

REPUBLICQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE – MINISTERE DE
L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE –
ECOLE NATIONALE POLYTECHNIQUE.

D0007/02A



DEPARTEMENT D'ELECTRONIQUE
**THESE DE DOCTORAT D'ETAT
EN ELECTRONIQUE**

Présentée par :

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

HAMAMI née MITICHE Latifa

Ingenieur d'Etat en Electronique

Magister en Electronique Option Acquisition et Traitement de l'Information

**SYSTEMES DE RECONNAISSANCE DE
L'ECRIURE ARABE IMPRIMEE**

Devant le Jury d'examen composé de:

Président :	AKSAS Rabia	Professeur	ENP	ALGER
Rapporteur :	BERKANI Daoud	Professeur	ENP	ALGER
Examineurs :	GUESSOUM Abderrezak	Professeur	Université	BLIDA
	SMARA Youcef	Maitre de conférences	USTHB	ALGER
	BELOUHRANI Adel	Maitre de conférences	ENP	ALGER

21 Septembre 2002

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE – MINISTERE DE
L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE –
ECOLE NATIONALE POLYTECHNIQUE.



DEPARTEMENT D'ELECTRONIQUE
**THESE DE DOCTORAT D'ETAT
EN ELECTRONIQUE**

Présentée par :

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

HAMAMI née MITICHE Latifa

- Ingénieur d'Etat en Electronique
- Magister en Electronique Option Acquisition et Traitement de l'Information

**SYSTEMES DE RECONNAISSANCE DE
L'ECRITURE ARABE IMPRIMEE**

Devant le Jury d'examen composé de:

Président :	AKSAS Rabia	Professeur	ENP	ALGER
Rapporteur :	BERKANI Daoud	Professeur	ENP	ALGER
Examineurs :	GUESSOUM Abderrezak	Professeur	Université	BLIDA
	SMARA Youcef	Maître de conférences	USTHB	ALGER
	BELOUHRANI Adel	Maître de conférences	ENP	ALGER

21 Septembre 2002

DEDICACES

Je dédie ce modeste travail:

- A la mémoire de mon cher mari Ahmed, que Dieu lui accorde sa miséricorde et lui réserve une place dans son vaste Paradis.
- A mes quatre fils : Ameer el Amine, Adel, Mohamed el Badr et Billel qui resteront le but essentiel de ma vie ; que Dieu me les garde et me donne la force et la capacité de parfaire à leur éducation.
- A mes très chers parents qui ont su me donner les clés de la réussite : mon éducation et ma formation. Que Dieu les récompense.

Remerciements

Je remercie tout d'abord Dieu le tout puissant de m'avoir donné le courage de surmonter les moments difficiles de ma vie et la volonté de mener à terme ce travail de recherche.

Je remercie le professeur D. Berkani de m'avoir suivi après le départ du professeur A. Chekima et de m'avoir soutenu tout au long de ce travail. Ses conseils m'ont été très précieux.

Je remercie messieurs A. Guessoum, Y. Smara et A. Belouchrani d'avoir accepté d'être membres du Jury et d'user de leurs compétences pour examiner et juger ce travail.

Je remercie également le Professeur R. Aksas d'avoir accepté la présidence du Jury.

Je remercie toutes les personnes qui ont, à divers titres, contribué à faire avancer mon travail et particulièrement les étudiants en graduation et post-graduation qui sous ma direction ont participé à la recherche.

Je remercie enfin mon fils Ameer el Amine, étudiant à l'ENP, pour sa contribution quant à la finalisation de ce document.

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

Sommaire

Liste des figures

INTRODUCTION

1

CHAPITRE I : Généralités sur la Reconnaissance des Formes

I. Introduction	4
II. Qu'est ce que la Reconnaissance des Formes ?	5
III. Domaines d'application de la RF	5
III.1 Reconnaissance de signaux d'origine humaine	5
III.2 Reconnaissance des caractères ou lecture optique	6
III.3 Reconnaissance de signaux d'origine naturelle	7
IV. Système Général de RF	8
IV.1 Acquisition de la forme	9
IV.2 Prétraitement	10
IV.2.1 Le filtrage	10
IV.2.2 La segmentation	12
IV.3 Paramétrisation ou Extraction des caractéristiques	18
IV.4 La Reconnaissance	20
IV.4.1 Apprentissage	20
IV.4.2 La Décision	21
V. Méthodes utilisées pour la RF	22
V.1 Méthodes statistiques	22
V.2 Méthodes structurelles	25
V.3 Méthodes connexionnistes	27
VI. Conclusion	28

CHAPITRE II : Reconnaissance de l'écriture

I. Introduction	29
II. Historique et état de l'art	29
III. Les systèmes OCR dans l'analyse de la structure de documents	34
III.1 Normes et structuration du document	36
IV. Les problèmes de l'OCR	41
V. Quelques définitions utilisées en OCR	42
VI. Les différents aspects de l'OCR	43
VII. Les techniques de l'OCR	44
VIII. Structure générale d'un système OCR	45
VIII.1 Schéma fonctionnel d'un système OCR	45
VIII.2 Module d'acquisition	46
VIII.3 Prétraitement	47
VIII.4 Segmentation	49
VIII.5 Reconnaissance	50
IX. Conclusion	51

CHAPITRE III : Implémentation des modules de Prétraitement

I. Introduction	53
II. Présentation de l'écriture arabe	53

S

O

M

M

A

I

R

E

II.1 Historique	53
II.2 Styles d'écriture	55
II.3 L'alphabet arabe	56
III. Acquisition et Filtrage	60
IV. segmentation d'un texte	61
IV.1 Localisation des lignes de texte	62
IV.2 Séparation des lignes en parties connexes	64
IV.3 Segmentation des parties connexes en caractères	65
IV.4 Problèmes rencontrés et solutions apportées à la segmentation	67
V. Conclusion	69

CHAPITRE IV : Implémentation des modules de Reconnaissance et Décision

I. Introduction	70
II. Méthode statistique	
II.1 Extraction des caractéristiques	71
II.2 Apprentissage	73
II.3 Reconnaissance et Décision	73
II.4 Conclusion	74
III. Méthode structurelle en arbres	75
III.1 Classification des caractères	75
III.2 Apprentissage	85
III.3 Reconnaissance et décision	89
III.4 Conclusion	94
IV. Méthode morphologique	94
IV.1 Classification et interprétation des caractères	95
IV.2 Apprentissage	106
IV.3 Reconnaissance et Décision	107
IV.4 conclusion	109
V. Utilisation des réseaux de Neurones	110
V.1 détermination du nombre de neurones sur chaque couche	110
V.2 Détermination des paramètres d'apprentissage	111
V.3 Architecture du réseau adopté	111
V.4 Apprentissage	112
V.5 Conclusion	113

CHAPITRE V : Résultats et interprétations

I. Introduction	114
II. Comparaison des différentes méthodes implémentées	114
III. Résultats obtenus en reconnaissance des blocs de documents	116
IV. Présentation du logiciel	116
V. Déroulement des différentes phases du logiciel	126
V.1 Résultats de l'acquisition et du prétraitement	127
V.2 Résultats de la segmentation d'un texte	129
V.3 Résultat de la Reconnaissance	132
VI. Conclusion	134

CONCLUSION	135
-------------------	------------

BIBLIOGRAPHIE

ANNEXES

Annexe A : Le format Tiff et organigramme de lecture d'un fichier Tiff.

Annexe B : Normalisation de la taille d'un caractère.

Annexe C : Algorithme de Reconnaissance par la méthode morphologique.

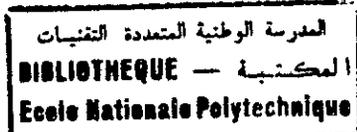
Annexe D : Les Réseaux de Neurones multicouches.

Annexe E : Algorithme de Rétro propagation du gradient de l'erreur

S
O
M
M
A
I
R
E

Liste des figures

Liste des figures



Chapitre I

Figure I.1 : Système général de Reconnaissance des Formes	8
Figure I.2 : Principe du filtre médian	11
Figure I.3 : Dérivation en présence d'un contour. a) Contour, b) Gradient, c) Laplacien	14
Figure I.4 : Segmentation par étiquetage en composantes connexes	17
Figure I.5: Les méthodes statistiques	23
Figure I.6 : Graphe ET/OU	26
Figure I.7 : Graphe relationnel attribué	27

Chapitre II

Figure II.1 : Mots arabes et leurs différentes parties connexes (ou sous mots)	33
Figure II.2 : Place de l'OCR dans "Document Image Processing"	35
Figure II.3 : Relation entre les structures physique et logique avec le contenu d'une lettre d'affaire	38
Figure II.4 : Objets génériques de la norme ODA	39
Figure II.5 : Relations entre les structures logique et physique dans un document ODA	39
Figure II.6 : Système de Reconnaissance des caractères	52
Figure II.7 : Exemple du mot "عمل" squelettisé	49
Tableau II.1 : Extrait du jeu de délimiteurs dans SGML	40

Chapitre III

Figure III.1 : Carte du monde arabe et de ses zones d'influence linguistique due à la pénétration de l'Islam	54
Figure III.2 : Exemples de ligatures	55
Figure III.3 : Différentes positions de la lettre hamza	56
Figure III.4 : Quelques styles utilisés en langue arabe	57
Figure III.5 : Caractères arabes selon leurs quatre positions dans un mot	58
Figure III.6 : Caractères ayant même tracé principal	59
Figure III.7: (a) image bruitée, (b) image filtrée	60
Figure III.8: Block diagramme d'un système OCR	62
Figure III.9: Segmentation horizontale	63
Figure III.10 : Détection d'une ligne de texte	64
Figure III.11 : Segmentation verticale	65
Figure III.12 : Segmentation en caractères	67
Figure III.13 : Segmentation du caractère « sin »	68

Chapitre IV

Figure IV.1 : Organisation du dictionnaire	73
Figure IV.2 : Classification des caractères	77
Figure IV.3 : disposition des 8-voisins de p_i	78
Figure IV.4 : Re-échantillonnage a) Nouvelle grille d'espacement b) Forme re-échantillonnée	80
Figure IV.5 : Les huit directions de Freeman	82
Figure IV.6 : Contour (a), Squelette (b) et Suivi du squelette (c) du caractère « sin »	82

Liste des Figures

Figure IV.7 : Arbre du caractère « Sin »	83
Figure IV.8 : Structure du dictionnaire	88
Figure IV.9 : Corvar (taux de remplissage)	96
Figure IV.10 : Encadrement du caractère	98
Figure IV.11 : Les transitions	99
Figure IV.12 : Les points essentiels	99
Figure IV.13 : Exemple de détection de concavités et boucles	101
Figure IV.14 : organisation du dictionnaire	108
Figure IV.15 : Architecture du réseau de neurones adopté	111
Figure IV.16 : Procédure de reconnaissance par réseau de neurones	112
Tableau IV.1 : vecteurs de caractéristiques associés aux différentes classes	87



Chapitre V

Tableau V.1 : Caractéristiques des différentes méthodes implémentées	115
Figure V.1 : Image originale	117
Figure V.2 : Séparation des blocs majeurs du document de la figure V.1	118
Figure V.3 : Séparation des lignes à l'intérieur des blocs texte du document de la figure V.1	119
Figure V.4 : Reconnaissance des blocs du document de la figure V.1	120
Figure V.5 : Séparation des mots à l'intérieur des lignes de texte du document de la figure V.1	121
Figure V.6 : Ouverture d'un fichier	127
Figure V.7 : Résultat du filtrage médian	128
Figure V.8 : Résultat de la segmentation horizontale	129
Figure V.9 : Résultat de la segmentation verticale	130
Figure V.10 : Résultat de la segmentation en caractères	131
Figure V.11 : Une partie des codes des caractères reconnus	132
Figure V.12 : Boîte de dialogue permettant de fixer les paramètres d'apprentissage	133
Figure V.13 : Résultats de la reconnaissance	133
Figure V.14 : Menu d'aide	134

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Introduction

INTRODUCTION

Les systèmes informatiques ne se contentent plus d'aider l'homme dans des tâches qualifiées de répétitives et fastidieuses, mais tentent plutôt de communiquer avec lui et d'acquérir les mêmes possibilités de perception. Bien qu'ils soient loin d'avoir les mêmes capacités, les mêmes facilités de perception et d'adaptation que l'homme, les machines informatiques entrent de plus en plus en compétition avec lui et parfois avec beaucoup de succès.

Si nous pensons à remplacer l'homme par des machines dans des tâches composées de situations plus complexes d'interaction avec l'environnement ou à automatiser certaines activités pour lesquelles, nous employons des données plus réelles et plus naturelles ; il nous paraît de plus en plus nécessaire de munir ces machines d'un système de perception visuelle. L'analyse et l'interprétation d'une image sont à la base d'un tel système.

On définit la reconnaissance des formes comme étant la science qui réunit un ensemble de techniques informatiques de représentation et de décision permettant aux machines de simuler quelques comportements similaires à ceux des humains, en leur offrant ainsi le privilège d'être qualifiées de machines intelligentes.

La reconnaissance vise donc à équiper les ordinateurs d'organes de sens. Cela ne signifie pas simplement la mise en place de systèmes physiques, mais aussi et surtout l'extraction d'informations pertinentes: la catégorisation des phénomènes perçus (y compris l'apprentissage des catégories) et l'utilisation des connaissances des experts humains dans les domaines concernés.

Autrement dit, la reconnaissance de formes a pour objet, d'une part de capter et de décrire en mémoire des formes (c'est à dire des manifestations de l'univers extérieur auquel la machine a été rendue sensible) et d'autre part de prendre sur la représentation mémoire ainsi obtenue, une décision d'identification par référence à un ensemble d'apprentissage déduit d'une représentation analogique.

La reconnaissance de l'écriture s'inscrit dans le cadre de la reconnaissance de formes, elle a été élargie depuis quelques années à l'analyse de document composite devenu de jour en jour un thème de recherche à part entière.

Il est certain que la lecture automatique a une importance capitale si on compte les centaines de milliers de lettres, chèques et rapports qui s'échangent quotidiennement. C'est pour cette principale

raison qu'est née la reconnaissance de caractères dont le but essentiel est la simplification des rapports entre l'homme et la machine.

Il ne s'agit plus de considérer le caractère comme la seule composante de l'image, mais il faut reconnaître la structure du document, séparer les parties textuelles des graphes et textes, coder et interpréter les graphes, etc.[1,2]

Un système de reconnaissance de caractères appelé **OCR** (Optical Character Recognition) consiste à récupérer automatiquement des textes à partir d'un document imprimé, en évitant la phase fastidieuse de la saisie.

Pour cette phase, on peut dire que pouvoir s'affranchir de l'opération de saisie de documents imprimés ou dactylographiés grâce à un système automatique constitue un vieux rêve des services informatiques. Des recherches ont été entreprises, de part le monde pour reconnaître l'écriture imprimée, dactylographiée ou même manuscrite. C'est seulement en 1984/85 que les premiers produits sur micro-ordinateur ont fait leur apparition, et il a fallu attendre 1988 pour disposer d'un logiciel d'OCR Latin quasiment opérationnel. De nos jours, des systèmes très performants arrivent à des taux de reconnaissance très élevés pour l'écriture imprimée latine. Le manuscrit pose encore quelques problèmes vu la cursivité des lettres et leur variabilité d'une personne à une autre.

L'écriture arabe, objet de notre étude, a bénéficié de quelques recherches ; sa difficulté réside dans sa cursivité, sa large gamme de styles ainsi que les différentes formes d'un même caractère selon sa position dans le mot.

L'intérêt des systèmes **OCR** est varié. Parmi les tâches spécialisées dont l'automatisation présente un bénéfice économique important et des gains de productivité remarquables, on peut citer :

- La saisie de données (secteur bancaire).
- L'acquisition de textes repris de manuscrits ou de livres (secteur de la presse ou de l'édition).
- La reprise de textes dans tous les logiciels de traitement de textes ou de publication assistée par ordinateur.
- La saisie de textes pour entrer en synthèse vocale ou en édition sur imprimante Braille pour non voyants **etc.**

Notre travail consiste en la réalisation d'un système de reconnaissance de l'écriture arabe imprimée de même ordre de complexité que l'écriture manuscrite latine à cause de sa cursivité. Pour ce faire, et dans le but de munir notre système des performances espérées pour un OCR, nous avons implémenté plusieurs méthodes de reconnaissance à savoir : une approche statistique utilisant les sept moments invariants [3,4], une approche structurale utilisant les arbres comme descripteurs [5] ainsi qu'une méthode structurale syntaxique utilisant des paramètres morphologiques du caractère pour sa classification ainsi que d'autres paramètres heuristiques qui ont permis de différencier entre les caractères d'une même classe [6,7,8]. Cette dernière approche est celle qui corrige les faiblesses des deux précédentes, nous avons essayé de l'optimiser en utilisant un réseau de neurones multicouches [9]. Notre système de reconnaissance procède en trois étapes, à savoir :

1. L'acquisition des données à lire (texte arabe imprimé).
2. Le filtrage du texte à étudier ainsi que sa segmentation en caractères.
3. La reconnaissance des caractères pour laquelle nous utilisons les trois méthodes déjà citées.

La segmentation d'un texte arabe en caractères n'est pas sans difficultés ; le système que nous avons réalisé résout des problèmes non encore résolu jusque là, à savoir ceux de la sur segmentation de certains caractères arabes assez complexes dans leurs formes [10,11].

Notre thèse est organisée ainsi en cinq chapitres :

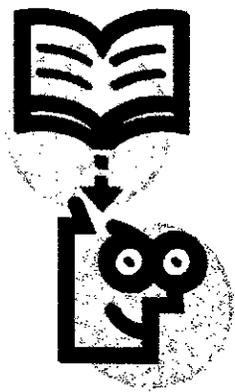
On présente, à travers les deux premiers chapitres, un état de l'art sur la reconnaissance des formes en général puis sur la reconnaissance de l'écriture en particulier où une étude théorique sur les systèmes OCR et la place qu'ils occupent dans l'analyse du document sont données.

Les trois chapitres suivants concernent la partie pratique. On y décrit les différents systèmes réalisés. Les parties acquisition et segmentation du texte sont communes à tous les systèmes OCR réalisés et constituent le chapitre trois ; quant à la partie reconnaissance et décision, elle est différente d'une approche à une autre. Nous avons adapté plusieurs approches de reconnaissance des formes aux caractères dans le souci d'apporter des améliorations au système précédemment implémenté. Les différents systèmes implémentés sont présentés au chapitre quatre. Les résultats obtenus ainsi que leurs interprétations et les améliorations à y apporter éventuellement sont données au chapitre cinq.

Nous proposons enfin quelques perspectives en conclusion.

CHAPITRE I

Généralités sur la Reconnaissance des Formes



- I. Introduction
- II. Définition de la RF
- III. Domaines d'application de la RF
- IV. Système Général de RF
- V. Méthodes utilisées pour la RF
- VI. Conclusion

I. Introduction

L'énorme quantité d'informations apportées par les systèmes informatiques de saisie et de stockage de mesures que l'homme est amené à traiter ces dernières années, a favorisé le développement de branches comme la reconnaissance des formes (RF) pour l'interprétation automatique des données.

Si les machines informatiques sont désormais acceptées comme indispensables en tout ce qui concerne le calcul numérique et la gestion, on a quelques réticences à les reconnaître aptes au traitement des données non purement chiffrées.

Pourtant l'ordinateur est par définition l'outil universel capable d'effectuer n'importe quel type de calcul, numérique ou symbolique. Cette universalité provient du fait qu'on a le choix du type de représentation pour les données dans les machines, et que l'on peut sur ces représentations programmer n'importe quel algorithme.

Aucun argument fondamental n'empêche de tenter de faire réaliser à de telles machines des algorithmes qui mènent à des comportements traditionnellement réservés à l'être intelligent : reconnaître une personne à sa voix, comprendre un texte, se déplacer dans un environnement non connu a priori, etc....

Le problème peut se résoudre en représentant d'une façon adéquate les données et en les modélisant sous forme d'un programme.

La difficulté provient du fait que l'homme effectue généralement ses tâches d'une manière inconsciente et inanalysable, après un long apprentissage et selon des structures de représentation et de décision mal connues : nous sommes bien en peine d'expliquer comment nous lisons, comment nous reconnaissons un objet, une parole, encore mieux nous sommes incapable de ne pas les reconnaître.

Bien qu'elles soient loin d'avoir les mêmes capacités, les mêmes facilités de perception et d'adaptation que l'homme, les machines informatiques rentrent en compétition avec lui parfois avec beaucoup de succès.

C'est pourquoi, des recherches sont entreprises pour comprendre et approcher ce que l'homme peut faire, aussi bien dans le domaine de la perception (Reconnaissance des Formes) que celui de la compréhension (Intelligence Artificielle).

II. Qu'est ce que la Reconnaissance des Formes ?

La reconnaissance des formes est définie comme étant l'ensemble des techniques informatiques de représentation et de décision permettant aux machines de simuler un comportement sensible.

Le but de la reconnaissance des formes est d'obtenir des programmes permettant de reproduire efficacement les phénomènes de perception de l'être humain. Elle consiste donc à doter la machine d'organes de sens (capteurs physiques tels que : scanner, caméra, microphone) captant l'information extérieure sous forme variée.

Elle a pour objectif d'une part de décrire ces informations captées, et d'autre part, de prendre sur la représentation ainsi obtenue, une décision d'identification par référence à un ensemble d'apprentissage.

De ces définitions surgissent les difficultés suivantes :

- Saisir (ou percevoir) l'information.
- La représenter convenablement.
- Prendre une décision d'identification dans cet espace de représentation.

La reconnaissance des formes est un domaine qui regroupe plusieurs sous domaines telle que la reconnaissance de la parole, la vision par ordinateur... etc. La vision par ordinateur est un domaine qui s'intéresse au traitement et l'interprétation de l'image. L'entrée d'un système de vision est constituée d'une image. Le système apporte un certain nombre de connaissances qui interviennent à tous les niveaux. La sortie est une description de l'entrée en termes d'objets et de relations entre ces objets [12].

La vision par ordinateur permet, à partir de l'image d'un objet ou d'une scène réelle, d'en déduire des données exploitables par une machine. C'est aussi l'ensemble des techniques et méthodes qui participent à l'acquisition et au traitement des images en vue de prendre une décision de façon automatique. Elle fait intervenir les techniques d'intelligence artificielle (IA) tel que le traitement des connaissances et réseaux de neurones [13].

III. Domaines d'application de la R.F :

La diversité des organes de sens dont dispose la R.F donne une idée sur ses domaines ; nous allons citer quelques uns des plus importants.

III.1 Reconnaissance de signaux d'origine humaine :

L'exemple le plus connu est celui de la parole dont les performances sont au stade de la réalisation de machines d'usage public reconnaissant des suites de mots choisis parmi plusieurs dizaines. Il y a

aussi la reconnaissance du locuteur (coopératif) parmi quelques dizaines, dont les applications, notamment en sécurité, sont évidentes.

Les techniques de la R.F peuvent apporter une aide appréciable aux médecins pour les diagnostics et dépouillements de signaux médicaux, tels l'électrocardiogramme, l'électroencéphalogramme, l'électrorétinogramme, l'analyse des chromosomes [14,15], la mesure des débits sanguins.

Cependant les applications les plus avancées concernent les images biomédicales : l'analyse des cellules sanguines [16,17,18], la cytologie (étude des segments d'A.D.N, comptage globulaire, étude des structures de virus...), l'analyse des cellules cérébrales, etc.

III.2 Reconnaissance des caractères ou lecture optique

Bien que les traitements de l'information soient de plus en plus rapides, la saisie des données constitue un frein considérable aux performances des machines.

Quel que soit le degré d'information de l'entreprise, il y aura toujours une quantité importante d'informations à intégrer, en provenance de l'extérieur, et dans un grand nombre de cas, elle sera sous forme d'écrit sur support papier, en caractères imprimés, dactylographiés ou même manuscrits.

C'est pour éviter la création d'ateliers de saisie démesurés, que dès les années soixante, la reconnaissance optique des caractères est apparue comme la solution naturelle à l'automatisation des entrées massives de données.

