : il permet le chargement de l'image de nom à reconnaître.
- ü Bouton reconnaissance de nom : il permet de déclencher l'opération de reconnaissance de nom.

Le résultat sera affiché dans la zone de texte.

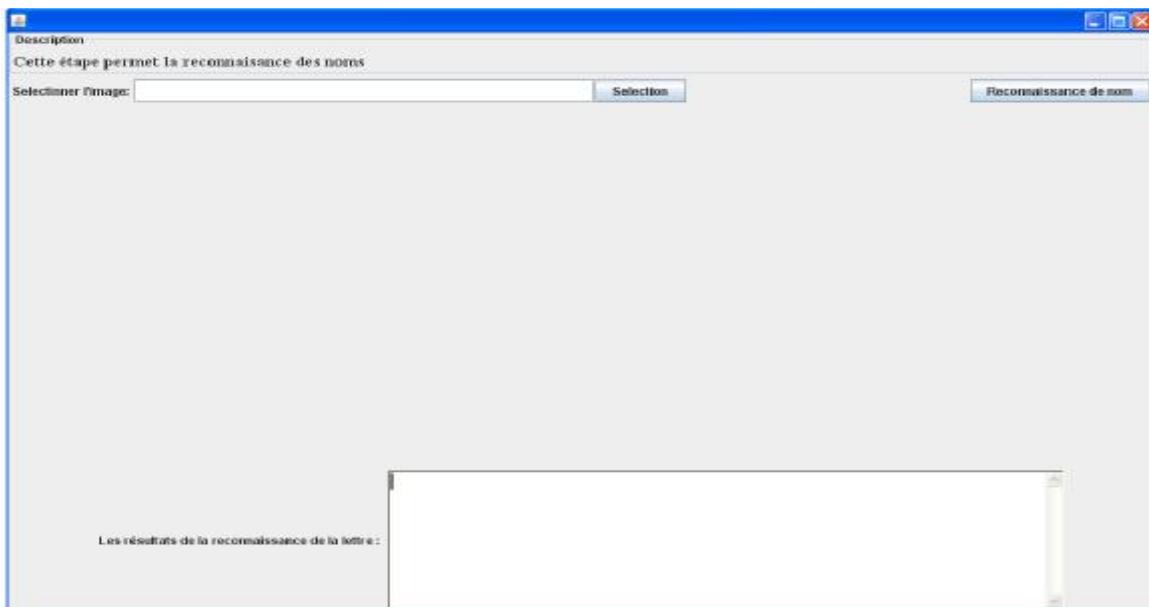


Figure 7 : interface de reconnaissance de nom.

III.2. Le fonctionnement de système

Dans ce qui suit, nous présentons le déroulement d'un exemple sur notre système afin de comprendre son fonctionnement.

III.2.1. Sous système d'apprentissage

Il s'agit lors de cette étape d'apprendre au système les propriétés pertinentes du vocabulaire (les différentes lettres arabes) et de l'organiser en modèle de référence (base de données d'apprentissage). Chaque lettre est représentée dans la base de données selon sa position dans le nom : début, milieu, fin ou isolée.

Avant l'apprentissage d'une lettre, elle doit d'abord passer par l'étape de prétraitement et l'extraction de primitives. Dans ce qui suit nous présentons un teste sur la lettre « alif ».

- a. Chargement de l'image : Il consiste à charger l'image de la lettre à prétraiter.

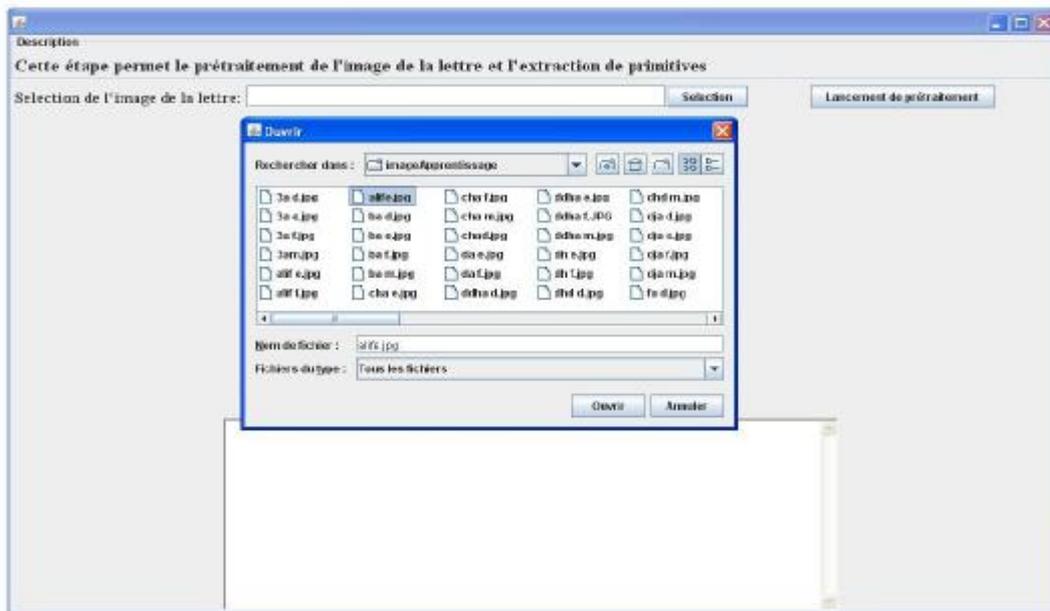


Figure 8 : chargement de l'image de la lettre « alif ».