En parlant de lecture optique, plutôt que de viser les machines capables simplement de lire des groupes de points et de les restituer, nous faisons référence aux machines capables de « reconnaître » qu'un tel groupe de points est un 'A' ou un 'B' : un texte reconnu par ces dernières pourra être introduit dans un traitement de textes pour le modifier ou dans une base de données pour l'indexer, de plus le stockage d'un texte sous forme graphique demande 100 fois plus de mémoire que sous forme ASCII, où chaque caractère est représenté par un nombre, quelque soit son corps (sa taille) et sa police (sa forme).

Cet autre signal important, émis par l'homme, qu'est l'écriture est un des champs favoris de la R.F, bien que les résultats restent médiocres comparés à la quantité de recherches effectuées dans ce domaine.

L'infinie variabilité de l'écriture manuscrite est à l'origine de la complexité du problème, et constitue la cause essentielle des semi-échecs récoltés. Mais ne nous leurrions pas car même l'être *intelligent*, considéré comme modèle idéal, n'est pas infallible : il commet 4% d'erreurs à la lecture d'un manuscrit en l'absence de contexte.

III.3 Reconnaissance de signaux d'origine naturelle

Des mesures faites sur la terre, les photos satellites, les contrôles sismologiques, la mesure de houle marine, sont des exemples de tels signaux ; les applications sont diverses :

- Détection des ressources, météorologie, prévision des phénomènes naturels, classement sur des images satellites (recherche des formes des routes, analyse des côtes marines et des rivières, etc.).
- En astronomie : reconnaissance des étoiles, étude des atmosphères planétaires, élimination des nuages sur les photos, etc.
- En physique : mesures sur les photos des chambres à bulles, traitement holographique, reconstitution des trajectoires de particules, etc.
- En chimie et matériaux : analyse de photos microscopiques de sédiments, défauts dans les alliages, analyse des fibres en papeterie, analyse des tableaux en muséologie, etc.
- En industrie : vision robotique (reconnaissance des pièces mécaniques, analyse des composants organiques dans les produits alimentaires, analyse des circuits VLSI, contrôle des soudures, diagnostics de pannes, tri du courrier, etc.).
- Détection à distance : identification des espèces végétales, étude de la pollution, ingénierie des travaux publics, étude du trafic routier, détection des zones inondables, hydrologie, température de la surface marine, étude de la houle, des glaciers, reconnaissance des roches, exploration minière, cartographie géologique, etc.

IV. Système général de Reconnaissance des Formes

La reconnaissance des formes est la première étape d'un long processus de compréhension de notre univers dans le cadre général de la communication homme-machine. Elle se doit de résoudre les premiers problèmes liés au codage des formes, leur paramétrisation et leur discrimination.

Nous allons nous intéresser particulièrement, à un système de reconnaissance d'images, appelé aussi système de vision par ordinateur.

Le but d'un système de vision par ordinateur, est d'analyser des images (bi ou tridimensionnelles) d'une scène donnée puis de reconnaître son contenu.

Le système doit passer au préalable par une étape d'apprentissage dont le rôle est d'éclairer la décision à l'aide de connaissances a priori.

La plupart de ces systèmes se partagent une structure générale qui est composée de quatre blocs principaux : (voir figure I.1)

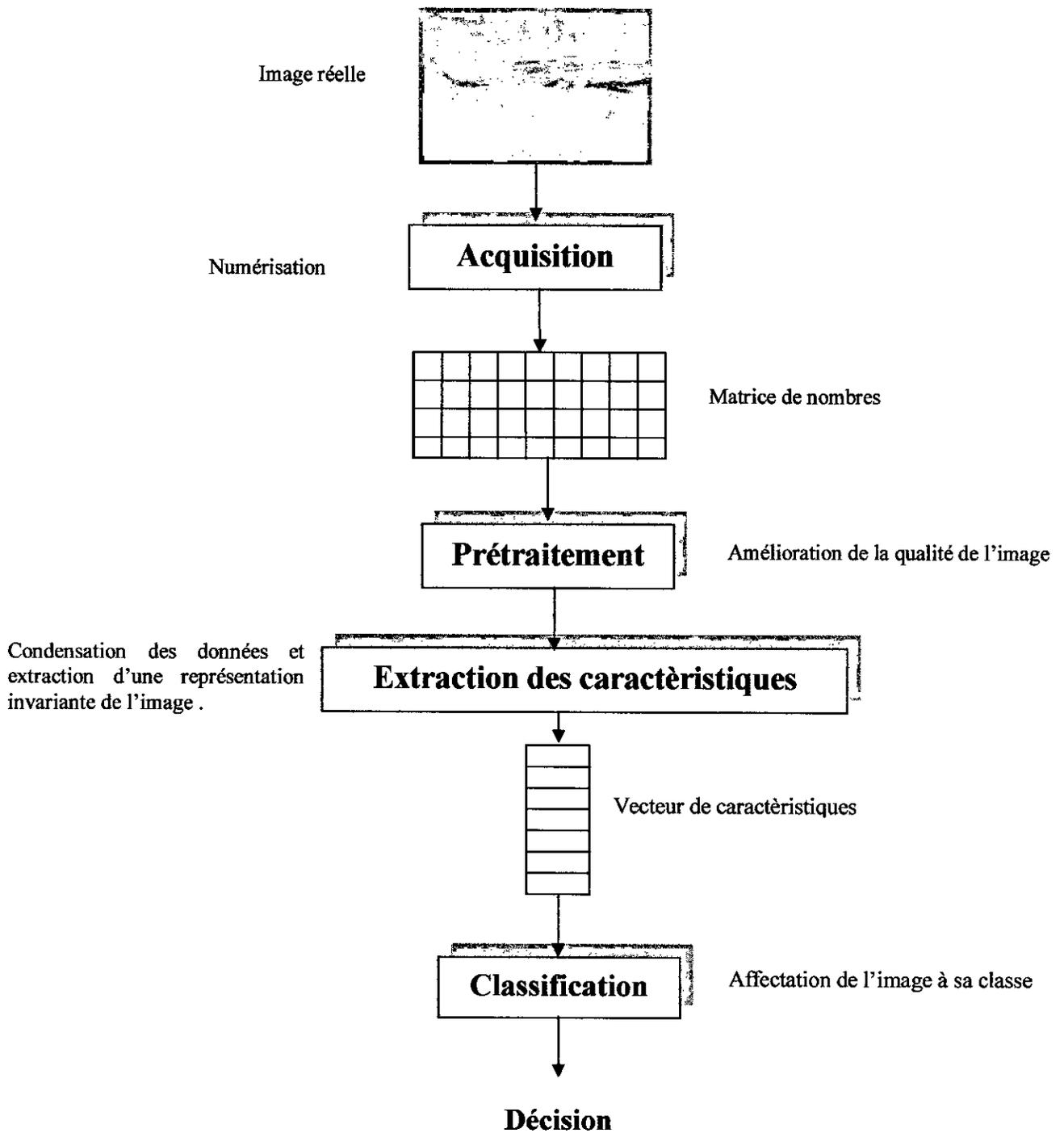


Figure I.1 : Système général de Reconnaissance des Formes

IV.1 Acquisition de la forme

Ce premier bloc convertit une scène en une matrice de nombres qui peuvent être manipulés par un ordinateur. Les outils les plus utilisés pour la digitalisation des images (passage du domaine continu vers le domaine discret) sont les caméras et les scanners qui captent l'intensité de la lumière émise par un certain nombre de points de l'image, et la codent pour former un fichier selon différents formats.

➤ La caméra est utilisée pour la prise d'images dans plusieurs domaines (cinématographie, biologie...). Il existe principalement trois types :

- **Les caméras à tube** : elles sont composées d'une cible photoconductrice qu'explore un faisceau lumineux, l'exploration se fait ligne par ligne (balayage). A chaque ligne correspond en sortie un signal analogique proportionnel à l'intensité.
- **Les caméras CCD (Charged Coupled Device)** : constituées par un assemblage de photodiodes, chacune d'elles délivre une intensité proportionnelle à un point de l'image appelé pixel. Ces caméras délivrent un signal échantillonné.
- **Les caméras numériques** [19] : ce type d'appareils n'a pas besoin de pellicule. Les images sont stockées sous forme numérique dans une disquette, un disque dur ou même une puce.

Leur principal avantage est le développement instantané des photos. En effet, elles peuvent être reliées à un PC par un câble ce qui permet de récupérer l'image de façon facile et pratique. Un appareil numérique possède un viseur à cristaux liquides (mini écran à cristaux liquides affichant l'image). Ainsi on peut visualiser parfaitement ce qu'on est sur le point de photographier et d'effectuer les réglages nécessaires. Une fois la photo prise, le viseur permet de la consulter, de l'effacer ou de l'archiver.

La qualité de l'image dépend essentiellement de l'appareil utilisé.

➤ **Le scanner** [19] est un périphérique de l'ordinateur. Il reçoit la lumière réfléchie par l'image éclairée d'une source lumineuse constante qui est fournie par des tubes lumineux.

Ces rayonnements sont recueillis par un miroir ou une lentille qui émet une tension proportionnelle à ces derniers. Ensuite, à l'aide d'un convertisseur, l'information électrique est transformée en une information numérique qui constitue un fichier exploitable.

Il existe plusieurs types de scanners, dont les plus utilisés sont :

- **Les scanners à plat** : le document est placé face à la plaque de verre, sous le couvercle.
- **Les scanners à rouleau** : les documents s'introduisent comme dans un télécopieur, la face dessus.

IV.2 Prétraitement :

Le rôle du prétraitement est de «préparer» les données reçues du capteur à la phase suivante d'analyse. Cette phase n'est possible et surtout fiable que si les données du capteur sont dénuées de bruit, corrigées de leurs erreurs éventuelles et réduites à l'essentiel. Toutes les techniques élaborées dans ce sens se gardent de modifier les propriétés essentielles des formes, ce qui pourrait conduire, dans le cas contraire à de graves erreurs d'analyse et plus tard de reconnaissance [20].

Si l'image obtenue lors de la phase d'acquisition contient plusieurs objets, elle est segmentée afin que chaque objet soit analysé séparément.

Le prétraitement peut donc être considéré comme étant constitué de deux phases : le filtrage et la segmentation.

IV.2.1 Le Filtrage :

Le bruit dans une image est l'ensemble des perturbations qui affectent la qualité de la segmentation des régions de l'image [21]. Il se manifeste par la présence d'informations résiduelles venant perturber les données propres à la forme [20]. Les pixels d'une image ont lorsqu'ils sont voisins les mêmes caractéristiques physiques, le bruit est un phénomène de brusque variation par rapport à ces voisins.

Le bruit est dû principalement [20] :

- Au dispositif d'acquisition : capteur.
- Aux conditions de prise de vue : éclairage, positionnement.
- A la scène elle-même : poussière, rayure.
- A l'environnement de la scène : fond, composition des formes, nature de la matière.

Pour palier à ces dégradations on utilise le filtrage. On distingue essentiellement deux types de filtres :

- **Filtres linéaires** : la transformation sur le pixel est le fruit d'une combinaison linéaire des niveaux de gris des pixels voisins.
- **Filtres non linéaires** : les pixels voisins interviennent dans la transformation suivant une loi non linéaire.

Dans ce qui suit, nous allons décrire deux exemples de filtres.

❖ **Le filtre moyen** [22] :

C'est un filtre linéaire dont le principe est de considérer chaque point de l'image et d'en faire la moyenne avec les huit pixels qui lui sont voisins. Ce filtre peut être mis sous la forme d'un masque

H , que l'on va déplacer sur toute l'image. Il faut noter que le pixel concerné par la transformation est le pixel central.

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Si I_1 représente l'image originale et I_F représente l'image finale après application du filtrage, alors :
 $I_F = H \otimes I_1$

Où \otimes représente le produit de convolution discrète du masque H avec l'image I_1 .

Effectuer le produit de convolution revient à calculer la formule (1), pour chaque pixel de l'image :

$$I_F(x, y) = \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} H(i+1, j+1) I_1(x+i, y+j) \quad (1)$$

❖ **Le filtre médian [22]:**

C'est un filtre non linéaire dont le principe est de calculer en un point non pas une combinaison linéaire des niveaux de gris de ses voisins, mais une valeur médiane dépendant d'un tri des niveaux de gris des points voisins.

On peut alors suivre les étapes suivantes (voir Figure I.2) :

- 1) On classe les pixels voisins du pixel concerné par la transformation par valeurs croissantes.
- 2) On prend la valeur médiane des pixels classés puis on l'affecte au pixel concerné (central).

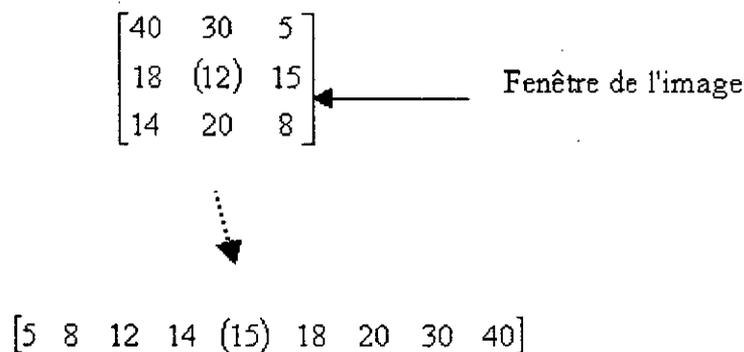


Figure I.2 : Principe du filtre médian.

Dans cet exemple 12 est l'ancienne valeur du pixel central et 15 sa nouvelle valeur.

Alors que le filtre moyen introduit du flou sur le bord des objets, le filtre médian permet l'élimination de parasites isolés dans l'image sans affecter les contours.

- **Filtrage morphologique** [22] :

Ce type de filtrage est utilisé pour éliminer des pixels isolés dans des images binaires (2 niveaux de gris : noir (0) et blanc (1)) qui sont considérés comme du bruit. Il met en correspondance chaque pixel et son voisin par une fonction logique (ET, OU, XOR). Parmi les opérateurs morphologiques, nous citons :

1. La Dilatation :

Elle permet d'éliminer les pixels blancs isolés. On effectue le **ET** logique des huit voisins du pixel considéré.

La dilatation élimine les tâches (les trous) blanches dans les zones noires mais ajoute des pixels noirs au contour des objets présents dans l'image.

2. L'Erosion :

Elle permet d'éliminer les pixels noirs isolés au milieu des parties blanches de l'image, on effectue le **OU** logique des huit voisins du pixel considéré.

En appliquant une érosion, ces tâches noires peuvent être éliminées mais la taille des objets présents dans l'image diminue car l'érosion enlève des pixels du contour, entraînant une déformation de certains objets.

3. L'Ouverture :

L'ouverture est constituée par une opération d'érosion suivie d'une dilatation. Elle permet de retrouver la taille normale des objets de l'image.

4. La Fermeture :

La fermeture est une opération morphologique qui consiste à faire subir à l'image une opération de dilatation suivie d'une érosion. Elle permet de retrouver la taille normale des objets de l'image.

IV.2.2 La Segmentation :

On est conduit pour analyser une image à la partager en un certain nombre de domaines connexes (unis) : ce processus joue un rôle de plus en plus important en traitement numérique des images. Deux points appartiennent à un même domaine s'ils possèdent une même propriété caractéristique P et si on peut les joindre par une chaîne continue de points voisins ayant cette propriété P. Une segmentation est donc définie par le type de propriété P et par la définition de voisinage [23].

La segmentation d'image étant la pierre de base de tout système de vision. Une image constitue une représentation d'un univers composé d'entités (objets). Le but de toute méthode de segmentation est l'extraction d'attributs caractérisant ces objets. Ces attributs correspondent à des points d'intérêt (contours) ou des zones caractéristiques de l'image (régions) [12]. D'où la naissance de deux approches : **Frontière et Région**.

- **Approche frontière** : les frontières correspondent aux contours des objets, qui possèdent l'essentiel des caractéristiques de la forme d'une région. La segmentation dans ce cas s'intéresse aux pixels composant le contour.
- **Approche région** : chaque région correspond à la surface de l'objet. La segmentation dans ce cas s'intéresse aux pixels composant ces régions.

➤ **Approche frontière :**

Un contour est un point de l'image qui appartient à la frontière entre deux régions voisines. C'est le lieu où l'on observe une discontinuité de la fonction des niveaux de gris [20].

Détecter les contours dans une image, revient à localiser ces discontinuités. Pour cela, deux approches sont utilisées : **Gradient et Laplacien**. Ces approches se basent sur la différentiation de l'image en la dérivant une fois, on obtient alors le **Gradient** ou bien deux fois on obtient ainsi le **Laplacien**. La (Figure I.3) montre les effets de la dérivation en présence d'un contour.

❖ **Approche gradient** : [22,12]:

Le contour est le lieu des points possédant des amplitudes élevées du gradient.

Formellement le gradient $\nabla f(x, y)$ d'un signal continu f est défini comme suit :

$$\nabla f(x, y) = \begin{pmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{pmatrix}$$

Le module du gradient comme étant:

$$N(x, y) = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2}$$

L'argument du gradient:

$$\arg(\nabla f(x, y)) = \arctg \left(\frac{\frac{\partial f(x, y)}{\partial x}}{\frac{\partial f(x, y)}{\partial y}} \right)$$

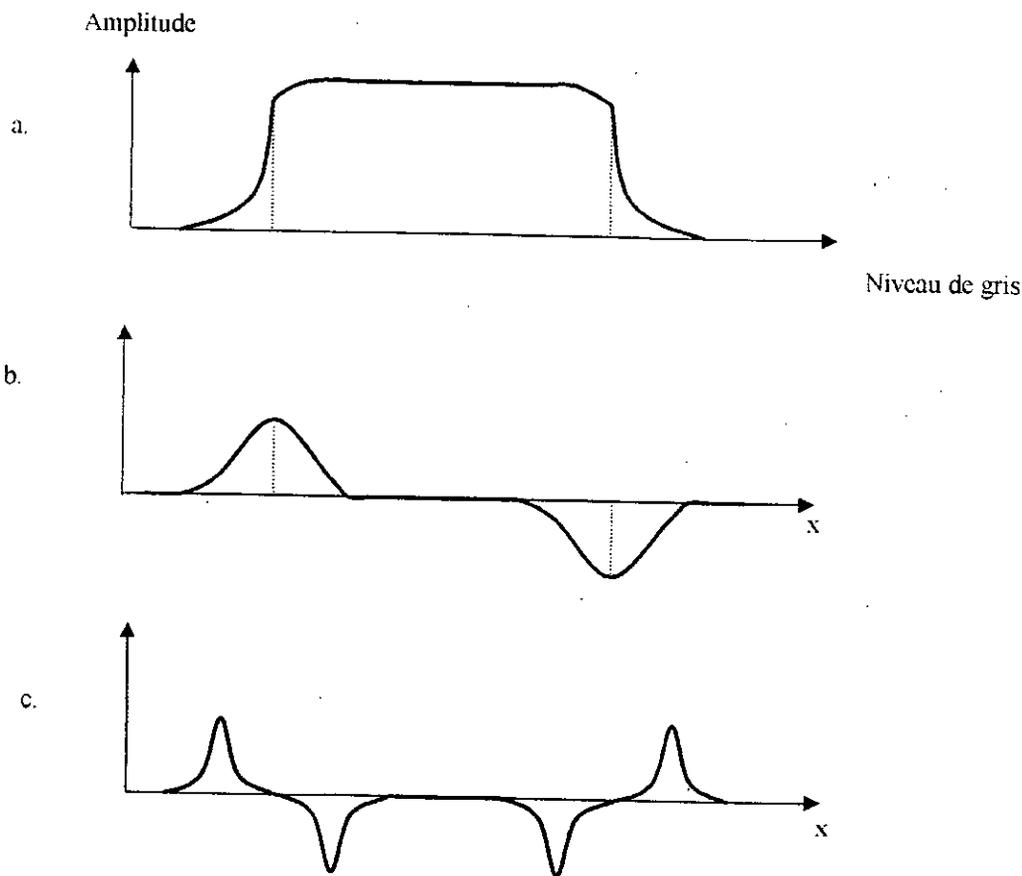


Figure I.3 : Dérivation en présence d'un contour. a) Contour, b) Gradient, c) Laplacien.

Dans le cas discret (l'image dans notre cas), pour détecter les contours il faut:

1) Calculer le gradient pour chaque point de l'image, ceci revient à utiliser l'approximation de ce dernier.

Dans une image I , l'approximation du gradient est le résultat de convolution d'une fenêtre de l'image I avec des masques, comme indiqué ci-dessous :

$$\frac{\nabla I(x, y)}{\hat{\partial}x} = M_1 ** I(x, y) \quad \text{Le gradient en } x$$

$$\frac{\nabla I(x, y)}{\hat{\partial}y} = M_2 ** I(x, y) \quad \text{Le gradient en } y$$

Un exemple de masques utilisés est celui de Perwitt où :

$$M_1 = \frac{1}{6} \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad \text{et} \quad M_2 = \frac{1}{6} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & 1 \end{pmatrix}$$

Un autre exemple est celui de Sobel :

$$M_1 = \frac{1}{6} \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad \text{et} \quad M_2 = \frac{1}{6} \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

2) Après cette opération, la détection de contour s'effectue en calculant d'abord le module du gradient pour chaque point de l'image (comme indiqué précédemment). Puis la détection des maxima locaux du gradient fournit l'ensemble des points contours de l'image.

Il faut noter qu'un point $P(x,y)$ est dit maxima local si:

Sa norme est supérieure aux normes de ses points adjacents se trouvant dans la même direction que le gradient.

❖ **Approche Laplacien** [22] :

Les points de contour correspondent aux valeurs nulles du Laplacien de l'image.

Formellement, le Laplacien $L(x, y)$ de tout signal continu $f(x, y)$ est défini comme étant la dérivée seconde de tout signal continu f .

$$L(x, y) = \frac{\partial^2 f(x, y)}{\partial^2 x} + \frac{\partial^2 f(x, y)}{\partial^2 y}$$

Pour une image I , la détection des contours revient donc à :

1) Calculer son Laplacien qui se ramène à effectuer le produit de convolution bidimensionnel de l'image avec l'un des masques H_1, H_2, H_3 :

$$H_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad H_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}, \quad \text{et} \quad H_3 = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

2) Déterminer les points de contour ; cela revient à détecter les passages par zéro du Laplacien de l'image I .

L'utilisation du Gradient demande des calculs complexes mais donne de bons résultats. le Laplacien est très sensible au bruit, de ce fait des procédures de seuillage sont alors appliquées. De plus d'autres procédures sont utilisées après la détection de contour qui sont :

- La binarisation pour mettre en évidence les points de contour acceptés ou rejetés.
- L'amincissement du contour pour que celui-ci ait une épaisseur d'un seul pixel.

- La fermeture de contour qui consiste à prolonger les extrémités des contours qui ne sont pas fermés.

➤ **Approche région :**

La segmentation en régions homogènes vise à segmenter l'image I en se basant sur des propriétés intrinsèques de la région qui déterminent le critère de segmentation [24].

- Segmentation par utilisation des critères d'homogénéité [20,22] :

La construction des régions de l'image se fait en utilisant un ou plusieurs critères d'homogénéité. Les principaux critères d'homogénéité d'une région utilisés sont :

- Niveau de gris.
- La couleur pour les images en couleur.
- Le mouvement.

Trois méthodes de segmentation en régions sont utilisées :

❖ Méthodes par séparation (méthodes globales) :

- A partir d'une image initiale, on teste si le critère d'homogénéité est vérifié.
- Si c'est le cas, l'algorithme s'arrête.
- Dans le cas contraire, on décompose l'image en zones plus petites, on ré applique la méthode pour chaque zone, jusqu'à l'obtention d'une zone qui valide le critère.

❖ Méthodes par fusion :

- A partir d'une partition assez fine de l'image (parfois composée d'un seul pixel).
- On fusionne les régions adjacentes si elles vérifient le critère d'homogénéité.
- On s'arrête quand le critère de fusion n'est plus vérifié.