- b. Prétraitement et l'extraction de primitives : ces opérations sont déclenchées en cliquant sur le bouton de « lancement de prétraitement », les résultats de cette étape sont l'image prétraitée affichée à l'écran et la matrice d'observations de la lettre affichée dans la zone de texte. Ceci est monté dans la figure suivante :

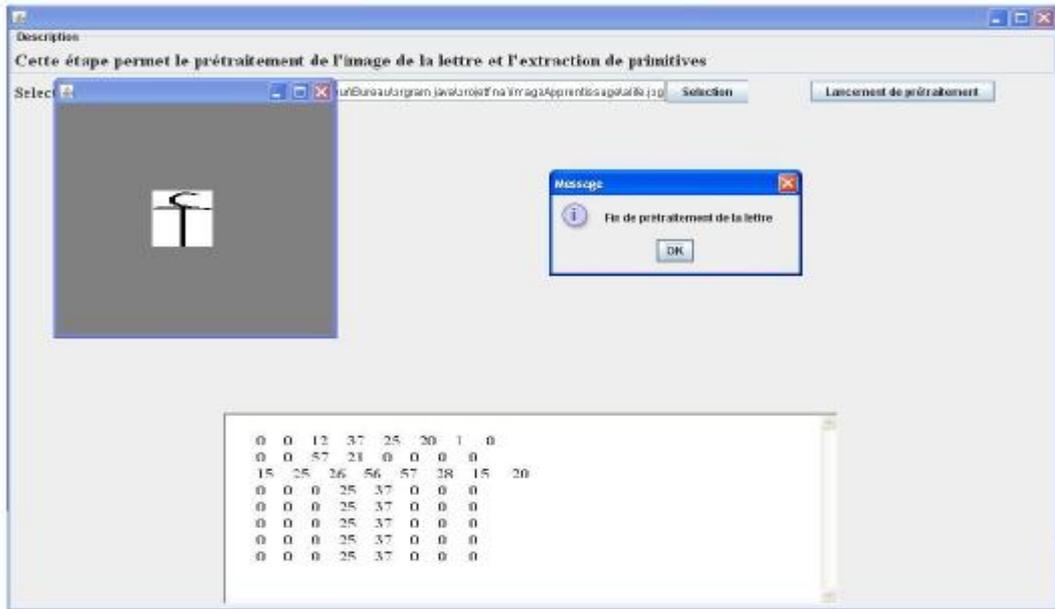


Figure 9: Les résultats de prétraitement et l'extraction de primitives.

- c. L'apprentissage : cette opération est effectuée après le chargement de l'image de la lettre et la sélection de son type dans la liste de choix.

Les résultats de l'apprentissage consistent en trois matrices A, B, et P qui correspondent aux paramètres de HMM. Ces matrices seront enregistrées dans la Base de données pour représenter la lettre « alif ».

La figure suivante montre le résultat de l'apprentissage :

- a. Segmentation en lignes : Après le chargement de l'image de nom, en cliquant sur le bouton « extraire les lignes » la segmentation sera effectuée. Les résultats seront sauvegardés dans un dossier créé par l'utilisateur afin d'être utilisé dans la segmentation en caractères.

La figure suivante montre les résultats de la segmentation en lignes de l'image de la figure 10:

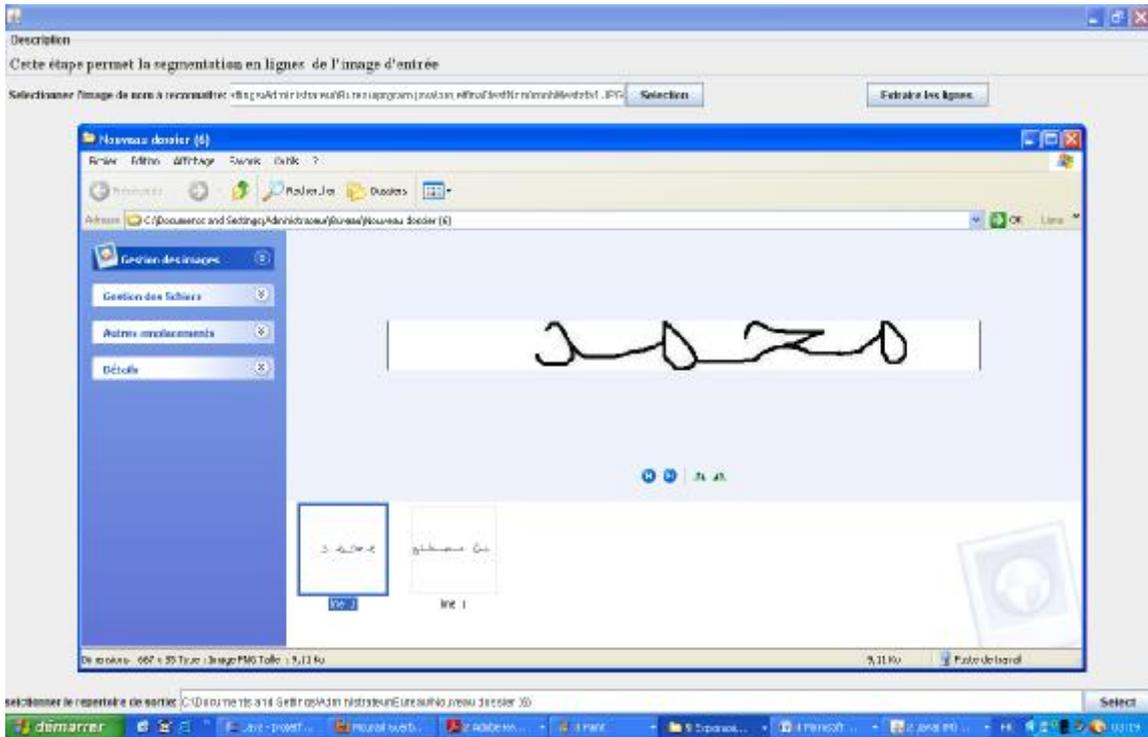


Figure 12: la segmentation en lignes

- b. Segmentation en caractères : Après le chargement de l'image à segmenter en caractères, en cliquant sur le bouton « segmentation en caractères », la segmentation sera effectuée et les résultats seront sauvegardés dans un dossier créé par l'utilisateur afin d'être utilisé pour la reconnaissance de nom.