❖ Méthodes mixtes (split and merge) :

Ces méthodes combinent les méthodes précédentes. A partir d'une partition arbitraire de l'image en régions, à chaque étape de l'algorithme les régions sont soit divisées si elles ne sont pas homogènes, soit fusionnées si elles sont similaires.

Ces méthodes nécessitent l'utilisation de structures de données compliquées (arbre). Par conséquent, le temps de calcul (parcours de l'arbre) et l'espace mémoire utilisé seront considérables.

- Segmentation par seuillage [24] :

Cette segmentation consiste à associer à chaque pixel la classe (région) à laquelle il appartient, et cela en comparant son attribut (par exemple niveaux de gris) à un ou plusieurs seuils.

Elle est utilisée pour segmenter une image composée de régions dont chacune est caractérisée par un attribut différent de l'autre. Formellement on écrit :

$$P_{i(i=1, \dots, I)} \in C_{l(l=1, \dots, L)} \text{ si } T_l < A(P_i) < T_{l+1}$$

Où P_i est le point i de l'image originale qui contient I points.

C_l est la classe numéro l , le nombre total des classes est L .

$A(P_i)$ est l'attribut du point P_i .

T_l et T_{l+1} sont les seuils qui déterminent la classe C_l .

Le problème réside dans le choix des seuils.

• Segmentation par étiquetage en composantes connexes [21] :

L'étiquetage en composantes connexes d'une image, attribue à tous les pixels d'une composante connexe (région, objet) une même étiquette (valeur entière). Cette méthode détecte les adjacences entre pixels et définit l'étiquette du point courant en fonction de celles des points voisins. Son principe est le suivant :

- Rendre l'image binaire.
- Partant d'un pixel, si une étiquette a été attribuée aux pixels voisins alors affecter la même étiquette à ce pixel sinon lui affecter une nouvelle étiquette.

Ce processus est répété jusqu'à ce que tous les pixels de l'image soient étiquetés. On obtient ainsi une image dont chaque région possède une étiquette E_i différente de l'autre (Figure I.4).

Cette méthode est utilisée pour segmenter une image qui possède plusieurs objets similaires en niveau de gris.

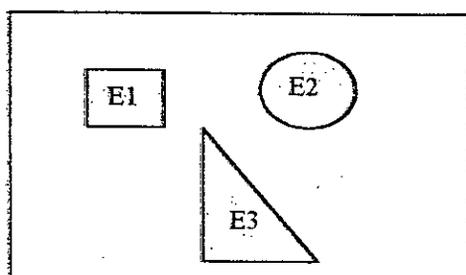


Figure I.4 : Segmentation par étiquetage en composantes connexes.

Puisque cette méthode opère sur une image binaire, nous allons expliquer brièvement la binarisation.

La binarisation [22, 25]:

La binarisation est une procédure de représentation d'images sur deux niveaux (0 pour noir et 255 pour blanc dans le cas où le pixel est représenté par un octet). Cette opération permet de créer une image binaire à partir de l'image d'origine. La binarisation peut être la première étape d'isolement des objets par rapport au fond. Elle nous permet soit d'isoler les objets en noir sur fond blanc ou l'inverse, soit pour isoler une partie de l'image (masquage) ou pour y effectuer des transformations morphologiques.

Algorithme de binarisation :

Pour chaque pixel de l'image :

si le niveau de gris du pixel > seuil alors pixel = blanc (255) ;

sinon pixel = noir (0) ;

IV.3. Paramétrisation ou Extraction des caractéristiques

C'est l'étape la plus importante et la plus délicate. Elle consiste à choisir l'espace de représentation de la forme.

A ce stade, l'image est représentée par un ensemble de caractéristiques numériques afin de réduire sa taille tout en gardant l'information essentielle et en supprimant la redondance. L'ensemble des caractéristiques sélectionnées doit posséder le maximum d'informations utiles existant dans l'image.

Cette étape est très importante puisque, à partir de cet instant, on ignore l'image initiale pour ne considérer que ses caractéristiques. Toute mauvaise extraction se propage vers les étapes ultérieures et influe négativement sur la décision finale du système.

De bonnes caractéristiques sont celles qui satisfont deux conditions :

- Petite variance intra classe : c'est à dire que des formes légèrement différentes avec des caractéristiques générales similaires, doivent avoir des valeurs numériques proches.
- Grande variance inter classe : c'est à dire que des caractéristiques de différentes classes doivent être bien différentes numériquement.

➤ **Extraction des paramètres [20] :**

L'étape d'extraction des paramètres suit l'étape de segmentation; elle consiste à extraire de l'image les informations les plus importantes qui permettent de décrire l'objet ou sa forme. Ces informations, se calculent en général non pas à partir de la totalité des points de l'image mais à partir d'un sous-ensemble de points (par exemple le squelette qui est le résultat d'une opération de squelettisation).

Selon le type d'images qu'on traite ainsi que la nature des objets, on peut distinguer plusieurs types de paramètres :

- **Les paramètres topologiques :**

Parmi lesquels on peut citer :

La surface S : égale par exemple au nombre de pixels situés à l'intérieur d'une région.

Le périmètre P : égal au nombre de pixels situés le long du contour d'une région.

- **Les paramètres de formes :**

Ces mesures permettent de donner des informations sur la forme des régions.

Exemple: la compacité C définie comme $C = \frac{S}{P^2}$. (S étant la surface, P le périmètre) permet de dire qu'un objet est un cercle si la valeur de C est maximale ($C = 1/4\pi$).

- **Les paramètres relationnels :**

Ces paramètres décrivent les relations ou rapports entre différentes régions. Il s'agit de définir une relation $f(R_i, R_j)$ entre deux régions R_i et R_j : la valeur de cette relation appartient à un ensemble fini de réponses et peut être de type booléen ce qui permet de confirmer ou infirmer une certaine propriété entre les deux régions.

Cependant, on peut distinguer trois types de paramètres : ceux de taille, de position et d'orientation.

- ❖ **Les paramètres de taille :**

où la relation se déduit à partir de la comparaison de la taille des régions (surface, périmètre...etc.) telle qu'une région soit plus grande qu'une autre ou qu'une région ait la même taille qu'une autre...etc.

- ❖ **Les paramètres de position :**

qui permettent de situer une région par rapport à une autre, on peut parler alors de position absolue d'une région ou encore de sa position relative. La première position peut se matérialiser par exemple par le centre de gravité de la région dans le plan, alors que la position relative prend en considération les régions voisines pour dire qu'une région se situe à gauche d'une autre région par exemple. Ce type de paramètre regroupe aussi la notion de distance entre régions définie par exemple comme étant la distance entre les centres de gravité de deux régions.

- ❖ **Les paramètres d'orientation :**

L'orientation d'un objet n'a de sens que si l'objet est allongé, c'est à dire que ses axes de moindre inertie sont parallèles (par exemple la barre verticale de la lettre « l »). Elle est utile pour décrire les relations d'alignement, de positionnement, de continuité et de parallélisme entre plusieurs objets

dans une scène (par exemple, trouver tous les segments verticaux représentant les bords des fenêtres dans une scène d'intérieur).

- **Les paramètres statistiques :**

Ces mesures permettent d'analyser la distribution statistique des pixels dans l'image, ce qui permet de s'en servir ultérieurement pour une préclassification en sous-ensembles.

Parmi ces paramètres, citons l'histogramme qui est un outil très important puisqu'il permet de définir au mieux l'information globale que contient l'image.

Les mesures de textures appartiennent, elles aussi, à ces paramètres, on peut citer :

La moyenne $\bar{g}_R = \frac{1}{S} \sum_{g=0}^{g=n-1} g \times Fr(g)$,

La variance $\sigma^2_R = \frac{1}{S} \sum_{g=0}^{g=n-1} (g - \bar{g}_R)^2 \times Fr(g)$,

Où Fr est la fréquence d'apparition d'un niveau de gris g dans une région R de n niveaux de gris et S sa surface.

Plus le nombre des caractéristiques est réduit, plus le temps de calcul sera réduit pendant la phase d'analyse et de leurs extractions.

IV.4. La Reconnaissance

La reconnaissance regroupe les deux étapes d'apprentissage et de décision qui jouent des rôles assez proches dans les systèmes de RF. En effet, à partir de la même description de la forme en paramètres, elles tentent, toutes les deux, d'attribuer cette forme à un modèle de référence.

IV.4.1 Apprentissage

La décision nécessite de définir clairement la connaissance que l'on a sur les formes à traiter. Cette définition repose sur l'apprentissage qui se charge d'acquérir la connaissance et de l'organiser en classes ou modèles de référence. Cette phase sert donc à construire le dictionnaire contenant les caractéristiques d'un ensemble d'exemples de prototypes afin de servir de référence pendant la reconnaissance. Le résultat de l'apprentissage peut être soit la réorganisation ou le renforcement des modèles existants, soit la création d'un nouveau modèle représentant la forme entrée.

Il existe deux types d'apprentissage, supervisé et non supervisé.

- **Apprentissage supervisé :**

L'apprentissage est dit supervisé si les différentes familles des formes sont connues a priori et si la tâche d'apprentissage est guidée par le superviseur (ou le concepteur) qui indique pour chaque

forme échantillon rentrée, le nom de la famille qui la contient. Les familles sont réparties dans des classes séparées entre elles.

L'apprentissage se déroule de la manière suivante : on choisit un ensemble des formes à reconnaître. On décrit chacune des formes après l'avoir analysée; cette description est prise en compte par la tâche d'apprentissage. L'opérateur indique ensuite le nom de la forme entrée, c'est à dire celui de la classe dans laquelle il souhaite la ranger. Le travail de l'apprentissage consiste alors à analyser les ressemblances entre les formes d'une même famille et les dissemblances entre les formes de familles différentes pour en déduire les classes de l'apprentissage avec les meilleures séparatrices possibles. Les paramètres décrivant cette partition sont stockés dans une base d'apprentissage.

Plus l'ensemble d'apprentissage est grand, plus les séparatrices peuvent être déterminées avec précision et meilleurs sont les résultats de la décision.

• **Apprentissage non supervisé :**

On l'appelle aussi classification automatique ou encore apprentissage sans professeur. Il s'agit, à partir d'échantillons de référence et de règles de modélisation, de construire automatiquement les classes ou les modèles sans intervention de l'opérateur. Ce type d'apprentissage nécessite un grand nombre d'échantillons et des règles de construction précises et non contradictoires. Ce mode évite l'assistance d'un opérateur mais n'assure pas toujours une classification correspondant à la réalité attendue par l'utilisateur.

IV.4.2 La Décision

C'est l'étape finale du système de reconnaissance qui consiste à identifier la forme inconnue en la comparant à des prototypes établis dans l'étape précédente.

La comparaison avec l'un des prototypes peut se faire de deux manières :

- Superposition de formes.
- Détection de similitudes.

La première solution consisterait à superposer l'image à reconnaître à un modèle préalablement appris et stocké dans une base de données. L'inconvénient est que la méthode est bonne pour des objets simples, bien définis et se présentant de la même façon (même orientation, même échelle). De plus, la base de données est grande. Ceci nous amène à utiliser la notion de distance entre images (mesure de dissemblance entre l'image et le modèle). L'image est identique au modèle si la distance est inférieure à un certain seuil ou inférieure à la distance entre la même image et chacun

des autres modèles. L'inconvénient est que la distance n'est pas invariante vis-à-vis d'un déplacement (rotation ou translation) de l'image ou d'une déformation.

La seconde solution consiste à passer par une représentation des connaissances, ce sont donc des systèmes à base de connaissances. Chaque objet est défini comme un ensemble de primitives ou paramètres qui permettent de déterminer ce que représente l'objet de l'image. Ces paramètres sont considérés comme une signature de l'objet qui sera stockée et comparée à des données préalablement apprises. L'inconvénient est que la reconnaissance dans un système de vision concerne des objets non nécessairement définis d'avance ou attendus. Or même les systèmes à base de connaissances ne résolvent que très péniblement et incomplètement ce type de problème, faisant intervenir des connaissances très diverses et très diffuses.

D'où l'idée de se tourner vers les réseaux de neurones qui sont des structures appropriées pour résoudre les problèmes de classification et se comportant comme des mémoires associatives.

V. Méthodes utilisées pour la RF

Les méthodes d'identification relèvent de trois grandes catégories :

- Les méthodes statistiques
- Les méthodes structurelles, qui s'intéressent à la structure des formes et à leur description en terme d'assemblage de sous formes primitives.
- Les méthodes connexionnistes qui sont basées sur les réseaux de neurones formels considérés comme des discriminateurs linéaires complexes.

Des liens ont été établis par des chercheurs entre ces modèles et les méthodes statistiques. En effet, un réseau connexionniste fournit une estimation de ressemblance d'une forme donnée avec celle apprise par le réseau.

V.1 Méthodes statistiques

Ces méthodes se fondent sur une caractérisation statistique des paramètres des formes étudiées. Elles permettent de prendre une décision de classification d'une forme inconnue suivant un critère de « probabilité maximale d'appartenance à une classe ». Elles peuvent être paramétriques ou non paramétriques comme explicité sur la figure I.5.

Ces méthodes effectuent des mesures sur la forme à identifier ; la forme est représentée par un vecteur de n composantes ; chacune de ces composantes est elle même une évaluation du trait pertinent qu'elle représente.

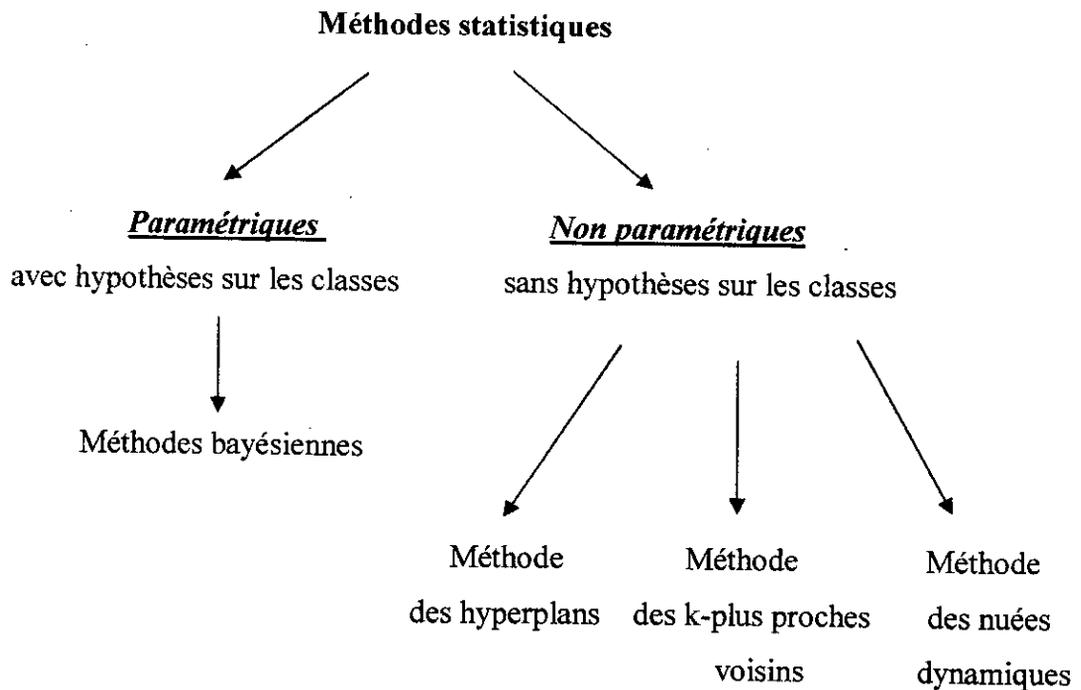


Figure I.5: Les méthodes statistiques

V.1.1 Méthode de BAYES [26]

Cette technique est une méthode optimale de reconnaissance des formes dans un cadre probabiliste. La théorie bayésienne de la décision sert de référence pour les autres méthodes, en particulier pour l'appréciation du taux d'erreur.

Cette technique peut se formuler de la façon suivante :

Soit un ensemble caractérisé par un ensemble de N observations $X = \{x_1, x_2, \dots, x_n\}$ réparties en C classes $\{c_1, c_2, \dots, c_c\}$, on cherche à estimer la probabilité d'appartenance a posteriori de x à chacune des C classes.

Règle de décision bayésienne

On utilisera les notations suivantes :

$P(c_i)$: probabilité a priori d'appartenance à la classe c_i .

$P(x / c_i)$: la densité de probabilité conditionnelle dans la classe c_i .

Supposons que soient connues $P(c_i)$ et $P(x / c_i)$, pour tout i variant de 1 à C , et supposons que l'on veuille identifier la classe d'un objet inconnu x :

La règle de Bayes consiste à déterminer la probabilité d'appartenance a posteriori de x à chacune des C classes.

$$P(c_i / x) = \frac{P(x / c_i) * P(c_i)}{P(x)}$$

$$P(x) = \sum_{i=1}^C P(x / c_i) * P(c_i)$$

L'objet x sera affecté à la classe pour laquelle $P(c_i / x)$ sera maximum : le classificateur Bayésien est un classificateur MAP (Maximum A Posteriori).

Avantage : Le rôle de la théorie de la décision Bayésienne est de fournir une fonction de décision qui minimise le risque par décision prise. En raison de cet avantage, la classification de Bayes est très utilisée en reconnaissance des formes.

Inconvénients :

- La conception de la méthode repose sur la connaissance des probabilités a priori de chaque classe et du coût de décision.
- Le calcul des probabilités nécessite un très grand nombre d'échantillons ou d'exemples de données, afin d'avoir une bonne approximation.
- Le stockage des probabilités nécessite une grande place mémoire centrale.

V.1.2 Méthode des k-plus proches voisins

La règle du plus proche voisin est souvent utilisée lorsque l'on ne dispose d'aucune connaissance sur la distribution statistique de l'ensemble d'apprentissage que l'on considère.

L'idée de cette règle réside dans le fait que deux individus proches, au sens d'une distance, ont plus de chance d'appartenir à une même classe que deux individus plus éloignés.

Soient $X = \{x_1, x_2, \dots, x_n\}$ de R^n l'ensemble des observations d'apprentissage réparties en C classes, $C = \{c_1, c_2, \dots, c_c\}$, et $d(x,y)$ une distance entre les observations x et y .

Supposons qu'on veuille classer une forme inconnue X , alors on cherche autour de ce point (la forme à reconnaître est un point dans l'espace R^d) les K plus proches voisins déjà affectés d'un numéro de classe. La classe majoritaire parmi les K numéros ainsi déterminés sera attribuée à la forme à reconnaître.

Cette technique présente l'avantage d'être cohérente non paramétrique et simple à mettre en œuvre, mais elle dépend du choix de K et de la distance utilisée. De plus un objet (forme) inconnu est

comparé à chacun des individus (formes déjà classées) de la base. Son unique défaut est d'avoir un temps de calcul proportionnel au nombre de formes d'apprentissage.

V.2 Méthodes structurelles

Les approches précédentes ne permettent pas de prendre en compte l'information structurelle et contextuelle d'une forme. Ces informations sont pourtant importantes dans de nombreux problèmes de reconnaissance de formes.

Les méthodes structurelles s'intéressent à la structure des formes et à leur description en terme d'assemblage de sous formes primitives. On distingue les méthodes syntaxiques et les méthodes descriptives [26].

V.2.1 Les méthodes syntaxiques

Dans ces méthodes, on cherche et on exploite les règles de construction des formes à partir de leurs composantes d'où l'analogie que l'on peut faire avec l'analyse grammaticale ou syntaxique.

Les éléments de base constituent l'alphabet d'un langage. Les lois d'assemblage sont les règles grammaticales dont l'application engendre les formes étudiées qui sont les phrases du langage. La reconnaissance d'une forme inconnue se fait alors par une analyse syntaxique de la phrase qui la décrit.

En effet, à partir d'un ensemble d'éléments primitifs de formes, et d'un ensemble de règles de combinaison de ces composantes, on constitue d'autres composantes intermédiaires qui seront combinées à leur tour avec d'autres éléments. Nous citerons pour ce cas la structure de chaînes.

➤ Structure de chaînes

Une description simple de la structure d'une forme peut être faite en la codant par un ensemble fini. En théorie des langages, un tel ensemble de symboles s'écrit :

$$X = \{a, b, \dots, n\}$$

Où X est appelé alphabet et a, b, ..., n sont les formes élémentaires ou primitives.

Une phrase sur X est définie comme une suite ordonnée d'éléments de X représentée par simple juxtaposition de ces éléments.

$$x = [a_1 a_2 \dots a_p] \quad \text{avec } a_i \in X$$

Exemple

$X = \{a, b, c\}$ est un alphabet

$x = bcaba$ est une phrase de X.

Dans le cas où la forme à étudier a fait l'objet d'une description en chaînes, il faut pour pouvoir la reconnaître être capable de la comparer à des prototypes décrits dans le même formalisme [27].

V.2.2 Les méthodes descriptives

Dans ces méthodes, on retrouve celles dites structures d'arbres ou de graphes. Un graphe est défini par un couple $G = (V, E)$ où V est un ensemble fini de nœuds ou de sommets et E un ensemble de couples ordonnés de sommets appelés arcs.

Exemple

$$V = \{1, 2, 3, 4\}$$

$$E = \{(1, 2), (2, 1), (1, 3), (2, 3), (3, 4), (4, 2)\}$$

Dans l'approche des graphes, on retrouve des graphes de décomposition hiérarchique comme ceux de (ET/OU) où la racine correspond à la forme elle-même et les feuilles aux primitives (voir figure I.6 où l'on reconnaît la forme du « A » à partir de ses éléments de base). Il existe aussi des graphes relationnels attribués où les nœuds représentent les sous-formes et les arcs les relations entre ces formes (voir figure I.7).

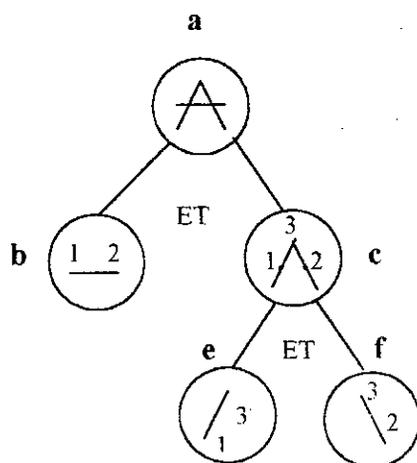


Figure I.6 : Graphe ET/OU

a est reconnu si b et c le sont.

c est reconnu si e et f le sont.

Les chiffres indiquent les points d'attache des composants du caractère « A ».

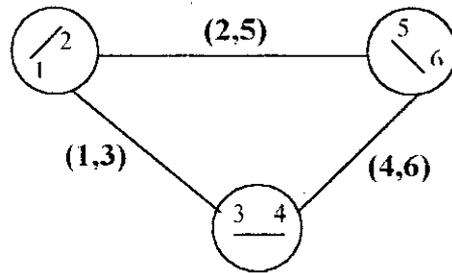


Figure I.7 : Graphe relationnel attribué

Le principe de reconnaissance par cette méthode se décompose en deux étapes :

- Récupération des représentations sous forme d'arbres ou de graphes des formes de référence.
- Comparaison de la représentation de la forme inconnue à celle de la référence en effectuant un calcul de distance entre arbres ou graphes.

V.3 Méthodes connexionistes

Des résultats théoriques montrent que les méthodes connexionistes sont des approximateurs universels de fonction : pour une fonction donnée, il existe un modèle connexioniste capable de l'approximer jusqu'à la précision désirée [28, 29].

Les modèles connexionnistes peuvent être utilisés dans le cas de la classification ou en d'autres termes, de reconnaissance des formes.

Un problème de classification peut être résolu en trouvant une fonction qui associe un ensemble de formes de départ à un ensemble de classes d'arrivée, c'est à dire que le réseau connexionniste jouera le rôle de fonction de transfert telle que : étant donnée une valeur en entrée, on applique la fonction de transfert qui, elle, nous fournit la valeur de sortie.

Bien sûr, dans le cas de la classification automatique on espère que la valeur de sortie corresponde à la classe d'appartenance de la valeur d'entrée.

Les réseaux de neurones mémorisent une forme ou un ensemble de paramètres en ajustant les coefficients entre neurones. Les modèles en couches ont des capacités de classification. Lorsque des neurones de la couche d'entrée sont stimulés par des paramètres ou par une image, une évolution se propage à tout le réseau aboutissant à un état stable fourni en sortie. Un réseau peut posséder plusieurs états de sortie dont chacun correspond à une forme préalablement apprise. Le réseau

revient automatiquement à l'un de ces états lorsqu'il a été stimulé par la forme correspondante, même si celle-ci est altérée ou incomplète.

Il est donc possible de reconnaître des formes très floues ou imprécises pour leur appliquer la programmation classique ou à base de connaissance.

Nous pouvons conclure, d'après ce qui vient d'être dit, qu'aucune des deux approches (statistique ou structurelle), n'est meilleure que l'autre. Certains problèmes se résolvent naturellement mieux à partir d'une approche statistique (ceux où le nombre de mesures est important). Il est donc très avantageux pour un structuraliste d'introduire un peu de décision statistique à un certain niveau de son problème, de même pour un statisticien de tenir compte des propriétés structurelles pour éviter des calculs exhaustifs.

Un modèle connexioniste peut par contre être utilisé avec l'une des deux méthodes et constitue alors une optimisation de ces dernières puisqu'il est capable de les approximer jusqu'à une précision désirée.

VI. Conclusion

L'extraordinaire variété des domaines d'application de la vision par ordinateur explique la vitalité de cette discipline. Dans ce chapitre, nous avons expliqué les différentes étapes d'un système de vision par ordinateur (ou de reconnaissance des formes).

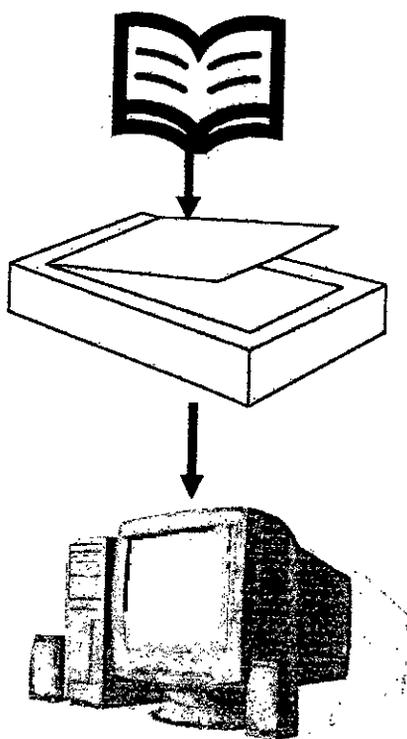
Afin de résoudre un problème de reconnaissance des formes, le concepteur doit donc trouver une procédure d'identification qui réduise la quantité d'informations mais surtout qui coïncide avec les décisions de reconnaissance d'un observateur humain.

En pratique, il s'agit de construire des fonctions de décision, c'est à dire développer des programmes qui à partir de la représentation d'une forme permettent de lui donner automatiquement une identité.

Dans les prochains chapitres, nous étudierons en détail la reconnaissance optique des caractères, objet de notre recherche.

CHAPITRE II

Reconnaissance de l'écriture



- I. Introduction
- II. Historique et état de l'art
- III. Les OCR dans l'ARD
- IV. Les problèmes de l'OCR
- V. Définitions utilisées en OCR
- VI. Les différents aspects de l'OCR
- VII. Les techniques de l'OCR
- VIII. Structure d'un système OCR
- IX. Conclusion

I. Introduction

L'OCR (Optical Character recognition) acronyme américain de la reconnaissance optique des caractères, émerge aujourd'hui dans de nombreuses applications, elle permet des gains de productivités remarquables, qui vont de la reconnaissance de caractères imprimés issus de documents scannés à la reconnaissance de l'écriture manuscrite en passant par la reconnaissance de signature ou de documents télécopiés.

Le but de la reconnaissance de l'écriture est de transformer un document de texte écrit sur un support papier, en une représentation compréhensible par un ordinateur et facilement reproductible par un traitement de texte ; c'est une interface permettant de faciliter la communication homme-machine. C'est aussi un problème d'identification d'une forme donnée qu'on affecte à une classe obtenue par apprentissage. Cette technique n'est pas facile vu que les mots possèdent une multitude de représentations dues aux nombreuses polices de caractères existant avec de nombreux styles (gras, italique, ...).

II. Historique et état de l'art

L'intérêt suscité par la reconnaissance des caractères est justifié par ses nombreuses applications dans divers domaines d'activité.

Des recherches ont été entreprises de part le monde, depuis plusieurs décennies, pour reconnaître l'écriture imprimée, dactylographiée et même manuscrite. Des systèmes très performants arrivent à des taux de reconnaissance très élevés pour l'écriture latine, chinoise, indienne, etc... quant au manuscrit, il pose encore quelques problèmes vu la cursivité des lettres et leur variabilité d'une personne à une autre.

C'est en 1940 qu'apparaissent les premières applications informatiques de l'OCR. Ainsi, en 1975, les japonais utilisaient couramment des lecteurs qui déchiffraient le code postal inscrit à la main ou tapé à la machine. Dans la même période, le français CONTER construisait un système de lecture optique automatique de textes imprimés destinés aux non voyants. Par la suite, la compagnie américaine KURZWEIL a réussi à améliorer le système précédent en proposant des machines à lire pour aveugles formulant le texte à haute voix par synthèse vocale.

L'écriture Arabe, objet de notre étude, a bénéficié de quelques recherches. C'est en 1980, à l'université de Nancy, que A.Amin a (parmi les premiers chercheurs) débuté ses travaux sur les OCR arabes et réalisa le système IRAC (Interactive Recognition of Arabic Characters) [30].

Depuis, plusieurs recherches ont succédé sur l'écriture arabe en France[30,46,56], en Syrie[41,51], en Egypte[36,37], en Algérie[4,8], en Tunisie[103]... utilisant différentes approches.

Les différentes approches utilisées en reconnaissance de caractères peuvent entrer dans la catégorie on-line ou off-line, chacune ayant ses propres matériels et algorithmes d'identification.

Dans les systèmes de reconnaissance de caractères on-line, l'ordinateur identifie les symboles pendant qu'ils sont dessinés. La surface d'écriture la plus commune est la tablette graphique, qui fonctionne par l'intermédiaire d'un stylo spécial en contact avec la surface de la tablette (numériseur) et émet les coordonnées des points tracés à une fréquence constante. La rupture de contact incite la transmission d'un caractère spécial. Ainsi, l'enregistrement sur la tablette graphique produit une suite de coordonnées séparées par des signes indiquant quand le stylo a cessé de toucher la surface de la tablette numérique.

La reconnaissance on-line a plusieurs caractéristiques intéressantes. La première est que l'identification est exécutée sur des données unidimensionnelles plutôt que des images bidimensionnelles comme dans le cas de l'identification off-line. La ligne d'écriture est représentée par une séquence de points dont l'adresse est une fonction du temps.

Plusieurs méthodes sont utilisées [31, 32, 33, 34, 35] dans les systèmes on-line de reconnaissance des caractères arabes manuscrits.

La reconnaissance off-line est réalisée après que l'écriture ou l'impression soit accomplie. Dans un système OCR typique, les caractères entrés sont lus et digitalisés par un scanner. Chaque caractère est alors localisé et segmenté et la matrice résultante est introduite dans le module de pré traitement pour effectuer un éventuel filtrage, réduction de bruit, et une normalisation de la taille si ceci est nécessaire. La reconnaissance off-line peut être considérée comme le cas le plus général: aucun dispositif spécial n'est exigé pour l'écriture et l'interprétation du signal est indépendante de sa génération, comme dans la reconnaissance humaine.

Plusieurs articles traitent de la reconnaissance des caractères latins [82,140], chinois et japonais [34,35]. Cependant, bien que presque un tiers de milliard de personnes dans le monde entier, dans plusieurs langues différentes, emploient les caractères arabes pour l'écriture, peu de progrès de recherches, que ce soit en on-line ou en off-line, ont été réalisés en reconnaissance automatique des caractères arabes. Ceci résulte du manque d'appui et naturellement de la nature cursive de ses règles d'écriture.

Deux approches ont été appliquées à la reconnaissance de caractères arabes imprimés et manuscrits. Celles-ci peuvent être classées par catégorie comme suit:

1- l'approche analytique dans laquelle les mots sont segmentés en caractères (ou pseudo caractères) [8, 36, 37, 38, 39, 40, 41, 42, 43, 44,45]. Ces caractères sont alors envoyés à un système de reconnaissance de caractères et la combinaison des résultats d'identification produit une liste rangée de mots possibles. Cette approche est théoriquement plus forte en manipulant un grand vocabulaire. En effet avec un nombre constant de classes (par exemple le nombre de lettres dans l'alphabet), on peut manipuler n'importe quelle suite de caractères et donc un nombre illimité de mots.

2- L'approche globale dans laquelle l'identification considère des mots et ne tente pas d'identifier des caractères individuellement. Cette approche a un avantage qui est celui d'éviter l'étape difficile de segmentation. Cette approche a été, à l'origine, présentée pour la reconnaissance de la parole et peut exister dans deux catégories :

- a) Méthodes basées sur des mesures de distance et utilisant la programmation dynamique [46, 47].
- b) Méthodes basées dans un cadre probabiliste (modèles cachés de Markov) [48, 49,50].

D'ailleurs, pendant les deux dernières décennies, une variété d'approches a été proposée, comprenant les méthodes statistiques [51, 52,53], les méthodes structurelles et syntaxiques [54, 55, 56] ainsi que les réseaux de neurones [57, 44, 58,59].

Le texte arabe (imprimé ou manuscrit) est en général cursif et les lettres arabes sont normalement reliées sur la ligne de base. Cette connectivité s'avérera un problème important dans le procédé de segmentation. Dans la majorité des écritures les textes imprimés ne sont pas cursifs, mais la plupart des textes arabes le sont, et de ce fait il n'est pas étonnant que le taux de reconnaissance des caractères arabes soit bas par rapport aux caractères non connectés tels que le latin imprimé.

L'écriture arabe est semblable au latin parce qu'elle emploie des lettres, numéros, signes de ponctuation, ainsi que les espaces et des symboles spéciaux. Elle diffère cependant du Latin dans sa représentation des voyelles puisque l'Arabe utilise diverses inscriptions diacritiques. La présence et l'absence de voyelles diacritiques indiquent différentes significations dans ce qui serait autrement le même mot. Par exemple, مدرسة est le mot arabe pour l'" école " et la" maîtresse ". Si

le mot est isolé, il est essentiel de mettre les diacritiques pour distinguer entre les deux significations possibles. S'il se trouve dans une phrase, l'information contextuelle inhérente à la phrase peut être employée pour associer la signification appropriée.

L'alphabet arabe se compose de 29 lettres. Cependant, la forme d'un caractère dépend de sa position dans le mot, de ce fait le nombre de formes de caractères augmente de 29 à environ 60. L'Arabe est écrit de droite à gauche. Chaque caractère, dans un mot, peut être relié à gauche et/ou à droite par un point de raccordement qui, habituellement, se trouve sur une ligne imaginaire, qu'on appelle ligne de base.

Des types d'écritures arabes peuvent, en général, être classifiés dans l'imprimé (Naskhi), le manuscrit (Ruq'a) et l'artistique (ou la calligraphie décorative : Kufi, Diwani, royal et Thuluth) [60]. Le manuscrit ainsi que les styles décoratifs présentent habituellement des combinaisons verticales des caractères appelés les ligatures. Ceci rend difficile la détermination des frontières des caractères.

L'alphabet arabe est représenté numériquement par un code standard d'échange de communication approuvé par l'organisation arabe de norme et de métrologie (ASMO : Arab Standard and Metrology Organization). Comme dans le code ASCII, chaque caractère dans le code ASMO est représenté par un byte. Une lettre latine a deux formes possibles : Majuscule et minuscule ; le code ASCII fournit des représentations différentes pour les deux formes, tandis que les lettres arabes ont seulement une représentation dans la table ASMO. Ceci ne signifie pas, cependant, que la lettre arabe a seulement une forme. Au contraire, une lettre arabe pourrait avoir jusqu'à quatre formes différentes, selon sa position dans le mot. Par exemple, la lettre (ع: AIN) a quatre formes différentes: au début du mot (précédé par un espace), au milieu du mot (aucun espace autour), à la fin du mot (suivi d'un espace), et isolé (précédé par une lettre non liée et suivi d'un espace).

En outre, certains caractères arabes différents peuvent avoir exactement la même forme et se distinguent les uns des autres par seulement l'ajout de caractères complémentaires. Ces caractères complémentaires (une partie d'un caractère qui est nécessaire pour compléter un caractère arabe) peuvent être un point, un groupe de points ou une hamza (celle-ci peut apparaître sur, au-dessus de ou au-dessous de la ligne de base) et positionnés selon le cas, au-dessus, au-dessous ou à l'intérieur du caractère. Il est à noter que n'importe quelle érosion ou suppression de ces caractères complémentaires a comme conséquence une mauvaise représentation (fausse déclaration) du caractère. Par conséquent, n'importe quel algorithme amincissant (filtrage ou squelettisation) doit prendre en considération ces points pour ne pas changer l'identité du caractère.

L'écriture arabe est cursive et est telle que des mots sont séparés par des espaces. Cependant, un mot peut être divisé en plus petites unités appelées les sous mots (ou parties connexes comprenant un ou plusieurs caractères reliés). Certains caractères arabes ne peuvent pas être reliés avec les

caractères suivants. Par conséquent, si un de ces caractères existe dans un mot, il divise ce mot en deux sous mots. La figure II.1 montre trois mots arabes avec un, deux, et trois sous mots. Le premier mot se compose d'un sous mot, qui a quatre lettres ; le second a deux sous mots avec deux et une lettres respectivement. Le dernier mot contient trois sous mots, chacun se composant de seulement une lettre.

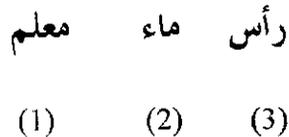


Figure II.1 : mots arabes et leurs différentes parties connexes (ou sous mots)

Parmi les applications de l'OCR nous pouvons citer :

- L'assistance et l'éducation pour la reconnaissance et la traduction de textes en braille, et l'apprentissage de l'écriture (lecture et écriture). Des photo senseurs et des simulateurs tactiques sont utilisés pour les non-voyants avec une sortie sonore [61, 62,63] ;
- Les banques et les assurances pour l'authentification des chèques pour les banques (correspondance entre montants et libellé d'une part, et entre l'identité du signataire et sa signature [64,65], d'autre part) [66], et la vérification des clauses de contrats pour les assurances [67] ;
- La poste pour la lecture des adresses et le tri automatique du courrier [68,69]
- Les télécommunications pour l'échange de fichiers informatisés à distance [70,71] ;
- La police et la sécurité pour la reconnaissance de numéros minéralogiques [72] pour le contrôle routier, l'authentification de manuscrits et l'identité du scripteur [73] ;
- La bureautique pour l'indexation et l'archivage automatique de documents, et pour la publication électronique [74,75] ;
- L'administration pour la reconnaissance de plans cartographiques [76] et la lecture automatique de documents administratifs [77]
- Les affaires et l'industrie pour la gestion des stocks et la reconnaissance de documents techniques [78].

III. Les Systèmes OCR dans l'Analyse de la Structure de Documents

Les connaissances quelles soient scientifiques, techniques, historiques économiques, juridiques, médicales, etc. sont en majorité mémorisées et véhiculées par le papier.

L'avènement actuel des réseaux et la dématérialisation de l'information, qui devient électronique, sont en train de transformer radicalement notre monde par une diffusion plus large de ces connaissances.

Les informations publiées récemment sont directement accessibles sous une forme codée, par contre, la majorité du patrimoine culturel et technique de l'humanité n'est disponible que sous forme de documents papier [79, 80]. Les entreprises et les institutions sont ainsi confrontées aujourd'hui à un besoin énorme en reprise de ces documents papier, pour les transformer en un format électronique codé. Ce constat, en plus du défi de faire lire l'ordinateur comme un être humain, a fait apparaître l'un des plus récents axes de recherche : « Analyse et Reconnaissance de Documents » (ARD). Cette tâche complexe recouvre plusieurs opérations difficiles à réaliser, dont la segmentation en blocs et la reconnaissance de caractères ne sont que des éléments importants.

La segmentation en blocs est une étape indispensable dans tout système d'ARD [1, 2, 81, 82, 83,84] (voir figure II.2). En effet pour le codage ou la reconnaissance de documents, il est essentiel d'identifier les régions (ou blocs) typographiquement homogènes (texte, graphe, ...) afin d'effectuer le traitement adéquat au type de bloc identifié. A titre d'exemple : on doit identifier les zones de texte avant d'appliquer les algorithmes de la Reconnaissance de Caractères (OCR), et les zones graphiques avant d'essayer de les interpréter ou de les vectoriser.

Les avantages d'une segmentation intelligente de la page peuvent se résumer dans les points suivants :

- Améliorer le taux de compression, par l'utilisation de codeurs spécifiques, par exemple : OCR pour les blocs texte, DCT pour les blocs image... ou bien utiliser une quantification vectorielle (VQ) avec des dictionnaires conçus pour chaque type de bloc.
- Améliorer la vitesse de transmission des blocs sélectionnés.
- Guider la reconnaissance de la structure logique.
- Permettre un archivage facilement et rapidement accessible avec un gain considérable en encombrement mémoire.
- Aider la conversion du document en HTML (Hyper-Text Markup Language) [85] ce qui le rend accessible dans le WWW (World Wide Web).

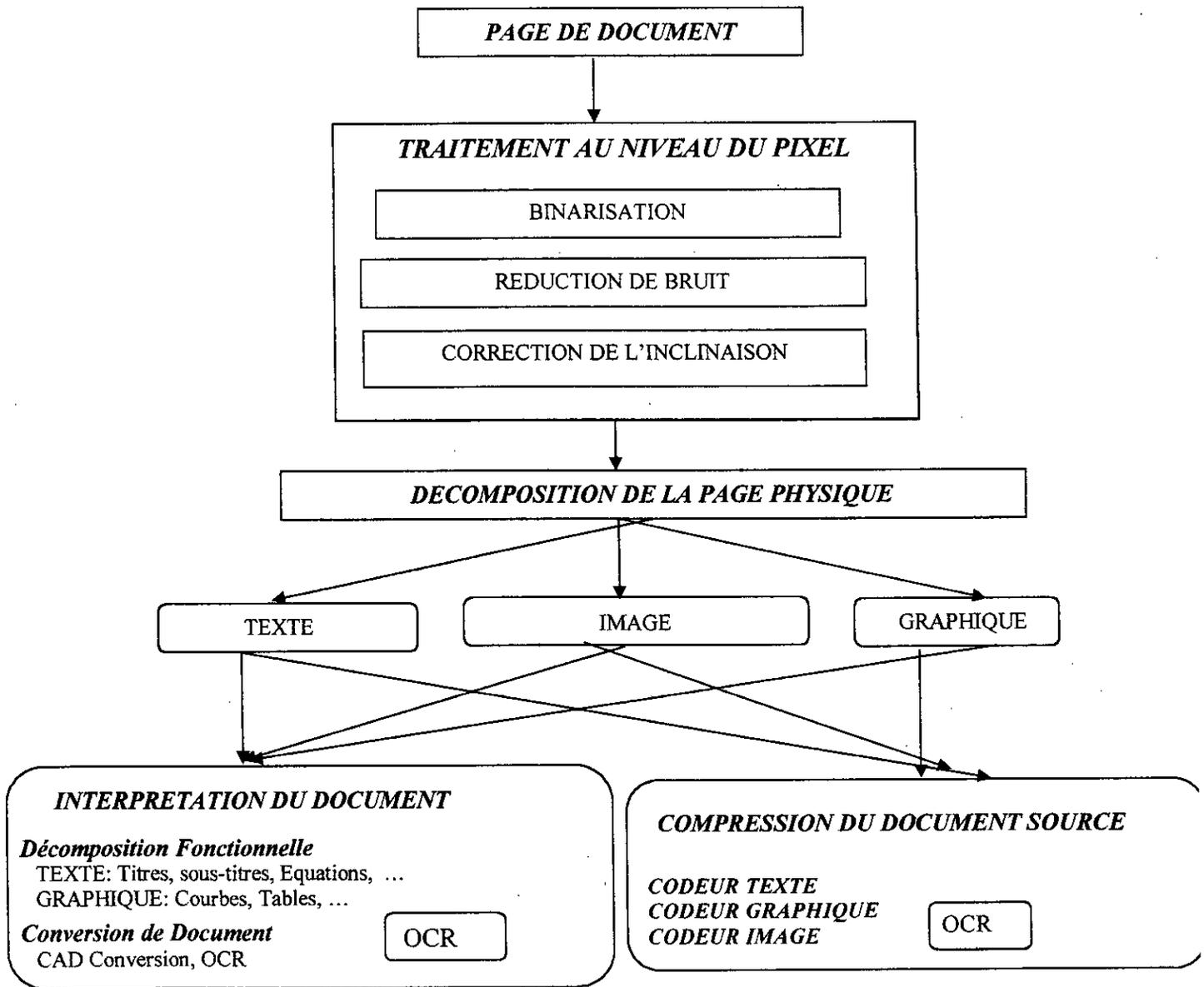


Figure II.2 : Place de l'OCR dans "Document Image Processing"

III.1 Normes et Structuration du Document

Pour réaliser un système ARD (analyse et reconnaissance de documents) complet il est nécessaire de disposer de modèle de structures de documents pour guider la reconnaissance et améliorer les résultats d'analyse.

Les problèmes de modélisation sont intimement liés à la catégorie du document à traiter : composite, postal, formulaire, etc.

Des systèmes de reconnaissance, utilisant des connaissances sur les structures de documents comme modèles a priori, ont été développés dans plusieurs domaines. On peut citer : Ingold [86] en reconnaissance de documents légaux, Dengel [87] en interprétation de lettres d'affaires, Belaïd [88] en reconnaissance de notices bibliographiques et Watanabe [89] en compréhension des formulaires.

III.1.1 Structuration des documents

En analyse de documents, la structure sert comme modèle de repérage de zones informatives. On distingue deux niveaux de structuration d'un document [1, 81, 82, 84] :

- La structure logique,
- La structure physique.

➤ Structure logique

La structure logique organise un document en terme logique et sous forme hiérarchique. Elle définit une représentation abstraite du document qui décrit l'organisation du document telle qu'elle est perçue par l'auteur. Les objets logiques peuvent être : le titre, le résumé, la section, etc.

➤ Structure physique

Elle résulte de la mise en page du document, et représente le premier niveau de données accessibles au système de lecture optique. Les objets physiques peuvent être : la page, la colonne, le bloc texte, etc. La figure II.3 montre un exemple d'une lettre d'affaire permettant d'illustrer la structure physique, la structure logique et leurs relations avec le contenu de la lettre [82].

III.1.2 Normes de représentation des documents

Dans le domaine de la bureautique, la question qui se posait était : comment échanger des documents bureautiques entre différents systèmes. Ce problème est dû à des incompatibilités entre les matériels et les logiciels utilisés dans les bureaux. Pour résoudre ce problème, l'ISO (International Standard Organization), l'ECMA (European Computer Manufacturers Association) et le CCITT (Consultation Committee on International Telephone and Telegraph) ont mené des études qui ont abouti à la mise en place d'un standard de communication d'information :

ODA/ODIF (Office Document Architecture/Office Document Interchange Format).

Pour l'édition, le problème ne concernait pas l'échange du contenu des documents mais aussi de la structure associée. La solution à ce problème est le développement des langages de balisage.

L'idée principale de ces langages consiste à séparer les informations relatives à la composition et à la mise en page d'un document de son contenu. Ces informations sont mises dans un fichier à part. Avec un tel procédé, un auteur peut se concentrer sur le contenu et la structure de son document, sans se préoccuper des détails de représentation. En conséquence, la production de l'information est devenue moins chère, l'information plus accessible et la mise à jour davantage pouvant être effectuée par le grand public.