La figure suivante montre les résultats de la segmentation en caractères du nom présenté dans la figure 10:

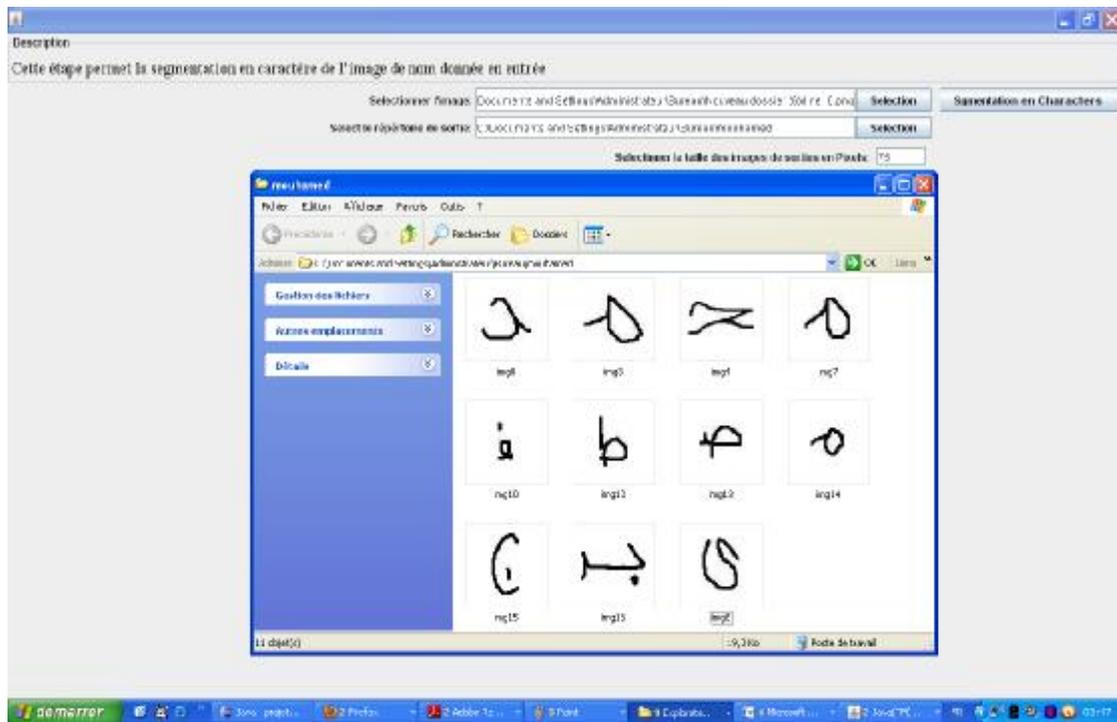


Figure 13: résultat de l'extraction de caractères.

c. La reconnaissance des caractères : Cette étape permet la reconnaissance de caractère donné en entrée.

Après le chargement de la lettre à reconnaître, en cliquant sur le bouton « reconnaissance », l'opération de reconnaissance sera effectuée et le résultat sera affiché dans la zone de texte :

- Ü La lettre correspond à l'image : il représente le nom de la lettre reconnue.
- Ü Son poids : il représente la somme des probabilités de chemin suivi dans le modèle HMM correspondant pour la reconnaissance de la lettre.
- Ü La lettre : elle représente la lettre reconnue en arabe.

La figure suivante représente le résultat de la reconnaissance de la lettre « ma » extraite de nom donné dans la figure 10:

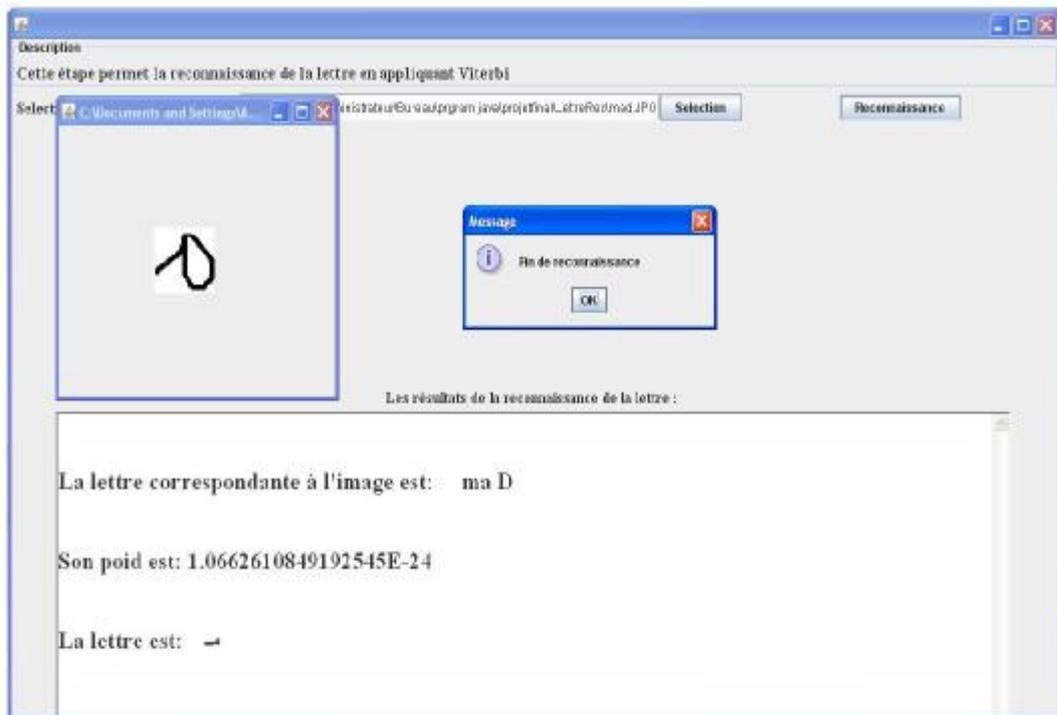


Figure 14 : Reconnaissance de la lettre « ma ».