➤ **La norme ODA**

Un document ODA est décrit par trois entités :

1. Ses éléments constitutifs ``Content Fragment `` :

Ce sont les primitives ou éléments simples du document. Ils sont de trois types distincts :

- Caractères,
- Éléments géométriques,
- Éléments photographiques.

Ils sont définis par les recommandations ISO-8859 pour les caractères et ISO-8632 pour les graphiques.

2. Sa structure Physique (Layout Structure) :

Elle décrit le document selon des critères morphologiques. Au niveau logique, la norme ODA précise cinq types de classes d'objets :

- Le document,
- L'ensemble de pages (set pages),
- La page,
- La région (Frame),
- Le bloc.

3. Sa structure logique (Logical Structure) :

Elle rend compte de l'organisation interne du document et fait appel à des concepts relevant d'une analyse intelligente de celui-ci.

Au niveau logique, la norme ODA distingue trois types de classes d'objets :

- Le document

- Les objets logiques (comme les chapitres, les sections,...)
- Les objets de base (comme les paragraphes).

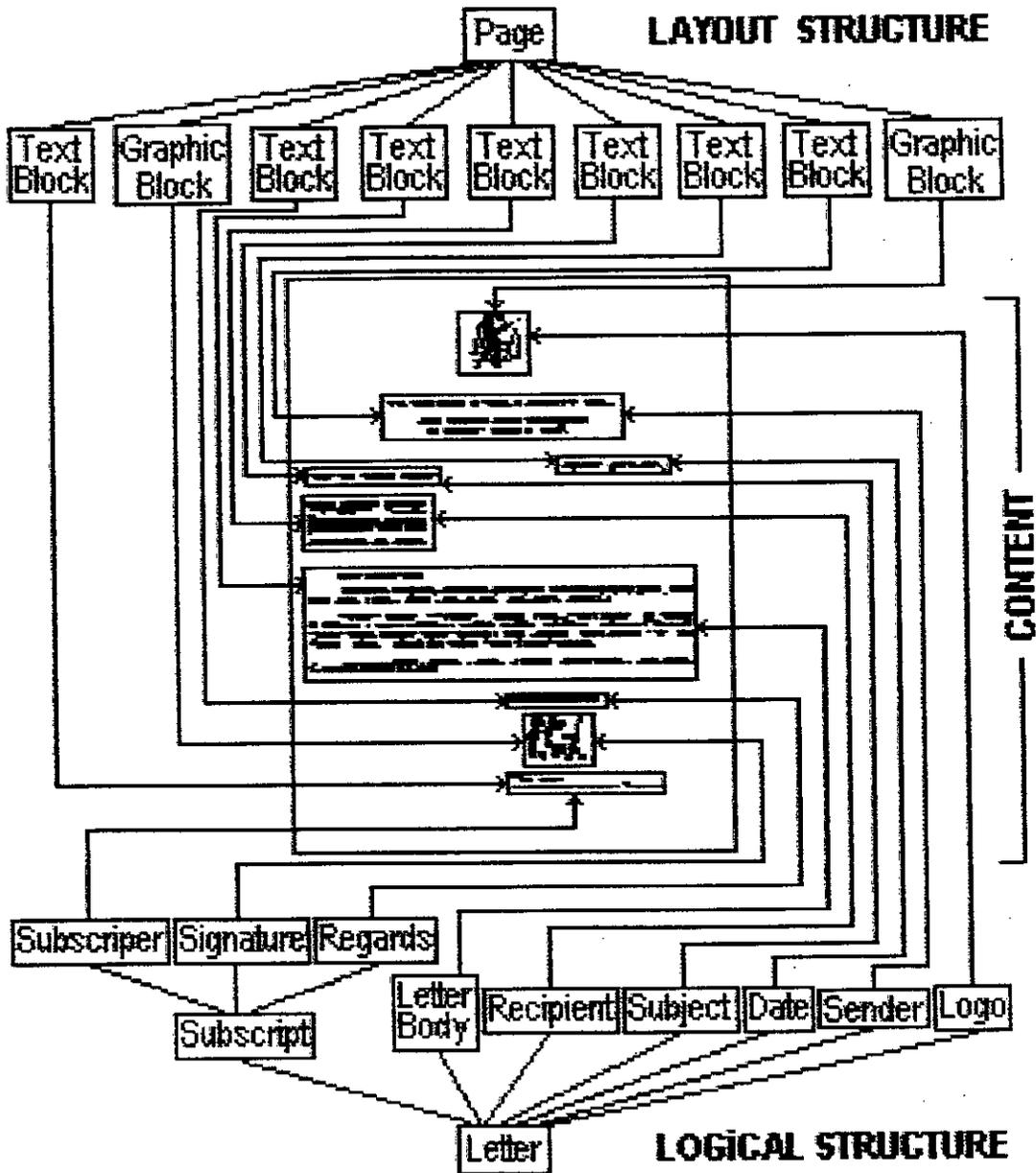


Figure II.3 : Relation entre les structures physique et logique avec le contenu d'une lettre d'affaire

La figure II.4 montre les objets génériques logiques et physiques définis dans la norme ODA.

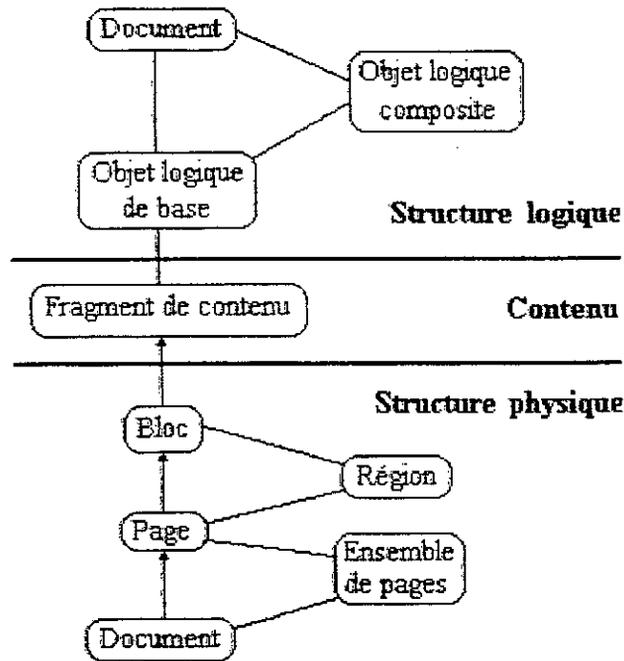


Figure II.4 : Objets génériques de la norme ODA.

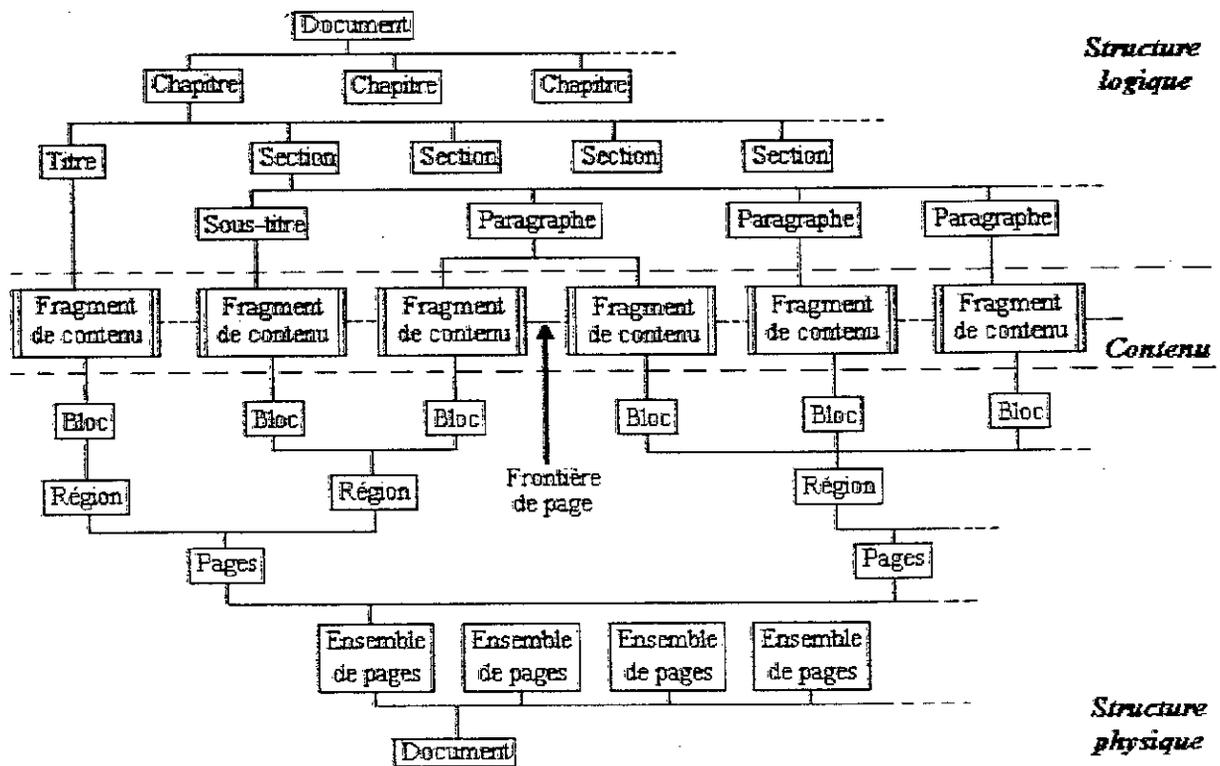


Figure II.5 : Relations entre les structures logique et physique dans un document ODA.

Les structures physique et logique ne se recouvrent pas, leurs éléments terminaux ne désignent pas les mêmes éléments constitutifs. Ainsi un chapitre, qui constitue un objet logique, peut être physiquement réparti en plusieurs pages, chaque page étant un objet physique distinct. La figure II.5 montre comment les éléments constitutifs d'un document ODA sont atteints par les deux structures physique et logique.

ODIF : Créé parallèlement à ODA, il rassemble les directives nécessaires au formatage de l'échange de documents électroniques de bureau. Il propose une représentation indépendante de la façon dont les documents ODA ont été codés par le système qui les a engendré et permet le transfert des structures et des contenus des documents vers d'autres systèmes.

➤ **La norme SGML**

Un document SGML (Standard Generalized Markup Language) est une chaîne de caractères constituant du texte de document placé entre les commandes de marquage (qui identifient les début et fin des entités logiques).

La norme SGML fournit un jeu de délimiteurs pour effectuer le codage et le balisage des structures et le contenu de document (tableau II.1.).

<i>Code</i>	<i>Usage</i>
<	Ouverture de balise de début
</	Ouverture de balise de fin
<!	Ouverture de déclaration de balisage
<?	Ouverture d'instruction de traitement
>	Fermeture de balise
--	Début et fin de commentaire
#	Indicateur de nom réservé
&	Ouverture d'une référence à une entité
&#	Ouverture d'une référence à un caractère
%	Ouverture d'une référence à un paramètre
;	Fermeture d'une référence
(Ouverture de groupe
)	Fermeture de groupe
=	Indicateur de valeur

Tableau II.1 : Extrait du jeu de délimiteurs dans SGML.

Chaque élément d'un document SGML commence par une marque de début « start-tag » et se termine par une marque de fin « end-tag ». Par exemple une description SGML d'un paragraphe s'effectue comme suit :

```
< Paragraphe > .....  
                .....  
< / Paragraphe >
```

} **LE CONTENU**
du paragraphe

Dans ce paragraphe, nous avons donné un bref aperçu sur la structuration et les normes de représentation des documents .

Le choix entre la norme ODA et la norme SGML n'est pas très important car l'essentiel en ARD est de pouvoir exprimer les objets recherchés et leurs relations. L'essentiel est de pouvoir produire le résultat dans un formalisme le plus proche possible d'une norme de manière à pouvoir bénéficier des logiciels associés pour la réutilisation du document.

IV. Les problèmes de l'OCR

La reconnaissance optique de caractères consiste à récupérer automatiquement des textes imprimés ou manuscrits en évitant la phase pénible de la saisie par clavier.

La transformation d'un texte imprimé en code ASCII se déroule en deux phases. La première se résume à numériser la page entière et la deuxième phase consiste, à partir d'un fichier graphique (BMP, PCX, TIF,...), à distinguer des formes que l'on identifie comme des lettres de l'alphabet.

Lors de la reconnaissance, on distingue quatre difficultés majeures : [1, 2]

- Distinguer le texte des autres éléments graphiques.
- Reconstituer l'ordre du texte.
- Reconnaître les caractères spécifiques qui varient selon les langues.
- Identifier les caractères quelque soit leur taille, leur épaisseur et la police utilisée.

Suivant le type de document, on peut avoir d'autres problèmes tels que le vieillissement du papier et surtout distinguer les textes des photographies dans les magazines. La reconnaissance des caractères constitue la phase la plus délicate.

V. Quelques définitions utilisées en OCR

1. Primitive

On appelle primitive une forme élémentaire constituée par une agglomération de pixels pertinents à laquelle il est possible de donner une interprétation [90]

2. Propriétés topologiques et géométriques

a) Propriétés topologiques

Elles correspondent à la définition des parties de la forme comme les cavités, les boucles et le nombre de parties composant la forme.

b) Propriétés géométriques

Elles correspondent aux distances entre parties, aux concavités, convexités, allongements, rapport de taille,...

3. Représentation

C'est la chaîne d'éléments d'un alphabet fini permettant de représenter (de coder) une information.

4. Identification

Acte de donner un nom à une représentation, également le résultat de cette action.

5. Définitions typologiques

a) Fonte

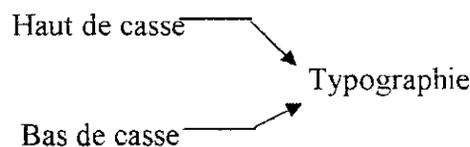
Une fonte est un ensemble de caractères d'une police particulière à une taille particulière.

b) Police

Une police indique le style ou le graphisme des caractères et leurs attributs (gras, italique, souligné, ombré,...)

c) Mot

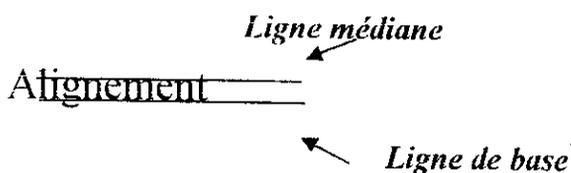
Un mot est une juxtaposition de lettres de type bas de casse ou haut de casse.



d) Alignement

Tous les caractères d'une même fonte s'alignent en pied des lettres courtes : a, e,... Cette ligne est appelée **ligne de base** et sert de référence pour les calculs d'interligne et d'inclinaison. La **ligne**

médiane est la ligne passant par le haut des lettres courtes et constitue avec la ligne de base une zone de forte densité de points appelée **bande centrale** [91].



VI. Les différents aspects de l'OCR

Il n'existe pas de système universel d'OCR, mais plutôt des voies d'approche dépendant du type de données traitées et bien sûr de l'application visée.

Dans le cas de la reconnaissance des caractères imprimés, ces derniers sont bien alignés et souvent bien séparés verticalement, ce qui simplifie la phase de lecture. De plus, le graphisme des caractères est conforme à un style calligraphique (une fonte) ce qui constitue un modèle pour l'identification.

VI.1 Reconnaissance monofonte, multifonte et omnifonte

Pour un texte imprimé, un système est dit monofonte s'il ne traite qu'une seule fonte à la fois. Dans ce cas, l'apprentissage est simple à réaliser et le taux de reconnaissance est très élevé.

Un système est dit multifonte s'il est capable de reconnaître un mélange de quelques fontes, parmi un ensemble de fontes préalablement apprises. L'apprentissage doit gérer les ambiguïtés dues à la ressemblance des caractères des différentes fontes. Enfin, un système est appelé omnifonte s'il permet de reconnaître toute fonte sans l'avoir absolument apprise [91].

VI.2 Reconnaissance On-line (en ligne ou en temps réel)

La reconnaissance on-line est une reconnaissance dynamique qui se déroule pendant l'écriture. Dans ce cas, on utilise la tablette graphique pour l'acquisition du document.

VI.3 Reconnaissance Off-line (hors ligne ou en différé)

Dans ce cas, la reconnaissance démarre après l'acquisition du document entré. Ce mode permet l'acquisition d'un nombre important de documents et nécessite un traitement complexe pour retrouver l'ordre de l'écriture.

VII. Les techniques de l'OCR

VII.1 Méthode de masquage

Le principe de cette méthode est basé sur l'examen d'un certain nombre de pixels dans l'image du caractère et l'étude de leur appartenance. Une forme est reconnue si l'ensemble de ces pixels sont présents. Cependant à cause de la rigidité de ces méthodes (conçues à partir d'emplacement fixe de pixels) chaque masque de la taille du motif recherché est déplacé sur toute la région où le motif est supposé être et l'on cherche le maximum de la corrélation. Cette méthode est la plus ancienne et n'est employée de nos jours que pour des produits spécifiques (banques, poste) ; elle présente l'avantage d'être très rapide, relativement fiable et surtout simple à mettre en œuvre, mais elle correspond à une série de caractères bien particuliers dans une police prédéfinie avec une taille fixe.

VII.2 Méthodes structurelles

Elles sont fondées sur la structure propre du caractère. La structure est exprimée en terme de composants primitifs correspondant à des formes élémentaires telles qu'un rebroussement, un changement d'orientation, un croisement, un accroissement ou un décroissement de la pente, etc. Ces composantes sont appelées primitives. Parmi ces méthodes, nous pouvons citer :

a) *Méthodes de tests*

Elles consistent à appliquer sur chaque caractère entré des tests de plus en plus fins sur l'absence ou la présence de ces primitives.

Le processus le plus répandu ou habituel consiste à diviser, à chaque test, l'ensemble des choix en deux jusqu'à n'obtenir qu'une seule forme correspondant au caractère entré [92].

b) *Comparaison de chaînes*

Les caractères sont représentés par des chaînes de primitives. La comparaison du caractère test avec un modèle de référence consiste à mesurer la ressemblance entre les deux chaînes en calculant des distances.

c) *Méthodes syntaxiques*

Chaque caractère est représenté par une phrase dans un langage où le vocabulaire est constitué des primitives. Les caractères d'une même famille sont représentés par une grammaire. La reconnaissance consiste à déterminer si la phrase de description du caractère peut être générée par la même grammaire. On définit la distance d'une phrase à une grammaire comme étant la distance minimale de modification à faire subir à la phrase pour qu'elle soit acceptée par la grammaire [93].

VII.3 Méthodes heuristiques

Les méthodes heuristiques sont des procédures basées sur l'intuition et l'expérience de l'automaticien. Les performances et la structure de ces systèmes en dépendront. Cette approche présente les avantages suivants :

- Elle peut simplifier la résolution de problèmes complexes.
- Elle est souple et peut s'adapter à divers types de problèmes.

Les règles et procédures utilisées dans cette technique sont spécifiques au problème à résoudre et ne peuvent donc pas être généralisées à d'autres.

VII.4 Méthode des experts

Chaque programme expert est destiné à un caractère. L'avantage de cette technique est la rapidité ; dès qu'un caractère est identifié, on passe à l'autre. Le seul problème de cette méthode est la complexité de programmation d'un expert.

Un programme expert est un logiciel issu de recherches en intelligence artificielle (IA) destiné à remplacer l'être humain dans toute tâche d'expertise sur un domaine ou un problème spécifique en tentant de reproduire le raisonnement d'un ou plusieurs experts humains.

VIII. Structure Générale d'un système OCR

Un système OCR consiste à associer à une technique de captation de l'image par procédé optique (caméra ou scanner), une technique de reconnaissance de caractères.

Un document se trouvant au départ sur un support papier, et après passage par un système OCR doit être reconstitué après avoir été segmenté.

La perception chez l'homme est le fruit d'un ensemble de processus parallèles et complexes, d'analyse et de traitement de l'information lumineuse par les voies visuelles d'une part, et par certaines parties spécialisées d'autre part ; par analogie au système visuel humain, le capteur physique d'un système OCR correspond à l'œil humain, alors que le processus et les algorithmes de traitement correspondent aux parties spécialisées du cerveau dédiées à la vision.

VIII.1 Schéma fonctionnel d'un système OCR :

Pour échapper au problème de complexité de calcul connu en reconnaissance de formes le système OCR doit être d'une structure modulaire où chaque module (étape), remplit une fonction particulière (voir figure II.6).

De manière générale, un système de reconnaissance est constitué de :

- Un module d'acquisition et de pré traitement
- Un module de segmentation
- Un module de reconnaissance et décision.

VIII.2 Module d'acquisition

L'acquisition est l'opération qui permet de capter les informations du monde extérieur auquel le système est destiné, c'est donc l'interface entre le système et l'environnement extérieur ; il n'a aucun pouvoir décisionnel mais il peut avoir des conséquences considérables sur les traitements ultérieurs : les structures des autres modules dépendent étroitement du capteur utilisé ainsi que de la nature de l'information qu'il délivre. En effet, l'opération d'acquisition permet le passage de l'information du monde réel au monde numérique de l'ordinateur. Ce passage nommé aussi numérisation ou digitalisation, n'est autre que la transformation d'un signal analogique délivré à travers un dispositif photosensible en un signal numérique.

L'acquisition de l'image passe par deux étapes, une matérielle qui permet de numériser l'image et l'autre logicielle qui permet de la visualiser de manière à pouvoir suivre les différents traitements qu'elle subit.

a) Acquisition physique

Le système d'acquisition, le scanner en l'occurrence, est relié à l'ordinateur et est piloté par un logiciel. Ce dernier permet de digitaliser l'information analogique et stocke les données numériques dans un fichier graphique. Cette étape de l'acquisition est dite physique.

b) Acquisition logique

Le deuxième niveau d'acquisition, dit logique, consiste à exploiter le fichier graphique pour en extraire l'image. Elle se présente alors sous forme d'un tableau M à n lignes et p colonnes. Chaque élément de la matrice $M(i,j)$ représente un pixel de l'image dont l'intensité est une valeur numérique couramment appelée niveau de gris. Celle-ci est en général égale à zéro pour le noir et N pour le blanc (N est égal à 255 pour un pixel codé sur un octet).

Pour des images binaires, ce qui est le cas des textes que nous traitons, chaque pixel est codé sur un bit, soit « 0 » pour le noir et « 1 » pour le blanc. De ce fait, l'espace mémoire requis pour stocker une image binaire est beaucoup plus faible que celui d'une image à plusieurs niveaux de gris.

L'image ainsi obtenue peut être stockée sur disque dur sous forme de fichier, ou sur la mémoire RAM, sous forme de matrice, ce qui réduit considérablement le temps d'accès aux données par rapport à l'accès au disque dur qui est relativement lent.

c) Fichiers graphiques

Plusieurs fichiers graphiques sont disponibles sur PC. La différence entre un format et un autre réside dans l'organisation de la structure, l'emplacement de l'image au sein du fichier, sa taille, etc.

L'information est inscrite dans un fichier selon des règles qui régissent la structure de chaque format. La structure peut être elle-même décrite au sein du fichier. Certains formats sont plus flexibles que d'autres. Cependant cette flexibilité est payée au prix d'une forme plus complexe.

L'accès à un fichier graphique nécessite la connaissance au préalable de sa structure, car la lecture suit les mêmes règles que l'écriture. Les principaux formats utilisés sur PC sont : BMP, TIF, GIF, PCX, TGA, JPEG...

Nous donnerons en annexe la structure du fichier TIF que nous utilisons dans notre application.

VIII.3 Prétraitement

Lorsque l'acquisition est réalisée, la plupart des systèmes OCR comportent une étape de prétraitement. Généralement, ces prétraitements ne sont pas spécifiques de la reconnaissance de textes, mais sont des prétraitements classiques en traitement d'images.

Le prétraitement peut comprendre des dizaines d'opérations différentes effectuées sur l'image numérisée. Cette dernière peut présenter des distorsions ou des perturbations provoquées par divers facteurs, et qui sont principalement l'outil d'acquisition (qualité du digitaliseur) et l'état du document (vieillesse, froissement...). A cette phase de l'analyse, le but est de préparer les étapes ultérieures de la reconnaissance. Parmi ces opérations, on peut citer par exemple le problème de la réduction de bruit d'acquisition ou celui de l'élimination du fond dans le cas de la lecture d'un texte écrit sur un fond texturé (papier tramé, chèques, etc.)

On inclut aussi, généralement dans les prétraitements des opérations de type redressement des écritures penchées, détection et éventuellement redressement des lignes de base, etc.[94]. Dans des images à plusieurs niveaux de gris, on peut recourir à une binarisation qui est considérée comme un traitement préliminaire, et ce, tant pour la facilité de manipulation d'images binaires que pour le faible espace mémoire requis pour les stocker. Le principal problème de la binarisation réside dans le choix d'un seuil et dans la détermination de la zone de validité de ce seuil de décision. Si la zone de validité représente l'image entière, le seuil est dit « global » ; dans ce cas, très courant en

pratique, le risque que l'image comporte des zones pour lesquelles ce choix est mal adapté est grand. Si la zone de validité est locale, les techniques de binarisation sont alors à seuil adaptatif [91].

L'amélioration des images consiste en un ensemble de méthodes qui visent à atténuer les effets indésirables. Deux principales approches ont été envisagées [22] :

- Transformations ponctuelles : pour lesquelles chaque pixel est transformé indépendamment des pixels voisins.
- Filtrage : pour lequel la valeur de chaque pixel est évaluée par une combinaison des pixels voisins.

Pour éliminer les redondances dans un document, on peut le soumettre à un traitement préliminaire appelé « squelettisation ». Celle-ci est une technique de traitement d'images qui a pour but de simplifier l'image afin de la rendre plus facile à traiter. Cette opération s'accompagne toujours d'une perte d'informations ; pour remédier à ce problème, les différents algorithmes de squelettisation tentent de conserver les propriétés topologiques et métriques de la forme liées à la courbure, à l'orientation et à l'échelle.

Principes de la squelettisation :

Les critères retenus pour les méthodes de squelettisation sont :

- L'épaisseur du squelette doit être de 1 pixel.
- Le squelette doit conserver les propriétés topologiques de la forme comme le nombre de parties, le nombre de trous et la connexité.
- Le squelette doit respecter les propriétés métriques de la forme comme la longueur totale et la distance entre les parties de la forme.

Il existe deux grandes classes d'algorithmes de squelettisation :

- 1) les algorithmes à critères topologiques qui sont itératifs et qui suppriment à chaque étape le long de la frontière les points appelés inessentiels (c'est à dire qui n'appartiennent pas au squelette); ils opèrent sur des voisinages de points, déterminant à chaque passage la validité des points frontières. Parmi ces algorithmes nous pouvons citer : l'algorithme de SERRA, celui de THOME et celui de ZHANG et SUEN que nous avons utilisé dans notre système et que nous détaillerons plus loin.
- 2) Les algorithmes à critères métriques qui, contrairement aux algorithmes précédents, s'intéressent aux points essentiels (c'est à dire aux points du squelette). Ces points sont situés normalement au centre de la forme (à égale distance des frontières). C'est la notion de

distance qui est à la base de ces algorithmes. Nous citerons dans ce cas l'algorithme de MARTON.

La figure II.7 illustre une opération de squelettisation.

0000011000000000000000	0000001000000000000000
0000011000000000000000	0000001000000000000000
0000011000000000000000	0000001000000000000000
0000011000000000000000	0000001000000000000000
0000011000000000001100	0000001000000000000000
000001100001100011110	000000100001000001100
000001100010110010000	000000100010100010000
000001111111111111111	000000100010110010000
100001111101101111110	100000011101001101110
1100011000000000000000	1000001000000000000000
1100011000000000000000	1000001000000000000000
1111110000000000000000	1000010000000000000000
0111100000000000000000	0111100000000000000000

Figure II.7 : Exemple du mot "عمل" squelettisé.

VIII.4 Segmentation

Les objets qui composent un document composite varient des plus simples (signes de ponctuation) aux plus compliqués (textes, tableaux, formules, graphiques). Ces objets sont appelés entités physiques et correspondent à des groupements de pixels délimités par un séparateur physique. L'entité de base est la composante connexe; le séparateur est alors la transition noir/blanc ou blanc/noir.

Les entités logiques correspondent à des groupements de pixels logiques délimités par un séparateur logique. Dans le cas de l'écriture, l'entité de base est le caractère qui ne correspond pas directement à l'entité physique pour les lettres attachées.

La segmentation est une opération qui consiste à séparer un document en ses entités élémentaires, lesquelles sont destinées au module de reconnaissance. Deux approches de segmentation différentes existent; celle qui tient compte du contour des objets : on parle alors de segmentation par extraction de contours. Et celle qui privilégie le contenu de la région à segmenter, donc les pixels qui la composent.

On ne peut pas dire qu'une approche est meilleure que l'autre, leur différence réside dans l'information qu'elles mettent en valeur et qui est le contour pour l'une et le contenu pour l'autre.

Comme nous le verrons plus loin, la segmentation d'un paragraphe de texte en caractères s'effectue en trois étapes : La segmentation du texte en lignes puis en parties connexes (qui peuvent être des mots ou des sous-mots) et enfin en caractères.

VIII.5 Reconnaissance

Elle constitue l'étape d'interprétation et de décision. Ce module de reconnaissance des caractères peut se décomposer en deux sous-modules [94]

- Extraction des caractéristiques.
- Classification et décision.

VIII.5.1 Extraction des caractéristiques

L'extraction des caractéristiques prend des formes différentes en fonction de la méthode utilisée. L'ensemble des caractéristiques le plus classique est obtenu en procédant à certaines mesures dont nous citons, pour le cas des méthodes structurelles[94] :

- Recherche de la distance entre chaque point du caractère et le bord du rectangle qui l'encadre, le nombre de trous fermés, le type de concavité,...
- Détection des angles.
- Calcul des centres de gravité.
- Recherche des points d'intersection, des points de branchement etc.

VIII.5.2 Classification et décision

La finalité de la reconnaissance est l'identification d'un caractère après extraction de ses caractéristiques. Un système ne peut reconnaître un caractère que s'il le connaît déjà. L'apprentissage est donc une phase importante de la chaîne de reconnaissance.

L'apprentissage consiste à construire un dictionnaire qui contiendra un certain nombre de prototypes obtenus en consultant toutes les formes possibles des caractères pouvant être soumis à ce système [95]. Plus le dictionnaire est riche, meilleure sera la reconnaissance.

Pour l'écriture arabe imprimée, on peut compter une vingtaine de fontes comportant en moyenne 300 formes différentes chacune [96]. Le caractère à reconnaître sera donc comparé à tous les prototypes, et sera identifié à celui dont les caractéristiques sont les plus proches. Le taux de reconnaissance pourrait être élevé, cependant, cette identification coûterait cher en temps de calcul et en espace mémoire, du fait de la grande taille du dictionnaire.

Un moyen pour pallier à cet inconvénient est de procéder d'abord à une classification des prototypes. Ainsi, le caractère à reconnaître ne sera plus comparé à tous les prototypes contenus dans le dictionnaire mais à un nombre restreint contenu dans une classe.

Donc dans cette étape, tout caractère saisi en vue de sa reconnaissance est représenté sous une forme analogue à celle utilisée pour décrire ceux du dictionnaire; il s'agit donc de trouver le prototype qui lui ressemble le plus et de décider si le caractère est identifié ou non.

Le résultat de l'identification pourrait être :

- **Une reconnaissance** : le caractère est effectivement reconnu (succès).
- **Un rejet** : le caractère n'est pas reconnu. Si l'apprentissage est dynamique, il permettra de rajouter les caractères non reconnus si ceci s'avère nécessaire.
- **Une ambiguïté** : le caractère est confondu, par le système, avec un autre (erreur de reconnaissance). Dans ce cas, il faudra rajouter des caractéristiques supplémentaires pour lever cette ambiguïté.

IX. Conclusion

Dans ce chapitre, nous avons fait une brève synthèse de la reconnaissance de l'écriture largement entamée dans la littérature [97, 98, 99,100] en spécifiant la place qu'elle occupe dans l'ARD et en faisant le lien avec le domaine de la RF. Nous avons présenté aussi une description générale d'un système de reconnaissance de l'écriture. Les différents modules d'un système OCR y sont décrits brièvement, à savoir les modules d'acquisition, de prétraitement, de segmentation et de reconnaissance. Cette dernière étape représente la partie la plus importante du processus. Ainsi, le premier obstacle que doit surmonter le concepteur est le choix de l'approche à utiliser pour son application.

Nous implémenterons dans le chapitre suivant plusieurs méthodes et mentionnerons les faiblesses et les apports de chacune d'elles par rapport aux autres..

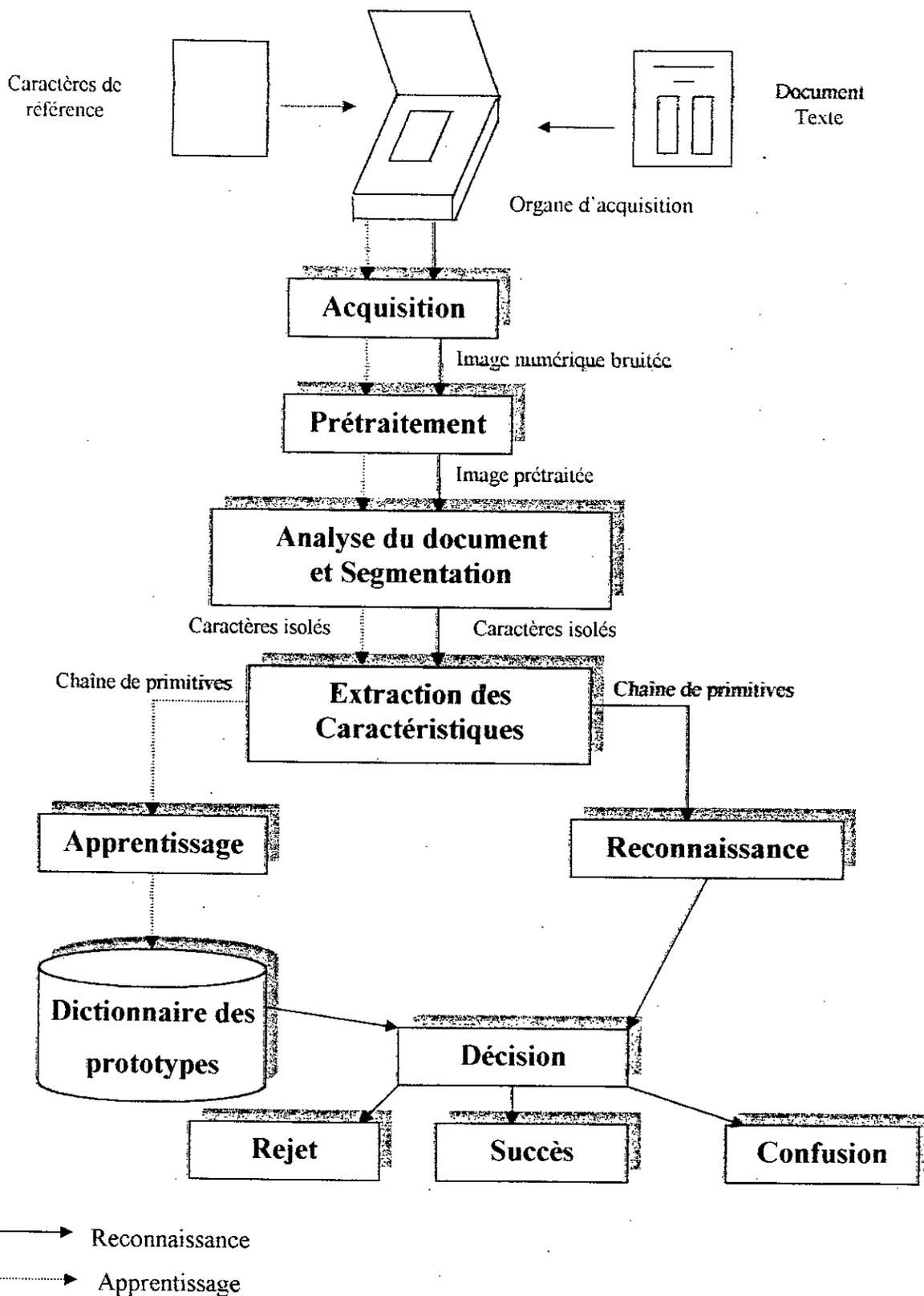


Figure II.6 : Système de Reconnaissance des caractères

CHAPITRE III

Implémentation des modules de Prétraitement

- I. Introduction**
- II. Acquisition d'un texte**
- III. Module de Filtrage**
- IV. Segmentation d'un texte**
- V. Conclusion**

I. INTROUCTION

Le travail réalisé consiste en la mise au point d'un système OCR destiné à reconnaître un texte arabe imprimé. Ce système, comme il a été décrit précédemment, est constitué des modules d'Acquisition, de Filtrage, de segmentation et de Reconnaissance. L'Apprentissage ou la constitution de la base de données est aussi une étape prédominante qui dépend très étroitement de la méthode de reconnaissance retenue.

Si la qualité du document texte n'est pas très bonne, alors l'opération de reconnaissance devient difficile à réaliser. A cet effet un module de filtrage basé sur des techniques d'amélioration d'images a été réalisé.

Le module de Reconnaissance repose quant à lui sur la qualité des résultats obtenus dans l'étape de segmentation, cette dernière est basée sur des constatations faites sur les caractères. La reconnaissance englobe deux phases qui sont : la classification, réalisée à partir des caractéristiques propres à l'écriture arabe et la décision.

Nous présenterons dans ce chapitre les modules d'acquisition, de filtrage et de segmentation réalisés.

Avant d'aborder ces modules, nous avons pensé utile de présenter une brève étude typographique de l'écriture arabe ainsi qu'un petit historique de la langue arabe.

II. PRESENTATION DE L'ECRITURE ARABE

II.1 Historique

La langue arabe sous l'une ou l'autre de ses formes parlées est la langue quotidienne de plus de 100 millions d'hommes. Sous sa forme écrite, elle est en outre la langue *liturgique* de plus de 600 millions de musulmans, arabophones ou non, répartis de l'Atlantique aux îles de la sonde, et des confins de la Sibérie à l'Equateur... C'est en effet la langue liturgique de l'Islam, mais aussi la langue de la littérature classique sous toutes ses formes. C'est encore essentiellement la langue de la littérature moderne, de la presse, de la radio, et de l'université [60].

En bref cela signifie qu'aujourd'hui tous les arabophones possèdent, en dehors de leur dialecte régional, une langue écrite commune, dont l'origine précède les écritures coraniques. Cette langue écrite reste commune, même si les dialectes parlés présentent d'importantes différences les uns par rapport aux autres. Elle est connue sous le nom d'arabe littéraire et est enseignée actuellement dans toutes les écoles du monde arabe. C'est l'écriture indiscutable du Coran et, de cette façon, étroitement liée à des millions de musulmans dans le monde répartis en Amérique, en Europe, en

Asie, en Afrique et en Océanie. C'est la langue et l'écriture de plus de quinze pays, de l'Asie jusqu'à l'Afrique (figure III.1).

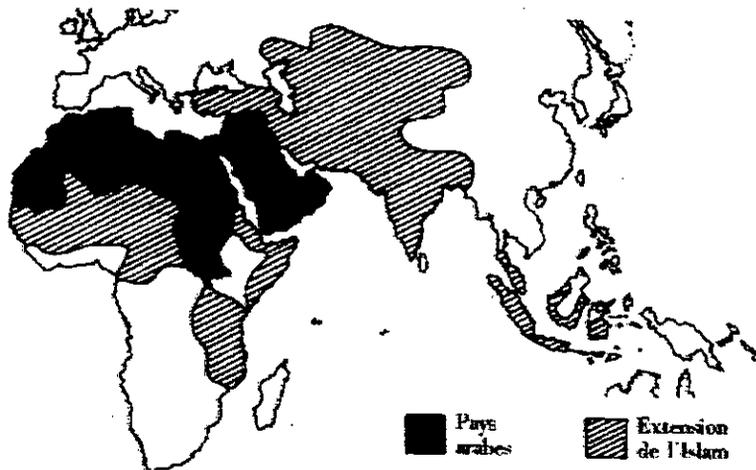


Figure III.1 : Carte du monde arabe et de ses zones d'influence linguistique due à la pénétration de l'Islam [60].

« L'écriture arabe est donc devenue et reste encore – en dépit de quelques reculs, et de la concurrence grandissante de l'alphabet latin- une grande écriture internationale [101].» Ceci nous indique également l'importance de l'écriture arabe, comparée à d'autres types d'écritures non latines comme l'hébreu, le japonais, le grec moderne, l'arménien, le coréen, le chinois etc.

Lecomte insiste sur l'importance qualitative de la structure de la langue : « ... Cette structure a l'avantage de présenter en principe une régularité morphologique et syntaxique étonnante, quasi géométrique, et de ce fait particulièrement séduisante aux esprits cartésiens. [60]»

L'écriture arabe a connu plusieurs styles qui se développèrent à travers les siècles. Parmi ces styles, une écriture angulaire de *style koufique* (de la ville de Koufa en Irak) a servi au début de l'islam à copier le Coran. Seulement à cause de sa structure complexe, elle est devenue une écriture décorative. Une autre structure plus souple appelée le *Naskhi* est venue ensuite pour remplacer rapidement la première. Cette écriture est la plus utilisée de nos jours par les imprimeries et les logiciels de traitement de texte.

Les caractéristiques essentielles de la calligraphie du Naskhi est la continuité et la cursivité (ronde). Les formes curvilignes révèlent un grand mouvement ondulatoire imprimé par la main qui progressent continuellement de droite à gauche et de haut en bas donnant ainsi un espace planaire plus important par rapport à l'écriture latine. De même la jonction entre les lettres constitue une caractéristique propre à l'écriture arabe imprimée. La nature de la liaison varie en fonction de deux formes voisines. Cette continuité engendre des altérations de formes curvilignes de base et certains

assemblages de lettres appelés *ligatures* qui sont fréquemment utilisées dans les textes arabes (voir figure III.2).

محمد الجزائر الحج ت محمد بحث أحمد

Figure III.2 : Exemples de ligatures

II.2 Styles d'écritures

L'écriture arabe varie, selon les milieux et les régions, d'une extrême simplicité formelle à la complexité exhaustive de l'arabesque. Des styles anciens, qui ne sont plus en usage aujourd'hui, ne restent compréhensibles que pour quelques spécialistes ; d'autres ne se retrouvent plus actuellement que sous la forme d'épigraphes et titres d'articles ou de livres, et naturellement sous la forme d'inscriptions religieuses. C'est pourquoi certaines écritures anciennes restent connues de tous. Certains dessins ont connu un développement récent, tels le *rekkah*, écriture manuelle rapide, inspirée de celle du Coran et le *naskhi*, utilisé dans l'imprimerie. Les autres formes actuelles sont généralement des mélanges d'écritures anciennes, essentiellement de style *koufique*. Les styles d'écriture les plus représentatifs et les plus utilisés sont par ordre d'importance : le Naskhi, le Koufi puis le Rakkah [60].

II.2.1 Style Naskhi

La majorité des écritures actuelles d'imprimerie sont une transcription directe de ce style. Il est caractérisé par une forte cursivité et une simplicité comparativement aux autres écritures. C'est ce principal style qui sera utilisé dans notre étude.

II.2.2 Style Koufi

C'est le style d'écriture le plus ancien. Il est formé de caractères de grande clarté ; bien qu'ils soient souvent surchargés d'ornements. Ce style se caractérise par des hauteurs centrales fortement prononcées et une épaisseur constante des traits.

II.2.3 Style Rekkah

D'origine turque, actuellement très répandu au Proche-Orient et particulièrement en égypte, il se présente comme une écriture très simplifiée caractérisée par des formes peu cursives. A titre d'exemple prenons la lettre *sin*, où les caractéristiques principales (trois dents) ont complètement disparues.

Les éléments qui composent une écriture d'imprimerie arabe peuvent être divisés en groupes :

- Caractères élémentaires
- Lettres combinées (ligatures),
- Chiffres,
- Autres signes (points diacritiques, signes orthographiques, etc.)

II.3 L'alphabet arabe

II.3.1 Les consonnes

L'alphabet arabe comporte 28 caractères de base. Le caractère « hamza » (ء) non considéré comme tel joue un rôle important dans la phonétique. Il peut être utilisé comme une lettre ordinaire indépendante, placé au dessus des lettres considérées comme des voyelles longues, au dessous de la lettre « Alif » ou au dessus de la forme finale et médiane de « Ya » (voir figure III.3).

أ ء ؤ لا إ ء

Figure III.3 : Différentes positions de la lettre hamza

Il existe un autre caractère de l'alphabet arabe, que l'on considère comme le 29^{ème}, et qui n'est autre que le graphisme contracté des deux lettres Lam et Alif : لا

Chaque caractère peut s'écrire selon deux à quatre formes différentes en fonction de sa position dans le mot : (voir figure III.5)

- ❖ Initiale : au début du mot.
- ❖ Médiane : au milieu du mot.
- ❖ Finale : à la fin du mot et lié à la lettre précédente.
- ❖ Isolée : non lié.

Six caractères parmi les 28 ont seulement deux formes distinctes initiale et finale :

alif, dal, dhal, zay, ra et waw (ا، د، ذ، ز، ر، و).

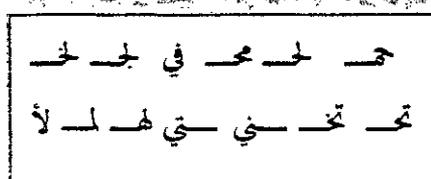
La présence de groupes de points et leurs positions jouent un rôle prépondérant pour la discrimination des caractères possédant le même tracé et appartenant à la même classe. Ces groupes de points dits diacritiques (1 à 3) peuvent avoir trois positions possibles : au dessus, au dessous ou au milieu du corps principal (voir figure III.6).

Traditional Arabic	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
Andalus	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
Deco Type Naskh	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
Deco Type Thuluth	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
Monotype Koufi	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
Mudir MT	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
Simplified Arabic	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
Traditional Arabic Backslanted	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
Simplified Arabic Fixed	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Figure III.4 : Quelques styles utilisés en langue arabe

Nom du Caractère	Position début	Position milieu	Position finale	Position isolée
Alif	ا	ا	ا	ا
Ba	ب	ب	ب	ب
Ta	ت	ت	تة	ة
Tha	ث	ث	ث	ث
Djim	ج	ج	ج	ج
Haa	ح	ح	ح	ح
Khaa	خ	خ	خ	خ
Dal	د	د	د	د
Dhal	ذ	ذ	ذ	ذ
Ra	ر	ر	ر	ر
Zay	ز	ز	ز	ز
Sin	س	س	س	س
Chin	ش	ش	ش	ش
Sad	ص	ص	ص	ص
Dhad	ض	ض	ض	ض
Tâ	ط	ط	ط	ط
Dhâ	ظ	ظ	ظ	ظ
Ain	ع	ع	ع	ع
Rain	غ	غ	غ	غ
Fa	ف	ف	ف	ف
Kâf	ق	ق	ق	ق
Kaf	ك	ك	ك	ك
Lam	ل	ل	ل	ل
Mim	م	م	م	م
Noon	ن	ن	ن	ن
Ha	هـ	هـ	هـ	هـ
Waw	و	و	و	و
Ya	ي	ي	ي	ي

Figure III.5 : Caractères arabes selon leurs quatre positions dans un mot



Exemple de quelques ligatures

<u>Tracé</u>	<u>Caractères</u>
ب	ب ت ث
ح	ح ج خ
د	د ذ
ر	ر ز
س	س ش
ص	ص ض
ع	ع غ

Figure III.6 : caractères ayant même tracé principal

II.3.2 Les voyelles

Il n'existe pas de voyelles brèves dans l'alphabet arabe mais celles-ci peuvent être représentées par certains signes qui sont au nombre de trois : fatha ou nasba (َ), kesra ou khafda (ِ) et damma ou rafaa (ُ). Ils sont parfois redoublés à la fin d'un mot et sont alors appelés « tenouin » et sont prononcés an(ّ), in (ّ), ouin (ّ). Ils servent à indiquer respectivement l'accusatif, le nominatif et le génitif dans l'arabe classique ou littéraire. La présentation habituelle des ouvrages imprimés omet ces signes voyelles, ce qui exige du lecteur une parfaite connaissance du vocabulaire de la langue. Quand les connaissances du lecteur sont insuffisantes, on recourt à la voyellisation ; on trouve généralement des textes voyellés dans les livres scolaires ou dans ceux où l'on prétend à une compréhension générale. Selon Qasim Amin, « dans les langues occidentales, on lit pour comprendre, alors qu'en arabe, il faut comprendre pour lire [102] ».