- d. Reconnaissance de nom : Le but de cette étape est la reconnaissance proprement dite de nom. Après le chargement de l'image de nom, en cliquant sur le bouton « reconnaissance de nom », l'opération de reconnaissance sera effectuée et le résultat sera affiché dans la zone de texte :

La figure suivante montre le résultat de la reconnaissance de nom « محمد » :

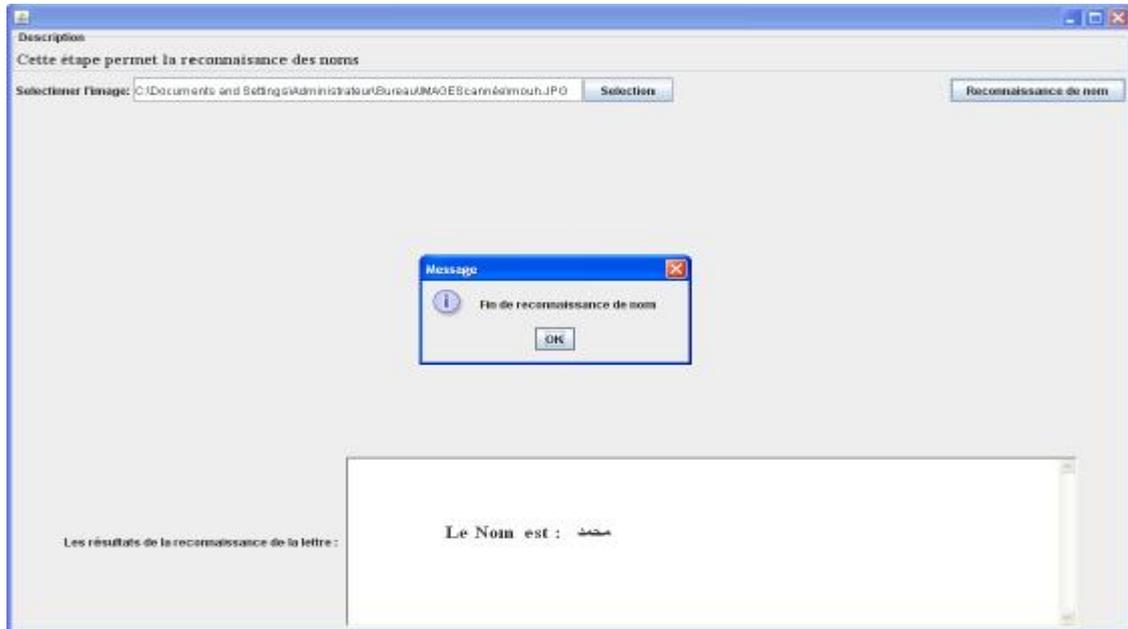


Figure 15 : Reconnaissance de nom

IV. Résultats Expérimentaux

A fin d'évaluer notre système de reconnaissance de noms arabes, nous avons effectué des expérimentations sur la reconnaissance des lettres arabes manuscrites et sur la segmentation en caractères. Car la reconnaissance de nom est fortement dépendante de la segmentation de nom en caractères et de la reconnaissance de ces derniers. Les résultats obtenus sont décrits ci-dessous :

IV.1. Tache de la reconnaissance des lettres arabes manuscrites

La performance de la reconnaissance de lettres peut être mesurée en calculant le taux de reconnaissance et le taux de substitution:

ü Taux de reconnaissance = Nombre de lettres reconnues / Nombre total de lettres.

ü Taux de substitution = Nombre de lettres males reconnues / Nombre total de lettres.

Après avoir testé la reconnaissance de lettre sur notre base de données composée de 98 lettres arabes, nous avons obtenu respectivement les taux de reconnaissance et de substitution (mauvaise reconnaissance) suivants : 73 % et 27%.

D'après nos testes, nous avons remarqué que notre système fait des confusions entre quelques lettres par exemple : da fin (ﺩ) et kha milieu (ﺦ) , qa milieu (ﻗ) et ta milieu (ﺕ), la début (ﺪ) et na début (ﻧ) ainsi entre les quatre lettres dja isolé (ﺝ) , 3a fin (ﻋ), ha fin (ﺡ) et Kha fin (ﺦ).

Ces confusions sont dues à l'étape de l'extraction de primitives qui est basé sur le calcul de pourcentage de pixels dans les zones de l'image. En effet, cette méthode ne fait pas la différenciation entre certaines lettres qui se ressemblent telle que : qa milieu (ﻗ), ta milieu (ﺕ) et 3a fin (ﻋ), ha fin (ﺡ).

IV.2. Tache de la segmentation

La performance de la segmentation en caractères est calculée comme suit :

Taux de segmentation = nombre de mots bien segmentés / sur le nombre total de mots

Nous avons testé la segmentation sur une base de teste composée de 100 noms arabes manuscrits, nous avons obtenu: 70%.

D'après nos testes, nous avons remarqués les noms contenant des lettres chevauchées et ligaturées sont mal segmentés.

La performance de notre segmentation est dépendante aussi de la hauteur des noms, tel que, dans certains cas où la hauteur de nom est grande le nom sera mal segmenté.

IV.3. Tache de reconnaissance des noms

La performance de reconnaissance des noms est dépendante de la segmentation en caractères et la reconnaissance des lettres.

Nous avons effectué des teste de reconnaissance des noms sur la base des noms utilisée dans la segmentation en caractères. Nous avons obtenus 71% noms reconnus.

V. Conclusion

Nous avons présenté dans ce chapitre les différents aspects du développement de notre système. Nous avons décrit précisément notre système et nous avons montré le mécanisme de reconnaissance des noms arabes manuscrits.

Pour valider notre système nous avons effectué des testes sur la segmentation en caractères des noms, la reconnaissance des caractères ainsi des testes sur la reconnaissance des noms.

Les résultats obtenus sont très encourageants, mais des détails demandent à être améliorés, comme par exemple l'extraction de primitives qui est l'une des raisons qui fait que le taux de reconnaissance des lettres n'est pas très élevé (73%). Ainsi, certaines erreurs de segmentation mais en les ignorant nous avons obtenu des résultats prometteurs.

*Conclusion
générale*

Conclusion

Notre travail se situe dans le cadre de la reconnaissance de l'écriture manuscrite. Nous nous intéressons en particulier, à la reconnaissance de l'écriture arabe manuscrite hors-ligne. La difficulté de ce sujet a amené plusieurs chercheurs à conduire plusieurs travaux pour remédier au problème de la reconnaissance. Cependant, nous la recherche demeure loin d'atteindre le niveau de la capacité humaine dans ce domaine.

L'handicap majeur de la reconnaissance pour les approches existantes est l'opération de la segmentation. Pour remédier à ce problème nous avons proposé un algorithme qui permet la segmentation de nom en caractères en éliminant les éléments d'épaisseur uniforme qui relient ces caractères.