On notera aussi que trois caractères peuvent se comporter comme des voyelles dans certaines de leur position dans le mot (alif, waw et ya).

On rencontre aussi des signes orthographiques qui sont au nombre de quatre : chedda (), hamza (), soukoun () et madda (). Les signes de ponctuation sont similaires à ceux de la langue latine : point, virgule (), point-virgule (), deux points, point d'exclamation et point d'interrogation ().

III. ACQUISITION ET FILTRAGE

III.1 Acquisition :

Pour l'acquisition du document nous utilisons un scanner à plat (type AGFA SNAPSCAN 1212 utilisant le logiciel ScanWise) avec une résolution de 200 dpi jugée suffisante pour obtenir une bonne qualité de l'image acquise. L'image texte est obtenue sous forme de bitmap (format TIF). Le texte, que nous traitons, ne doit pas comporter de voyelles.

Au cours de cette étape, nous devons transformer l'image fichier en une matrice qui représentera l'image sur laquelle nous devons effectuer nos traitements. Pour cela, une étude détaillée du format TIF a été faite et un algorithme d'extraction et de visualisation de l'image a été établi (Voir Annexe A).

Comme l'image que nous traitons est à deux niveaux de gris (noir et blanc), nous lirons seulement les champs qui nous intéressent, à savoir l'adresse de l'IFD (entête du fichier), la longueur de l'image (champs 257), sa largeur (champs 256), l'adresse du premier pixel de cette image (champs 273) et enfin la couleur attribuée aux pixels (champs 262).

III.2 Filtrage :

Cette étape consiste à éliminer les bruits dans l'image binaire obtenue à la sortie du scanner. Plusieurs filtres ont été implémentés dont les filtres linéaires (filtre de Gauss, filtre moyen, filtre de Deriche, Filtre de Gabor) et le filtre médian non-linéaire. Le filtre médian est celui qui s'adapte le mieux à notre application (puisque le texte présente dans la majorité des cas un type de bruit impulsif) et c'est donc lui que nous avons implémenté dans notre système. Ce filtre considère chaque point de l'image et lui donne la valeur médiane de ses huit voisins.

L'implémentation de ce filtre est expliquée au chapitre I (Figure I.2).

La Figure III.7 donne un exemple d'utilisation de ce filtre.

Remarque :

Cette étape de filtrage peut être optionnelle et prend en considération l'état du document et du système d'acquisition.

ربنا إفتح بيننا و بين قومنا بالحق
و أنت خير الفاتحين

(a)

ربنا إفتح بيننا و بين قومنا بالحق
و أنت خير الفاتحين

(b)

Figure III.7: (a) image bruitée, (b) image filtrée.

IV. SEGMENTATION D'UN TEXTE :

IV.1 Introduction

Après l'étape d'acquisition, l'image obtenue est une matrice binaire contenant des zéros (0) et des uns (1).

L'objectif de ce travail est principalement l'étude de la partie médiane d'un système de reconnaissance automatique de l'écriture arabe imprimée. Il s'agit du module de segmentation qui reçoit en entrée un paragraphe de texte, et qui fournit en sortie des caractères séparés destinés au module de reconnaissance (voir Figure III.8). La méthode étudiée est basée sur l'observation des histogrammes des lignes et des colonnes et tient compte des caractéristiques propres à l'écriture arabe.

Les projections horizontale et verticale sont définies comme suit :

$$h(i) = \sum_j g(i, j) \text{ projection horizontale}$$

$$v(j) = \sum_i g(i, j) \text{ projection verticale}$$

Où $g(i, j)$ vaut 0 ou 1, i désignant les lignes et j les colonnes.

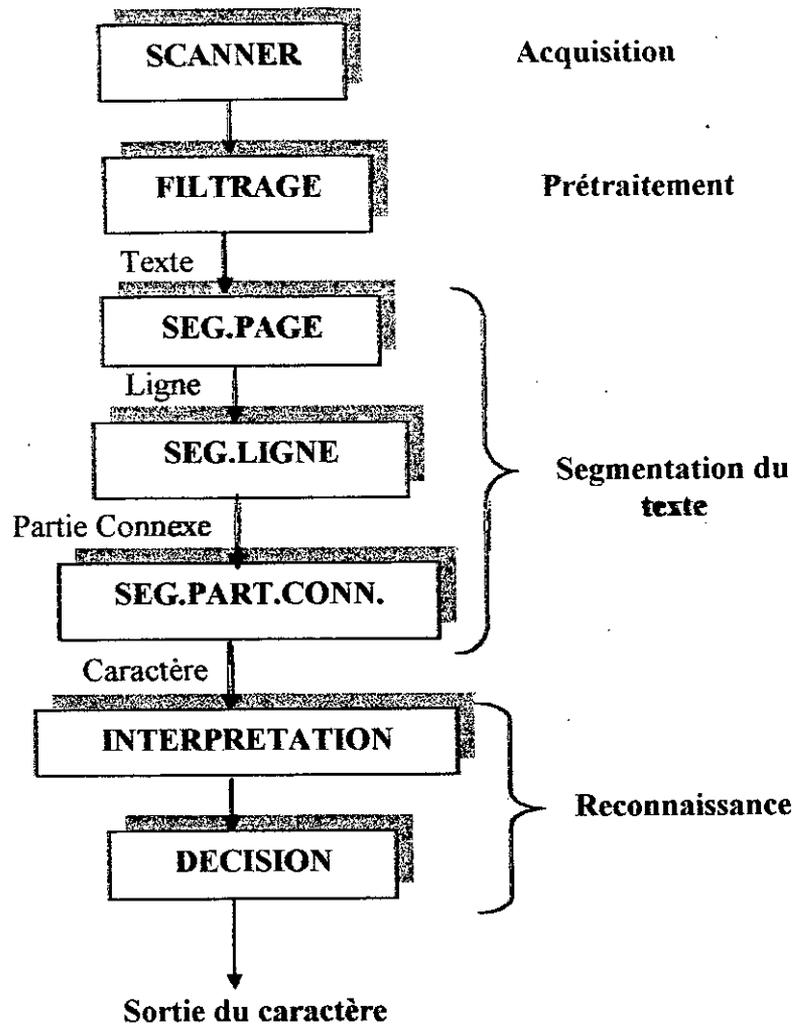


Figure III.8 : Block diagramme du système OCR

La segmentation se fait séquentiellement en trois étapes : [10,11]

- a) Localisation des lignes de texte.
- b) Séparation des lignes en parties connexes.
- c) Séparation des parties connexes en caractères.

La première étape constitue la segmentation horizontale et les deuxième et troisième étapes les segmentations verticales.

IV.2 Localisation des lignes de texte:

Cette étape constitue la segmentation horizontale (voir Figure III.9). Son principe est le suivant :

- Procédant à un balayage horizontal, on détermine le début de la ligne de texte qui correspond à la première ligne de la matrice binaire (ligne pixels) qui possède au minimum un pixel allumé.
- On détermine ensuite la fin de la ligne de texte qui correspond à ligne de la matrice binaire qui ne possède aucun pixel allumé. Le balayage horizontal s'effectue de droite à gauche car la langue arabe s'écrit dans ce sens.

Algorithme de la segmentation horizontale:

Début

Tant que non fin image (matrice binaire)

faire

Tant que la ligne de texte ne contient aucun point noir

faire

- Saut vers la ligne suivante

fait

Sauvegarder le début de la ligne de texte

Tant que la ligne contient au minimum un point noir

faire

- Saut vers la ligne suivante

fait

Sauvegarder la fin de la ligne du texte

Saut vers la ligne suivante

fait

Fin

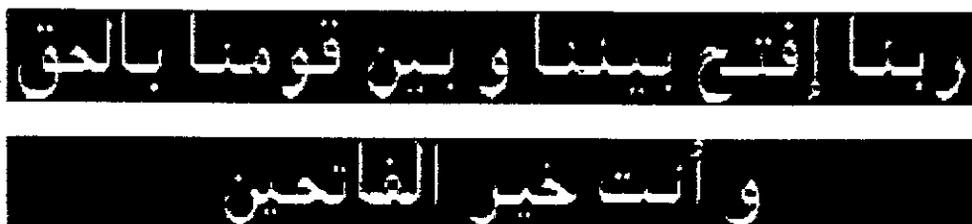


Figure III.9: Segmentation horizontale

On considère que deux lignes consécutives sont séparées par au moins une ligne pixels vide, pour laquelle la valeur de l'histogramme est nulle. On peut ainsi détecter les limites supérieure et inférieure de chaque ligne du texte. Si l'espace entre deux lignes de texte consécutives contient quelques points (chevauchement entre lignes), on ne peut plus détecter les limites d'une ligne.

A défaut de détecter les limites d'une ligne, on peut détecter la ligne médiane pour laquelle le nombre de pixels allumés est maximum : lignes à densité maximum (voir figure III.10).

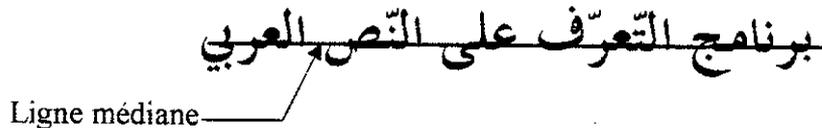


Figure III.10: Détection d'une ligne de texte

IV.3 Séparation des lignes en parties connexes

Cette étape est effectuée par une segmentation verticale. Elle consiste à déterminer le début et la fin de la partie connexe. Chaque ligne de texte obtenue dans la segmentation horizontale sera subdivisée en parties connexes qui ne sont pas forcément des mots (voir Figure III.11). Le balayage vertical se fait de haut en bas.

Son principe est le suivant :

- Procédant par un balayage vertical, on détermine le début de la partie connexe qui correspond à la première colonne de la matrice binaire qui contient au minimum un pixel allumé.
- On détermine ensuite la fin de la partie connexe qui correspond à la première colonne de la matrice binaire qui ne contient aucun pixel allumé.

Algorithme de la segmentation verticale :

Début

Pour chaque ligne de texte

faire

Tant que la colonne ne contient aucun point noir

faire

- Saut vers la colonne suivante

fait

Sauvegarder la colonne début de la partie connexe

Tant que la colonne contient un point noir au minimum

faire

- Saut vers la colonne suivante

Sauvegarder la colonne fin de la partie connexe

fait

Fin

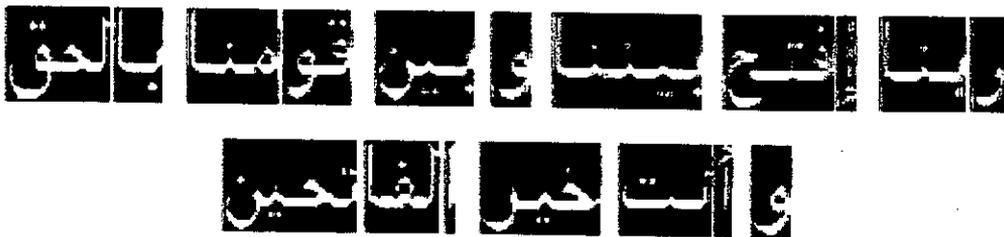


Figure III.11 : Segmentation verticale

IV.4 Segmentation des parties connexes en caractères

Son but est d'extraire les caractères de chaque partie connexe obtenue dans la segmentation verticale (voir Figure III.12).

Pour extraire un caractère de la partie connexe, il faut chercher :

- i- La ligne de jonction ou ligne médiane : celle ci constitue la région qui a la plus forte concentration de pixels en ligne.
- ii- La ligne de haut pour chaque colonne de la partie connexe : Par un balayage vertical commençant par la première ligne de haut, on détermine la ligne max. qui correspond au numéro de la ligne qui contient le premier point noir rencontré.
- iii- La ligne de bas pour chaque colonne de la partie connexe : Par un balayage vertical commençant par la dernière ligne, on détermine la ligne min. qui correspond au numéro de la ligne qui contient le premier point noir rencontré.
- iv- L'histogramme vertical pour chaque colonne de la partie connexe.
- v- Le seuil (ou zone de constance) qui correspond à la valeur la plus répétitive dans l'histogramme des colonnes.
- vi- Le nombre de transitions verticales (0 – 1) ou (1 – 0) pour chaque colonne de la partie connexe.

L'extraction d'un caractère se fait après avoir déterminé son début et sa fin.

Dans notre méthode nous avons introduit quelques conditions permettant la détection du début et de la fin d'un caractère.

- ❖ La colonne qui correspond au début d'un caractère doit vérifier la condition (1) suivante :
 - (1) L'histogramme vertical de cette colonne doit être supérieur au seuil.

- ❖ La colonne fin de caractère correspond à la dernière colonne qui précède le début du caractère suivant , de plus cette colonne fin doit vérifier les conditions (2) suivantes :
 - (2)
 - 1- La ligne haut de cette colonne doit être inférieure ou égale à la ligne médiane.
 - 2- La ligne bas de cette colonne doit être supérieure ou égale à la ligne médiane.
 - 3- La différence entre la ligne bas et la ligne haut doit être inférieure ou égale au seuil.
 - 4- L'histogramme vertical doit être inférieur ou égal au seuil.
 - 5- Le nombre de transitions verticales doit être égal à deux.
 - 6- La ligne haut de cette colonne doit être supérieure à la ligne haut de la colonne de début du caractère.

L'algorithme suivant permet l'extraction d'un caractère de la partie connexe après avoir déterminé son début et sa fin. Si le caractère est en position médiane , son début correspond à la colonne qui suit immédiatement la fin du caractère précédent.

Algorithme de la segmentation en caractères:

deb : variable booléenne (deb = vrai correspond au début du caractère).

Début

deb := vrai ;

Pour chaque partie connexe

faire

Tant que non fin de la partie connexe

faire

Si deb est vrai **alors**

- chercher la première colonne qui vérifie la condition (1);

- Sauvegarder la colonne début du caractère courant , soit j_1 ;
- $deb := faux$;

Si non

- $j_1 = j_2 + 1$;
- Saut vers la colonne suivante .
- Chercher la colonne de début du caractère suivant , soit j_1 ;

Si elle existe alors :

- Faire un balayage vertical de droite à gauche (de j_1 à j_1) ;
- Déterminer la première colonne qui vérifie les conditions (2) , soit j_2 ;

Si non dans ce cas on a atteint la fin de la partie connexe, soit j_2 ;

fait

Fin

- j_1 représente la colonne début de caractère.
- j_2 représente la colonne fin de caractère.

ربنا إفتح بيننا و بين قومنا بالحق
و أنت خير الفاتحين

Figure III.12 : Segmentation en caractères

IV.5 Problèmes rencontrés et solutions apportées à la segmentation

Nous avons testé cette méthode sur plusieurs textes arabes contenant 500 caractères . Les résultats comme attendus , n'étaient pas une segmentation à 100 % . En effet :

1- Le caractère " sin " demeure sur segmenté. Ce dernier est coupé en trois portions comme le montre l'image suivante :



Nous avons trouvé une solution à ce problème. A la base des constatations observées au niveau du caractère " sin ", nous avons pu déterminer quelques caractéristiques topologiques de ce caractère; ce caractère diffère des autres par :

- Il se compose de trois parties dont les débuts sont identiques.
- Le nombre de transitions verticales de chaque colonne du caractère " sin " est égal à deux.

Il faut donc modifier l'algorithme de segmentation en caractères comme suit :

Algorithme de segmentation modifié :

Début

- Déterminer le début du caractère courant , soit j_1 .
- Déterminer le début du caractère suivant , soit j'_1 .
- Déterminer le début du deuxième caractère suivant , soit j''_1 .

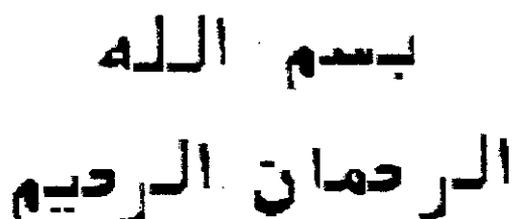
Si leurs histogrammes verticaux sont égaux et si les colonnes incluses entre ces trois colonnes (j_1 , j'_1 et j''_1) ont le nombre de transitions verticales égal à deux .

Alors Sauter vers colonne qui suit j''_1 .

Si non Sauter vers la colonne qui suit j_1 .

- Déterminer la fin du caractère courant.

Fin.



The image shows the Arabic calligraphic phrase "Bismillah" (Bismillah) written in a stylized font. The text is arranged in two lines: "بِسْمِ اللّٰهِ" on the top line and "الرّٰحْمٰنِ الرّٰحِیْمِ" on the bottom line. The character "sin" (س) is highlighted with a thick black border, illustrating the segmentation process described in the text.

Figure III.13: Segmentation du caractère " sin "

2- Il existe aussi certains caractères sous segmentés (cas des caractères qui se chevauchent et que l'on appelle ligatures). Ces problèmes ont été contournés en grande partie en considérant les caractères qui se chevauchent comme étant un seul.

Cette méthode a été testée sur les mêmes textes que précédemment, La majorité des caractères ont été correctement segmentés. Nous arrivons à un taux de segmentation très proche de 100 % (99,4 %), contrairement aux méthodes utilisées jusque là ([42, 103, 104]).

V. Conclusion

La reconnaissance dépend très étroitement de la segmentation. De ce fait, il faudra s'investir d'avantage dans cette partie de telle manière que la reconnaissance ne dépende plus que de la méthode de reconnaissance utilisée et du choix des paramètres.

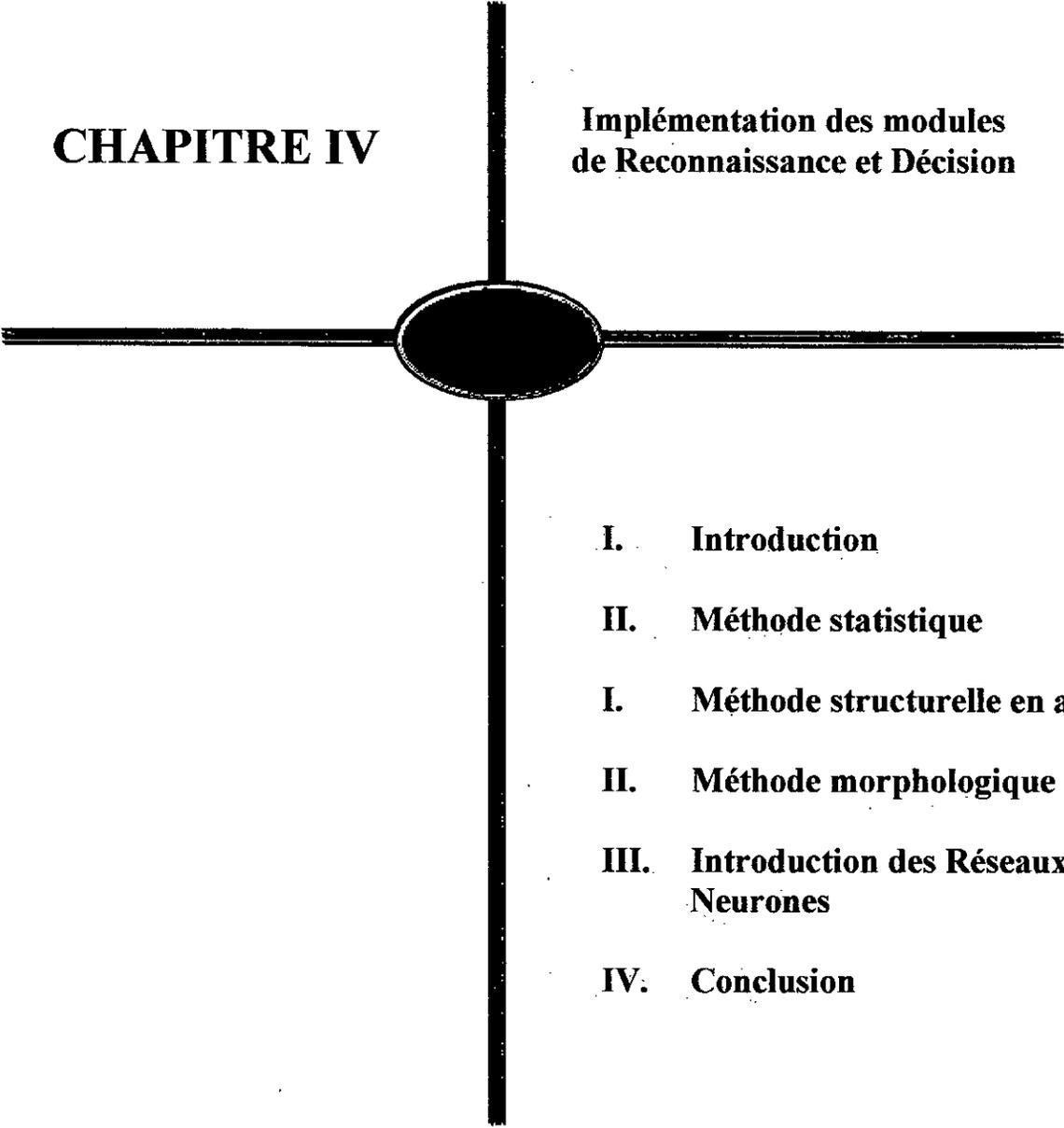
Dans la majeure partie des travaux de recherche, la segmentation d'un texte utilise une approche globale c'est à dire que la segmentation s'effectue sur les mots et non pas sur les caractères ; ceci facilite énormément l'étape de segmentation mais demande une base de données (un dictionnaire de mots) très volumineuse ainsi qu'une étude lexicographique et parfois même sémantique.

La segmentation en caractères est beaucoup plus complexe surtout quand il s'agit d'une écriture cursive (comme c'est le cas de l'arabe ou du latin manuscrit) ; mais dans ce cas, la base de données se limite au nombre de caractères existant dans l'alphabet et la recherche dans le dictionnaire prend un temps considérablement plus restreint.

Comme il a été montré dans ce chapitre, c'est ce deuxième type de segmentation que nous avons implémenté. Les résultats obtenus sont satisfaisants mais nous poursuivons nos travaux dans le but d'une amélioration. Nous avons essayé cette méthode sur l'écriture manuscrite mais les résultats sont nettement moins bons que pour l'imprimé ; nous suggérons, pour le manuscrit, l'utilisation de la segmentation par étiquetage qui résout une grande partie des problèmes de chevauchements très fréquents dans ce type d'écriture. Des travaux dans ce sens sont en cours au niveau de notre laboratoire.

CHAPITRE IV

Implémentation des modules de Reconnaissance et Décision

- 
- I. Introduction**
 - II. Méthode statistique**
 - I. Méthode structurelle en arbres**
 - II. Méthode morphologique**
 - III. Introduction des Réseaux de Neurones**
 - IV. Conclusion**

I. INTRODUCTION :

Comme nous l'avons vu dans les chapitres précédents, l'étape de reconnaissance et de décision constitue le dernier module d'un système de vision par ordinateur qui peut être réalisé par différentes méthodes.

Une fois la segmentation de l'image texte effectuée, chaque caractère sera modélisé dans le plan afin d'avoir une identité ; les caractères ne seront plus des formes géométriques mais deviendront *un vecteur d'attributs*.