Le système que nous avons proposé est composé de deux sous systèmes: un sous système d'apprentissage et un sous système de reconnaissance. Le système d'apprentissage a la capacité de traitement des images, et se charge aussi d'extraction des caractéristiques sous forme d'un vecteur de description, qui sera destiné à être traité par le modèle de Markov caché. Trois matrices (A, B, P) représentant chaque lettre sont générées (cf. algorithme de Baum Welch). Les résultats obtenus seront sauvegardés dans une base de données d'apprentissage. Chaque lettre est représentée sur la base de données d'apprentissage sous quatre formes différentes (début, milieu, fin, isolée).

Le sous système de reconnaissance à pour objectif la reconnaissance du nom. Il se charge d'abord du prétraitement de l'image du nom d'entrée. L'image sera segmentée en lignes puis en caractères. Chaque caractère passe par le module d'extraction des caractéristiques pour trouver son vecteur de description. Ce dernier sera utilisé par l'algorithme de Viterbi dont le but de trouver la lettre correspondante dans la base de données d'apprentissage. En fonction des lettres reconnues, le système sélectionne le nom le plus probable dans une base de noms arabes.

Nous avons implémenté notre système en langage Java. Le prototype réalisé respecte l'architecture que nous avons proposée pour la reconnaissance. Pour évaluer la performance de notre système en termes de précision de reconnaissance, nous avons mené une étude expérimentale qui porte sur la segmentation, la reconnaissance de caractères et la reconnaissance de noms. Les résultats obtenus sont prometteurs. En effet, nous avons reconnus un nombre importants des échantillons des noms testés.

Conclusion

Le travail réalisé nous ouvre plusieurs perspectives. Nous essayerons d'améliorer la performance de notre système en améliorant l'extraction de primitives qui est l'une des raisons qui fait que le taux de reconnaissance des lettres n'est pas très élevé. A cet effet, nous proposons la combinaison de la méthode zoning et le code Freeman (voir glossaire) qui décrit la forme géométrique d'un caractère, cette combinaison permet la distinction entre les caractères donc l'augmentation de taux de reconnaissance.

La performance de notre système peut être aussi améliorée par la combinaison de la méthode de Markov Caché avec la méthode de réseau de neurone.

Nous proposons un système de reconnaissance hybride qui intègre des modèles de Markov cachés (MMC) et des réseaux neuronaux (RN) dans une architecture probabiliste. Les noms manuscrits sont d'abord traités par un système de reconnaissance de mots basé sur des MMC guidé par un lexique. Une liste des N meilleures hypothèses ainsi que de la segmentation de ces mots en caractères est générée. Un classificateur à base de réseau neuronal calcule un score pour chaque caractère segmenté et les résultats des classificateurs MMC et RN sont combinés pour optimiser les performances de reconnaissance. Selon Ale Koerich et All [50], les résultats expérimentaux montrent que sur un vocabulaire de 80,000 noms le système hybride MMC/RN augmente le taux de reconnaissance de 9% relativement à un système de reconnaissance MMC seul.

Bibliographie

Bibliographie

- [1] : Nawwaf n. kharma et Rabab k. ward , « systèmes de reconnaissance de caractères pour les non-experts », université de colombie-britannique, 1999.
- [2] : Riadh Bouslimi, « système de reconnaissance hors-ligne des mots manuscrits arabe pour multi-scripteurs», mémoire de mastère, université de Jendouba, 2006.
- [3] : Farés Menasri, «contributions à la reconnaissance de l'écriture arabe manuscrite», thèse de doctorat, université Paris Descartes, 2008.
- [4] : Bougamouza Fateh, Hazmoune Samira et Benmohammed Mohammed, « nouveaux prétraitements et extraction des caractéristiques pour la reconnaissance des mots manuscrits arabes », université de Skikda, 1955
- [5] : Abdel Belaïd, « Reconnaissance automatique de l'écriture et du document », campus scientifique ,2002.
- [6] : Sylvain Chevalier, « reconnaissance d'écriture manuscrite par des techniques markoviennes : une approche bidimensionnelle et générique », thèse de doctorat, université René Descartes, 2004.
- [7] : Giraud Charles et Lacoste Bastien, « reconnaissance de caractères », Laboratoire de Recherche en Informatique Paris, 2006.
- [8] : Emmanuel Augustin, « reconnaissance de mots manuscrits par systèmes hybrides réseau de Neurones et modèles de Markov cachés », thèse de doctorat, université René Descartes Paris V, 2001.
- [9] : Mélanie Lemaitre, « approche markovienne bidimensionnelle d'analyse et de reconnaissance de documents manuscrits », thèse de doctorat, université paris 5 René Descartes département de mathématiques et informatique, 2007
- [10] : Emilie Poisson, « Architecture et apprentissage d'un système hybride neuro-marcovien pour la reconnaissance de l'écriture manuscrite », thèse de doctorat, université Nantes, 2005
- [11] : Fouad Slimane, « Modèles de Markov cachés et modèle de longueur pour la reconnaissance de l'écriture arabe à basse de résolution », thèse de doctorat, université de Fribourg, 2009.

Bibliographie

- [12] : Laurence Likforman-sulem, « analyse d'images de documents non contraints : de la structuration a la reconnaissance », mémoire de synthèse, université pierre et marie curie (paris VI), 2008.
- [13] : Mohamadally Hasan, « SVM : machines à vecteurs de support ou séparateurs à vastes marges », Versailles et Quentin France, 2006.
- [14] : Najoua Ben Amara, Abdel Belaïd et Noureddine Ellouze, « utilisation des modèles markoviens en reconnaissance de l'écriture arabe : Etat de l'art », l'école nationale d'ingénieurs de Monastir - 5019 Monastir – Tunisie, 2003.
- [15] : A. Benouareth, A. Ennaji et M. Sellami, « Utilisation des HMMs de Durée d'Etat explicite pour la Reconnaissance des Mots Arabes Manuscrits », Laboratoire de Recherche en Informatique, Laboratoire PSI -FRE 2645 Université de Rouen, INSA de ROUEN, 2000
- [16] : Khalid Hallouli, « reconnaissance de caractère par méthodes Markovienne et réseau bayésiens », Telecom Paris Ecole nationale supérieur des télécommunications, 2004.
- [17] : S.Aupetit, N.Monmarché et M.Sliman, « Apprentissage de Modèles de Markov caché par Essaim Particulière », thèse de doctorat, Laboratoire d'informatique de l'université François-Rabelais de Tours, 2000.
- [18] : Sayad Mahmoud et Abbaris Amara, « Modèle discriminant pour la classification de documents XML à l'aide des réseaux bayésiens et le noyau de Fisher », mémoire d'ingénieur d'état, Ecole nationale supérieur d'informatique INI, 2010.
- [19] : Alp Mestan, « Introduction aux réseaux de Neurone artificiels feed forward », 2008.
- [20] : Elise Gabarra, « De la binarisation de document vers la reconnaissance de symboles dans l'analyse de schéma électriques », thèse de doctorat, université de Pau et des Pays de l'Adour, 2008.
- [21]: Niblack, « An introduction to image processing », Prentice Hall, Englewood Cliffs, pp 115-116, 1986.
- [22]: E. Levin et R. Pieraccini, «Dynamic planar warping for optical character recognition», Proc. International conference on acoustics, speech and signal processing (ICASSP'92), pp. III-149-III-152, 1992.

Bibliographie

- [23] : Z. Wimmer, S. Garcia-Salicetti, B. Dorizzi et P. Gallinari, « Reconnaissance de mots manuscrits par une approche neuro-markovienne : coopération en-ligne hors-ligne », Proc. 1er Colloque international francophone sur l'écrit et le document (CIFED'98), pp. 354-363, Québec, Canada, 1998.
- [24]: SJ Young, «The htk hidden markov model toolkit: Design and philosophy », Cambridge university Engineering department Trumpington Street, 1994.
- [25]: R. Collobert, S. Bengio et J. Mariéthoz. Torch , «A modular machine learning software library», IDIAP Research Report 02-46, Martigny, Switzerland, 2002.
- [26]: Janez Demsar, Bláz Zupan, Gregor Leban et Tomaz Curk. Orange, « From Experimental machine learning to interactive data mining», In PKDD '04 : Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, pages 537–539, New York, NY, USA: Springer Verlag New York , 2004.
- [27]: K. F. Lee, H. W. Hon et R. Reddy, «An overview of the sphinx speech recognition system», Acoustics, Speech, and Signal processing, IEEE Transactions on, 38(1): 35–45, 1990.
- [28] : Abderrahmane kefali, Toufik Saret et Mokhtar Sellami, « Evaluation de plusieurs techniques de seuillage d'images de documents arabes anciens », Laboratoire de Recherche en Informatique (LRI), Département Informatique, BP 12 – Université Annaba, 23000, Algérie, 2000.
- [29] : Sophie Schbath, « Les chaînes de Markov cachées : présentation et usage en analyse de séquences, biologiques », Unité Mathématique, Informatique et Génome, INRA – Jouy-en-Josas, 1999.
- [30]: L. E. Baum, «An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes», Inequalities vol. III, 1972.

Bibliographie

- [31]: F. AGEN et J. MICHOT, « Chaînes de Markov cachées Algorithme de Baum-Welch », Ecole polytechnique université de Tours, 2004-2007.
- [32] : Kaouthar Bouriel1, Fadoua Bouafif Samoud, Haikal El Abed Samia Snoussi et Maddouri, « Stratégies d'évaluation et de comparaison de méthodes d'extraction de la ligne de base de mots manuscrits arabes », Ecole Nationale d'Ingénieurs de Tunis (ENIT) Laboratoire des Systèmes et Traitement du Signal (LSTS), 2000.
- [33]: R. Bertolami, S. Uchida, M. Zimmermann et H. Bunke, «Non-uniform slant correction for handwritten text line recognition, Document Analysis and Recognition», ICDAR Ninth International Conference on, 23-26 Sept. 2007.
- [34]: J. E. Bresenham, «Algorithm for computer control of a digital plotter», IBM System Journal 4(1) (1965), pp 25-30.
- [35]: S. Impedovo, R. Modugno et D. Impedovo, «Advancements in Handwriting Recognition », Département Informatique, Université de degli Studi de Bari, 1999
- [36] : Encyclopédie universelle Wikipedia – Le manuscrit : Notion générales sur les manuscrits. Disponible sur le Web à l'adresse <http://fr.wikipedia.org/wiki/manuscrit>.
- [37] : Alexandre Hervieu, « Analyse de trajectoires vidéos à l'aide de modélisations markoviennes pour l'interprétation de contenus », thèse doctorat, l'Université de Rennes, 2009.
- [38] : Cinthia O. A. Freitas, Luiz S. Oliveira, Simone B. K. Aires et Flávio Bortolozzi, « Metaclasses and Zoning Mechanism Applied to Handwriting Recognition », Pontifical Catholic University of Paraná – PUCPR, Brazil, 2008
- [39]: Cinthia O. A. Freitas, Luiz S. Oliveira et Flavio Bortolozzi, «Handwritten Character Recognition Using Nonsymmetrical Perceptual Zoning», Pontifical Catholic University of Parana (PUCPR), 2007
- [40] : Al Falou Wassim, «Reconnaissance de caractères manuscrits par réseau de neurones», thèse de doctorat, Université de Rennes – IRISA, 1998

Bibliographie

[41] : Adnan Amin, «Off line arabic character recognition - a survey», In ICDAR '97 : Proceedings of the 4th International Conference on Document Analysis and Recognition, pages 596–599, IEEE Computer Society, 1997.

[42]: Liana M. Lorigo et Venu Govindaraju, «Offline arabic handwriting recognition –survey IEEE Transactions on Pattern Analysis and Machine Intelligence», 2006.

[43]: M. Cheriet, «Strategies for visual arabic handwriting recognition : issues and case study», International Symposium on Signal Processing and its Applications, Sharjah, United Arab Emirates, 2007.

[44]: Abdallah Benouareth, Abdellatif Ennaji et Mokhtar Sellami, « Reconnaissance des Mots Manuscrits Arabes par Combinaison d'une Approche Globale et une Approche Analytique », Laboratoire de Recherche en Informatique, Université Badji Mokhtar, 2000.

[45]: J. Ziv et N. Merhav, «Estimating the number of states of a finite state source», Institut d'ingénieur en électrique et électronique, New York, NY, ETATS-UNIS, 1992.

[46] : Xavier Dupré, « Contributions à la reconnaissance de l'écriture cursive à l'aide de modèles de Markov cachés », thèse de doctorat, Université René Descartes - Paris V U. F. R. d'Informatique et de Mathématiques Appliquées, 2004.