Avant la reconnaissance proprement dite, il est nécessaire au préalable de constituer une base de prototypes: ceci constitue la phase d'apprentissage qui consiste à extraire les caractéristiques de tous les caractères prototypes. La dernière opération est la reconnaissance qui doit identifier le caractère à reconnaître à partir d'une règle de décision.

Nous avons implémenté plusieurs méthodes successivement dans le but constant d'une amélioration à apporter à la technique précédente. Comme nous le verrons dans ce qui suit, chaque méthode de reconnaissance implémentée améliore celle qui la précède.

Nous avons commencé par implémenter une méthode statistique qui a donné un bon taux de reconnaissance mais qui reste monofonte et monotaille (le problème de la taille a été résolu en ajoutant une routine de normalisation donnée en annexe B). Nous avons ensuite implémenté une méthode structurelle qui consiste à prendre comme descripteur l'arbre d'un caractère déterminé à partir de son squelette ; cette méthode a résolu le problème de la fonte mais non celui de la taille du caractère. Pour avoir un système multi fonte et multi taille sans avoir recours à la normalisation, nous avons pensé à utiliser une méthode qui prend en compte uniquement la morphologie du caractère ; pour cela nous avons étudié l'écriture arabe et défini des caractéristiques géométriques qui lui sont propres et qui vont nous permettre d'implémenter la dernière méthode.

Nous décrirons dans ce qui suit les différentes méthodes de reconnaissance implémentées, la constitution des bases de données (l'apprentissage) ainsi que les méthodes de décision qui leur sont associées.

II. METHODE STATISTIQUE

Dans ce cas précis, le vecteur d'attributs est constitué par des descripteurs de contour représentés par une série de sept moments. Le module d'interprétation et décision s'effectue en deux étapes ; la première consiste à extraire les caractéristiques et à réaliser la base d'apprentissage, la deuxième est celle qui aboutie à la reconnaissance en passant par une règle de décision appropriée.

II.1 Extraction des caractéristiques :

L'extraction des caractéristiques est le dernier processus de réduction de l'information, il est fortement lié à la structure de la bibliothèque des objets que l'on désire pouvoir reconnaître. En effet, ces caractéristiques doivent avoir la propriété d'effectuer une bonne discrimination d'un objet à l'autre tout en restant en nombre limité afin de réduire les temps de calcul tant au niveau de leur extraction qu'au niveau de leur analyse par le système de reconnaissance. Ceci explique pourquoi les systèmes actuels de vision sont capables de reconnaître un nombre limité d'objets pour une application donnée tout en gardant de bonnes performances.

Comme nous l'avons déjà vu précédemment, nous pouvons classer les caractéristiques suivant trois familles :

- Les caractéristiques topologiques.
- Les caractéristiques fonctionnelles.
- Les caractéristiques géométriques.

Dans notre cas, nous avons à faire à des caractères arabes isolés (segmentés). Ces derniers sont décrits par une série de moments, qui font partie des caractéristiques fonctionnelles. On obtient ainsi par décomposition de l'image dans une base de fonctions (moments) une représentation des caractères sous forme de série. La dimension de la base est sept du fait qu'on s'est limité aux sept premiers termes du développement.

Définition des Moments : [105]

Soit $I(i, j)$ l'image traitée que l'on veut reconnaître. Le moment généralisé d'ordre $(p+q)$ d'une telle image s'écrit : [105, 106]

$$m_{pq} = \sum_{i=0}^{\max \text{ lin}} \sum_{j=0}^{\max \text{ col}} i^p j^q I(i, j)$$

pour $p, q = 0, 1, 2, 3$.

Dans une image blanche sur fond noir, le moment d'ordre zéro représente **la surface du caractère** :

$$m_{00} = \sum_{i=0}^{\max \text{ lin}} \sum_{j=0}^{\max \text{ col}} I(i, j)$$

Les moments d'ordre 1 sont définis par :

$$m_{10} = \sum_{i=0}^{\max \text{ lin}} \sum_{j=0}^{\max \text{ col}} i I(i, j) \quad \text{et} \quad m_{01} = \sum_{i=0}^{\max \text{ lin}} \sum_{j=0}^{\max \text{ col}} j I(i, j)$$

Ils définissent le centre de gravité (\bar{X}, \bar{Y}) du caractère considéré :

$$\bar{X} = \frac{m_{10}}{m_{00}} \quad \bar{Y} = \frac{m_{01}}{m_{00}}$$

Les moments centrés d'ordre (p, q) sont donnés par :

$$\mu_{pq} = \sum_{i=0}^{\max lin} \sum_{j=0}^{\max col} (i - \bar{X})^p (j - \bar{Y})^q I(i, j)$$

Les moments centrés jusqu'à l'ordre 3 sont donnés par :

$$\mu_{00} = m_{00}$$

$$\mu_{10} = \mu_{01} = 0$$

$$\mu_{20} = m_{20} - \bar{X} m_{10}$$

$$\mu_{02} = m_{02} - \bar{Y} m_{01}$$

$$\mu_{11} = m_{11} - \bar{Y} m_{10}$$

$$\mu_{30} = m_{30} - 3\bar{X} m_{20} + 2m_{10} \bar{X}^2$$

$$\mu_{21} = m_{21} - 2\bar{X} m_{11} - \bar{Y} m_{20} + 2\bar{X}^2 m_{01}$$

$$\mu_{12} = m_{12} - 2\bar{Y} m_{11} - \bar{X} m_{02} + 2\bar{Y}^2 m_{10}$$

$$\mu_{03} = m_{03} - 3\bar{Y} m_{02} + 2m_{01} \bar{Y}^2$$

Ces moments sont invariants par translation de l'image.

Les moments centrés normés notés η_{pq} se définissent par l'expression :

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad \text{avec} \quad \begin{cases} p + q = 2, 3, \dots \\ \gamma = \frac{p + q}{2} + 1 \end{cases}$$

Ces moments engendrent des moments généralisés invariants par translation, rotation et homothétie (c'est à dire par le groupe des similitudes affines).

Nous en représentons ci-dessous un groupe de sept :

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right]$$

$$\phi_6 = (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right]$$

Ces moments ont la particularité d'être invariants par translation, rotation et changement d'échelle [105].

II.2 Apprentissage :

Cette étape consiste à établir une correspondance entre le caractère et son modèle. Notre système va extraire les moments de chaque caractère de l'alphabet de manière à constituer une bibliothèque de moments; cette notion de bibliothèque montre bien qu'on évolue dans un univers limité de caractères reconnaissables. La base de données que nous avons élaborée possède une structure matricielle de 61 lignes par 7 colonnes. Chaque ligne présente les caractéristiques (moments) d'un des caractères constituant la base de données. Chaque ligne porte le nom et/ou le code ASCII du caractère associé à ses descripteurs (voir figure IV.1).

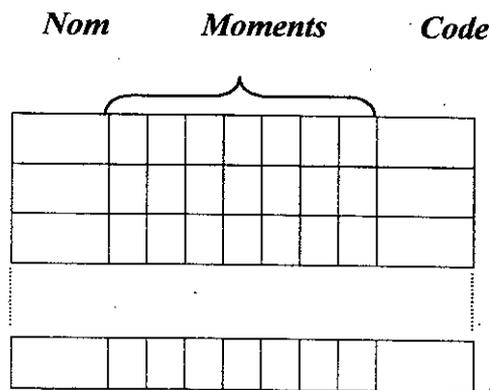


Figure IV.1 : Organisation du dictionnaire

II.3 Reconnaissance et Décision :

L'étape finale de la vision constitue l'interprétation des caractéristiques extraites de l'image ; c'est à dire la reconnaissance des formes. Le choix d'un processus de reconnaissance de formes est lié, à la base, au type de description que l'on désire pour caractériser un objet. Il en existe grossièrement deux :

- Description paramétrique.
- Description structurelle.

Le modèle choisi dans notre cas est un modèle paramétrique qui consiste à caractériser un caractère par un vecteur dont les éléments sont les sept moments invariants décrits précédemment. Le choix d'un tel modèle est dû à sa simplicité de construction. La seule difficulté consiste à choisir une liste de caractéristiques qui permettent une bonne discrimination entre les différents objets de la bibliothèque.

Règle de Décision :

Nous avons vu que chaque caractère de la base de données est représenté par un vecteur $S_{i,j}$ où $i=1..61$ et $j=1..7$, i désignant l'indice du caractère et j celui relatif aux caractéristiques (ordre du moment).

La règle de décision utilisée s'appuie sur la notion de distance euclidienne. On définit cette distance entre deux points repérés par leurs vecteurs r_1 et r_2 de dimension N par :

$$D(r_1, r_2) = \|r_1 - r_2\| \\ = ((r_1 - r_2)'(r_1 - r_2))^{1/2}$$

Où le facteur (') indique le vecteur transposé qui est utilisé pour mesurer la ressemblance entre deux objets.

La reconnaissance d'un caractère représenté par un vecteur R consiste à trouver la distance minimale $D(S_i, R)$ définie comme suit :

$$D(S_i, R) = \text{Min} \{D(S_i, R)\}_{i=1..61}$$

Donc la $i^{\text{ème}}$ ligne de la base de données représente les moments de l'élément le plus voisin du caractère x que l'on veut reconnaître.

II.4 Conclusion

L'objectif visé est largement atteint : il s'agissait d'étudier les performances des moments sur les caractères arabes isolés (segmentés) dans leurs différentes positions (début, milieu, finale et isolée). Les sept moments invariants ont été suffisants pour modéliser les caractères avec une bonne discrimination. En effet, le taux de reconnaissance global qui est approximativement de 99% est acceptable avec un temps de calcul suffisamment bon. La méthode s'est avérée efficace pour les caractères ayant subi des translations, rotations et symétries, mais moins performante pour

l'homothétie à cause des erreurs d'arrondi qui interviennent dans le calcul de cette transformation. Ce qui n'est pas le cas pour une homothétie réelle (non simulée).

Dans le cas des méthodes statistiques, le problème de fonte et de taille des caractères reste non résolu (car les moments représentent, entre autres, la surface du caractère et son centre de gravité). Ceci implique que nous devons apprendre à notre système de reconnaissance toutes les fontes de caractères que nous voudrions reconnaître, ce qui exige une grande taille mémoire pour la réalisation du dictionnaire.

Nous avons pour cela, introduit une routine de normalisation de la taille du caractère (voir annexe B) juste après l'étape de segmentation ; ce qui nous permet d'avoir une bibliothèque réduite et aussi une taille uniforme du caractère à l'entrée du module de reconnaissance. Trois fontes ont été apprises à notre système.

III. METHODE STRUCTURELLE (Structure d'arbres)

L'utilisation des arbres comme descripteur d'image peut être très utile en reconnaissance des formes. En effet leurs propriétés mathématiques autorisent l'utilisation d'algorithmes efficaces pour leurs manipulations. Les arbres permettent aussi de représenter une information structurelle plus abstraite comme les relations hiérarchiques entre des éléments donnés. Les méthodes de calcul de distance entre arbres ont été introduites. Elles permettent de décider de l'appartenance ou non d'une forme à une famille d'arbres.

Dans la méthode statistique, la ponctuation (ou les diacritiques) était traitée en même temps que le caractère car cette approche tient compte de la distribution des pixels dans l'image caractère. Par contre dans une approche structurelle, les points diacritiques sont une information supplémentaire dans la description du caractère. Pour cela, nous devons introduire une caractéristique supplémentaire qui pourrait être la classe par exemple des caractères ponctués ou pas. Le descripteur de chaque caractère sera un arbre dont le parcours est postfixé.

III.1 Classification des Caractères

Elle se fait en deux étapes : la ponctuation et le codage en arbre.

III.1.1 La Ponctuation :

Elle consiste à déterminer l'existence de points, leur nombre ainsi que leur position dans le caractère (1, 2 ou 3 points en haut, 1 ou 2 points en bas ou la hamza que nous considérons dans notre système comme une ponctuation).

Les caractères arabes sont divisés en deux classes principales, l'une contenant les caractères ponctués et l'autre ceux qui ne possèdent pas de points. Les caractères appartenant à la première classe diffèrent eux-mêmes selon le nombre et la position dans l'image des points qu'ils possèdent. Une telle image contient deux ou plusieurs amas de pixels allumés selon que le caractère possède un ou plusieurs points. Le caractère étant parcouru à partir du premier pixel allumé, deux cas peuvent se présenter :

- Le pixel rencontré appartient au corps du caractère qu'on appellera *amas du premier type*.
- Le pixel rencontré appartient au groupe de points du caractère qu'on appellera *amas du second type*.

Pour faire la distinction entre ces deux cas, il suffit de compter le nombre de pixels de l'amas rencontré. En effet, la dimension d'un groupe de points est généralement petite par rapport à la dimension du corps principal du caractère. Le rapport de ces deux dimensions est au maximum égal à 0,5, ce qui revient à dire qu'un groupe de points occupe au maximum le tiers du caractère entier.

L'organigramme de la Figure IV.2 représente la classification des caractères où les caractères de la première classe représentent ceux possédant des points diacritiques et ceux de la deuxième classe les caractères qui ne possèdent pas de points. D'où les possibilités suivantes :

- Le premier amas de pixels rencontré est un amas du second type (ex : ن ش ث...).
- Le premier amas de pixels rencontré est un amas du premier type, mais l'image contient encore un nombre non négligeable de pixels allumés n'appartenant pas à cet amas (ex : ي ب ج).

Si un caractère répond à l'une de ces conditions, il sera alors identifié comme appartenant à la première classe ; sinon il sera affecté à la deuxième classe.

III.1.2 Détermination de l'arbre du caractère :

L'arbre constitue le descripteur du caractère. Il sera représenté par une phrase cursive qui est le résultat d'une concaténation de sous-arbres. Pour la détermination de l'arbre on procède comme suit :

- On détermine le squelette du caractère (Par la méthode de Zang et Suen [105]).
- On réalise un échantillonnage du squelette qui permet de préserver sa connexité et pallier ainsi aux discontinuités survenues dans le squelette du caractère. Ces discontinuités, quoique ne dépassant pas 2 ou 3 pixels, sont très néfastes pour notre algorithme de parcours du squelette.

- On fait un suivi du squelette qui permet le codage du caractère selon les huit directions de Freeman. Nous effectuons un balayage du squelette de haut en bas et de droite à gauche ; le premier point rencontré ou *le sommet* est le plus haut à droite.

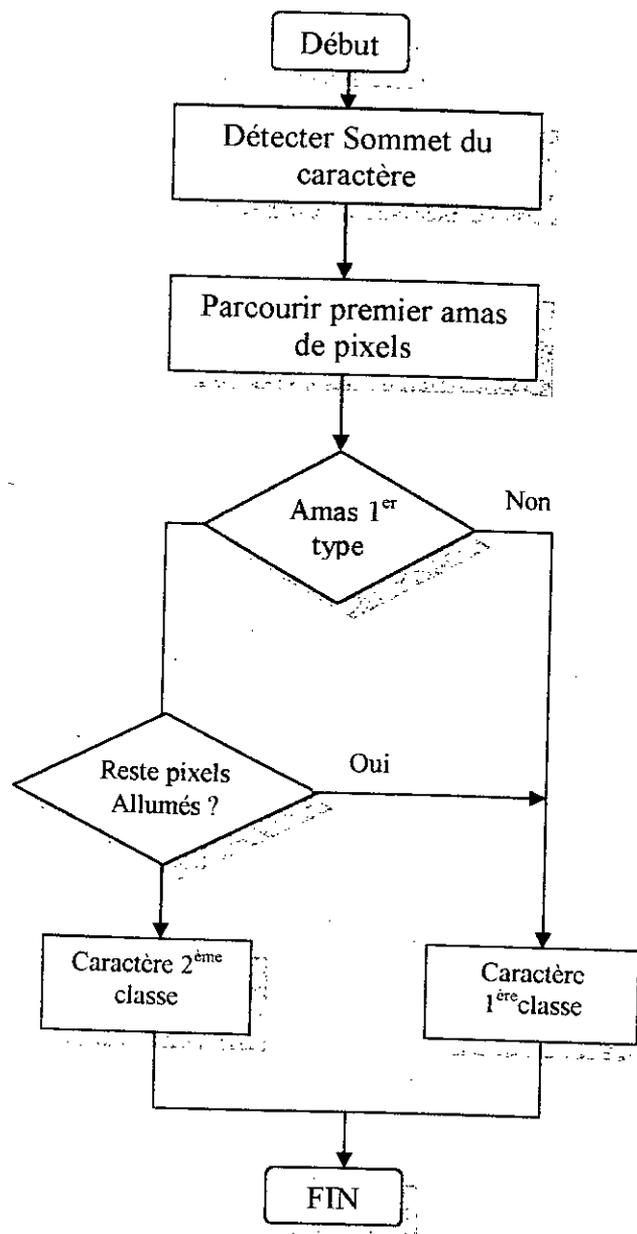


Figure IV.2 : Classification des caractères.

1. La Squelettisation :

C'est une technique de traitement d'images qui a pour but de simplifier l'image afin de la rendre plus facile à traiter. Cette opération s'accompagne toujours d'une perte d'informations ; pour remédier à ce problème, les différents algorithmes de squelettisation tentent de conserver les propriétés topologiques et métriques de la forme liées à la courbure, à l'orientation et à l'échelle.

Principes de la squelettisation :

Les critères retenus pour les méthodes de squelettisation sont :

- L'épaisseur du squelette doit être de 1 pixel ;
- Le squelette doit conserver les propriétés topologiques de la forme comme le nombre de parties, le nombre de trous et la connexité ;
- Le squelette doit respecter les propriétés métriques de la forme comme la longueur totale et la distance entre les parties de la forme.

Il existe deux grandes classes d'algorithmes de squelettisation :

❖ **Les algorithmes à critères topologiques** qui sont itératifs et qui suppriment à chaque étape le long de la frontière les points appelés inessentiels (c'est à dire qui n'appartiennent pas au squelette) ; ils opèrent sur des voisinages de points, déterminant à chaque passage la validité des points frontières. Parmi ces algorithmes nous pouvons citer : l'algorithme de SERRA, celui de THOME [107] et celui de ZHANG et SUEN que nous avons utilisé dans notre système et que nous détaillerons plus loin.

❖ **Les algorithmes à critères métriques** qui, contrairement aux algorithmes précédents, s'intéressent aux points essentiels (c'est à dire aux points du squelette). Ces points sont situés normalement au centre de la forme (à égale distance des frontières). C'est la notion de distance qui est à la base de ces algorithmes. Nous citerons dans ce cas l'algorithme de MARTON [108].

Algorithme de ZHANG et SUEN :

Les points de l'image sont traités par cet algorithme en deux étapes de base. Un pixel du contour étant tout pixel allumé ayant au moins un 8-voisin éteint. Soit p_2, \dots, p_9 les 8-voisins du pixel p_1 (voir figure IV.3).

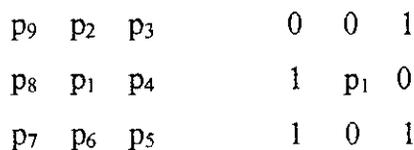


Figure IV.3 : disposition des 8-voisins de p_1

La première étape de cet algorithme force à zéro un point du contour, p_1 , qui satisfait les conditions suivantes :

1) $2 \leq N(p_1) \leq 6$

2) $S(p_1) = 1$

3) $p_2 \times p_4 \times p_6 = 0$

4) $p_4 \times p_6 \times p_8 = 0$

où : $N(p_1) = p_2 + p_3 + p_4 + \dots + p_9$ est le nombre de pixels allumés (à 1) voisins de p_1 .

$S(p_1)$ est le nombre de transitions 0-1 dans l'ordre p_2, p_3, \dots, p_9 .

Exemple : $N(p_1) = 4$ et $S(p_1) = 3$ pour la figure IV.3.

La première étape est appliquée à chaque point de l'image binaire. Si l'une ou plusieurs des conditions 1) à 4) ne sont pas vérifiées, la valeur du point en question reste inchangée. Si toutes les conditions sont vérifiées, alors le point est supprimé. On note qu'un point n'est supprimé qu'après que tous les points inessentiels aient été détectés afin d'éviter de changer la structure des données durant l'exécution de l'algorithme.

Dans la deuxième étape, les conditions 1 et 2 restent inchangées, mais les conditions 3 et 4 deviennent :

3') $p_2 \times p_4 \times p_8 = 0$

4') $p_2 \times p_6 \times p_8 = 0$

Cette étape est appliquée de la même manière que la première, à tous les points du contour de l'image résultante. On revient par la suite à l'étape 1 pour un deuxième balayage de l'image. Cette procédure est appliquée itérativement à l'image jusqu'à ce qu'il ne reste aucun point à supprimer, ce qui met fin à l'algorithme donnant le squelette de l'image.

Performances et Défauts de l'algorithme :

Cet algorithme est efficace et conserve l'essentiel de l'information dans le squelette ; de plus c'est l'un des algorithmes de squelettisation les plus rapides.

Comme tous les algorithmes de squelettisation, celui de Zhang et Suen présente certains défauts qui se résument dans les points suivants :

- On perd des pixels qui doivent faire partie du squelette, en général les traits obliques.
- Certaines formes disparaissent totalement : c'est le cas des formes carrées.

- Présence de discontinuités

Pour remédier au premier défaut, on peut modifier la condition $2 \leq N(p_1) \leq 6$ par $3 \leq N(p_1) \leq 6$. Cela permet de préserver les extrémités des différents segments du squelette mais conduit parfois à un squelette dont l'épaisseur est supérieure à 1 pixel.

Pour remédier au troisième défaut qui est la discontinuité, nous procédons à un re-échantillonnage que nous décrivons ci-après.

2. Re-échantillonnage :

Pour éliminer les discontinuités du squelette et simplifier son codage, il est nécessaire d'effectuer une opération de re-échantillonnage du squelette. On adopte pour cela une grille d'espacement plus large entre les différents points de l'image, un point du nouveau squelette sera assigné à chaque nœud de la grille si le nœud est proche du squelette original (voir figure IV.4).

En adoptant un pas de re-échantillonnage de 3 pixels, ceci nous a permis d'éliminer les discontinuités sans pour autant altérer la forme du squelette original et donc celle du caractère.

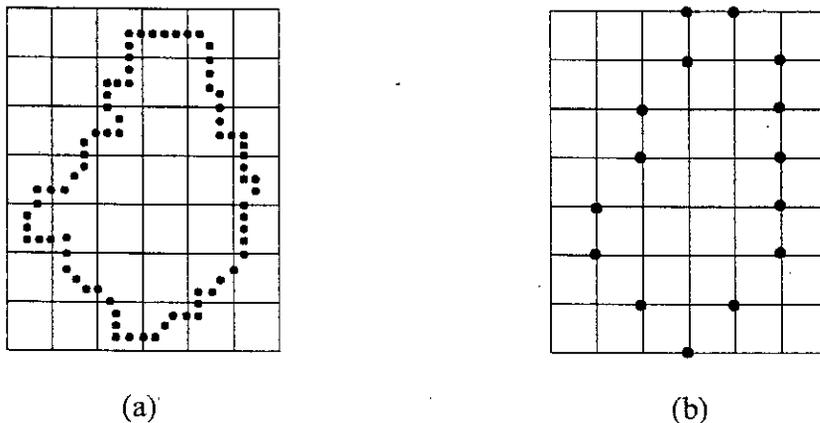


Figure IV.4 : Re-échantillonnage a) Nouvelle grille d'espacement
b) Forme re-échantillonnée

Algorithme de re-échantillonnage :

D1 : Matrice du squelette d'origine D1(n,m)

D2 : matrice du squelette après re-échantillonnage

(D1 : Données, D2 : Résultat)

Début

Pour chaque élément de la matrice D1

Faire

```
Si  $D1(I,J) = 1$  alors
  Faire
     $i = I \bmod 3$ ;
     $j = J \bmod 3$ ;
    Si  $i=2$  alors  $i=I+1$ ;
      Sinon  $i=I-i$ ;
    Si  $j=2$  alors  $j=J+1$ ;
      Sinon  $j=J-j$ ;
    Si  $D2(i, j)=0$  alors  $D2(i, j)=1$