[47] : Patrick PECCATE - Table ronde du Campus XML - 28/02/2003 – disponible sur le Web à l'adresse : www.ipa-france.com.

[48] : Marc Parizeau, « Le perceptron multicouche et son algorithme de rétropropagation des erreurs », département de génie électrique et de génie informatique, Université Laval, 2004.

[49] : Denis Arrivault, « Apport des Graphes dans la Reconnaissance Non-Contrainte de Caractères Manuscrits Anciens », thèse doctorat, Université de Poitiers, 2002.

[50] : Ale Koerich, Yann Leydier, Robert Sabourin¹ et Ching Y.Suen³, « Système Hybride de Reconnaissance de Mots Manuscrits sur un Grand Vocabulaire utilisant des Réseaux

Bibliographie

Neuronaux et des Modèles de Markov Cachés », Lab. d'Imagerie, de Vision et d'Int. Artificielle, École de Technologie Supérieure Montréal, QC, Canada, 2002.

Sources Internet

[a1]: http://fr.wikipedia.org/wiki/Traitement_d'image, dernière consultation : Mai, 2011.

[a2]: http://fr.wikipedia.org/wiki/Image_numérique, dernière consultation : Mai 2011.

[a3]: <http://fr.wikipedia.org/wiki/Pixel>, dernière consultation : Mai 2011.

[a4]: <http://fr.wikipedia.org/wiki/luminance>, dernière consultation : Mai 2011.

[a5]: http://fr.wikipedia.org/wiki/Probabilit%C3%A9_conditionnelle, dernière consultation :
Mai 2011.

[a6]: http://fr.wikipedia.org/wiki/Histogramme_imagerie_numérique, dernière consultation :
Mai 2011.

Notions d'image

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film ou divers procédés électroniques de visualisation [a1].

C'est aussi une projection à deux dimensions d'une scène à trois dimensions. Elle est définie comme une matrice où la valeur de chaque élément (pixel) représente l'intensité discrète de la lumière en ce point. Chaque cellule élémentaire de l'image (pixel) a comme caractéristique un niveau de gris ou couleur prélevé à l'emplacement correspondant dans l'image réelle.

Pixel

Le pixel est le plus petit élément de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification, aussi le petit élément que peuvent manipuler les matériels et les logiciels d'affichage ou d'impression [a3].

Image binaire

Une image binaire est une matrice dont les niveaux de gris sont réduits aux deux éléments 0 et 1. Le niveau 0 représente le noir absolu et le niveau 1 représente le blanc. Dans ce cas un seul bit est utilisé pour coder le niveau de gris [20].

Image numérique

L'image numérique est l'image dont laquelle ont été apporté des modifications par ordinateur elle est représentée par une série de bits. L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou *pixels*, ayant chacun comme caractéristique un niveau de gris ou de couleur prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter [a2].

La numérisation d'une image est la conversion de celle-ci à partir de son état analogique (distribution continue d'intensités lumineuses dans un plan xOy), vers une image numérique représentée par une matrice bidimensionnelle de valeurs numériques $f(x,y)$ où :

- x, y : coordonnées cartésiennes d'un point de l'image.
- $f(x, y)$: niveau de gris en ce point.

Résolution

La résolution d'une image est le nombre de pixels par unité de mesure de l'image. Lorsque la résolution est élevée, elle garantit une bonne qualité de l'image [20].

Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins [20].

La luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface. Pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet [a4].

Histogramme d'une image

L'histogramme d'une image est représenté par un graphique, l'axe horizontal (abscisse) représente l'ensemble des valeurs de luminosité potentielles de l'image [a6] : c'est la quantité de lumière (ou l'intensité lumineuse) sur 256 niveaux, allant de 0 (le noir) à 255 (le blanc). L'axe vertical (ordonnée) représente la proportion de pixels dans l'image pour chacun des niveaux de luminosité (Figure 1).

Les tons foncés sont représentés à gauche de l'histogramme. Plus on va vers la droite, plus on augmente la luminosité. De même, lorsque l'histogramme monte, le nombre de pixels augmente. Lorsqu'il descend, le nombre de pixels diminue. Si un grand nombre de pixels se situent sur la partie gauche de l'histogramme, cela signifie qu'une majorité de pixels se trouvent dans les basses lumières.

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

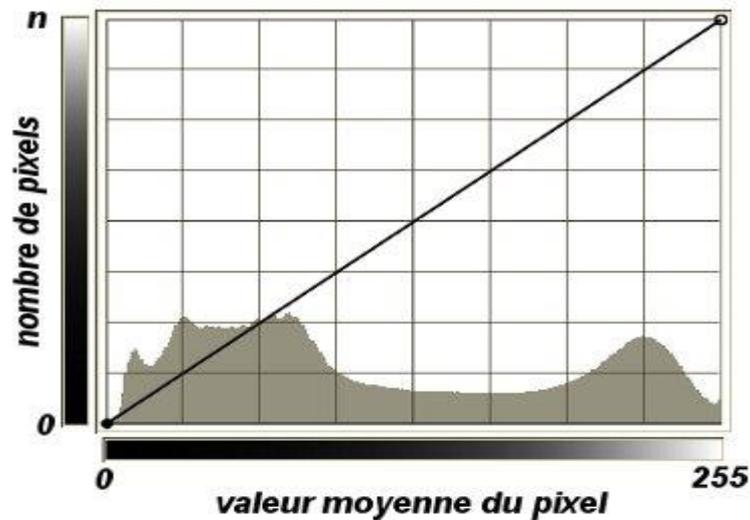


Figure 1: Exemple d'un histogramme.

Les règles Morphologiques

Le système décompose le mot en préfixe, suffixe, infixe et radical et reconnaît ces différentes parties. En pratique, on commence par extraire les candidats de préfixe et de suffixes qui sont cohérents les uns avec les autres [3]. Cette première sélection permet de déterminer la zone possible pour le radical, réduisant ainsi l'espace de recherche.

Les règles topologiques

Ce type de règles est basé sur des densités de pixels. On peut par exemple projeter des images de tailles différentes (les graphèmes) dans une matrice de taille fixe. Les caractéristiques extraites sont les valeurs des cellules de cette matrice [3].

Dans ce type de règles, on compte également les profils et les histogrammes. Pour maintenir un vecteur de taille fixe, on divise l'image en un nombre fixe de bandes horizontales et verticales. Les caractéristiques sont les moyennes des valeurs sur ces bandes.

Forward [37]

$\alpha_t(i)$: est une variable appelée variable forward, et définie par :

$$\alpha_t(i) = p(y_{1:t}, s_t = i | \lambda).$$

La variable forward $\alpha_t(i)$ est la probabilité d'observer la séquence partielle $y_{1:t}$ pour l'état $s_t = i$, conditionnellement à λ .

L'algorithme de calcul des $\alpha_t(i)$ est le suivant :

Initialisation :

$$\alpha_1(i) = \pi_i b_i(y_1).$$

Récurrance :

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^M \alpha_t(i) a_{ij} \right] b_j(y_{t+1})$$

Finalisation :

$$p(y_{1:T}|\lambda) = \sum_{i=1}^M \alpha_T(i)$$

Backward [37]

$\beta_t(i)$: est une variable appelée backward, définie par :

$$\beta_t(i) = p(y_{t+1:T} | s_t = i, \lambda)$$

La variable backward $\beta_t(i)$ est la probabilité d'observer la séquence partielle $y_{t+1:T}$ sachant que l'état $s_t = i$.

L'algorithme de calcul des $\beta_t(i)$ est le suivant :

Initialisation :

$$\beta_T(i) = 1$$

Récurrance :

$$\beta_t(i) = \sum_{j=1}^M a_{ij} b_j(y_{t+1}) \beta_{t+1}(j), \quad 1 \leq i \leq M, \quad t = T - 1, \dots, 1$$

Probabilité conditionnelle

En théorie des probabilités, la **probabilité conditionnelle** d'un événement A , sachant qu'un autre événement B de probabilité non nulle s'est réalisé (ou probabilité de A , sachant B) est le nombre noté : $\mathbb{P}(A|B)$ défini par [a5] :

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

Le réel $\mathbb{P}(A|B)$ se lit « probabilité de A , sachant B ».

Mathématiquement, soient $(\Omega, \mathcal{B}, \mathbb{P})$, un espace probabilisé et B un événement de probabilité non nulle (non quasi-impossible). À tout événement A de \mathcal{B} , nous associons le nombre noté $\mathbb{P}(A|B)$ ou $\mathbb{P}_B(A)$ défini par:

$$\mathbb{P}_B(A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} .$$

Nous pourrions vérifier que l'application \mathbb{P}_B définie par $A \mapsto \mathbb{P}_B(A)$ est une probabilité.

Si A et B sont indépendants alors: $\mathbb{P}(A|B) = \mathbb{P}(A)$

Le théorème de Bayes permet d'écrire

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)}$$

Rétro propagation du gradient

En informatique, la technique de rétro propagation du gradient (*Backpropagation* en anglais) est une méthode qui permet de calculer le gradient de l'erreur pour chaque neurone d'un réseau de neurones, de la dernière couche vers la première [48].

De façon abusive, on appelle souvent technique de rétro propagation du gradient l'algorithme classique de correction des erreurs basé sur le calcul du gradient grâce à la rétro propagation et c'est cette méthode qui est présentée ici. En vérité, la correction des erreurs peut se faire selon d'autres méthodes, en particulier le calcul de la dérivée seconde. Cette technique consiste à corriger les erreurs selon l'importance des éléments qui ont justement participé à la réalisation de ces erreurs. Dans le cas des réseaux de neurones, les poids synaptiques qui contribuent à engendrer une erreur importante se verront modifiés de manière plus significative que les poids qui ont engendré une erreur marginale.

Ce principe fonde les méthodes de type algorithme du gradient, qui sont efficacement utilisées dans des réseaux de neurones multicouches comme les perceptrons multicouches (MLP pour « multi-layers perceptrons »). L'algorithme du gradient a pour but de converger de manière itérative vers une configuration optimisée des poids synaptiques. Cet état peut être un minimum local de la fonction à optimiser et idéalement, un minimum global de cette fonction (dite fonction de coût).

Normalement, la fonction de coût est non-linéaire au regard des poids synaptiques. Elle dispose également d'une borne inférieure et moyennant quelques précautions lors de l'apprentissage, les procédures d'optimisation finissent par aboutir à une configuration stable au sein du réseau de neurones.

Les métadonnées

Définies comme données sur les données jouent un rôle fondamental dans la gestion des ressources numériques. Dans notre cas, nous devons définir des éléments de passerelle qui permettront aux utilisateurs d'accéder et d'utiliser les manuscrits en tant que ressources numériques [47]. Ces éléments passerelle sont un support informationnel sur la ressource recherchée, d'où le terme métadonnées.

Vocabulaire limité

C'est un vocabulaire restreint, généralement il est souvent appelé « dictionnaire », « lexique » ou tout simplement « vocabulaire » [3]. Il contient la liste des noms que le système de reconnaissance pourra être amené à reconnaître.

On utilise généralement la convention suivante :

- petits vocabulaires : quelques dizaines de mots.
- vocabulaires de tailles moyennes : quelques centaines de mots.
- vocabulaires de grandes tailles : quelques milliers de mots.
- vocabulaires de très grandes tailles : quelques dizaines de milliers de mots.
- vocabulaire ouvert : pas de lexique pour guider la reconnaissance.

Lorsque la taille du vocabulaire augmente, la tâche de reconnaissance devient de plus en plus complexe, car des mots ressemblants ont plus de chances d'être présents dans le vocabulaire. La complexité calculatoire est également liée à la taille du vocabulaire.

Code Freeman [49]

Ce code sert à coder un contour dans une image, et toute structure connexe est bien codée par Freeman. Chaque pixel est représenté dans la chaîne par l'entier qui désigne sa position par rapport au pixel précédent.

Le code de Freeman se base seulement sur 3 éléments pour caractériser une forme :

- Les coordonnées(x, y) d'un point appartenant au contour de la forme que l'on veut caractériser
- Un sens de rotation
- Une chaîne de caractères qui code le contour

La chaîne commence conventionnellement par les coordonnées du point de départ. Dans l'exemple de la figure suivante, le code de la chaîne de Freeman serait :

(3,3)707010133453210134321701133445456656666766

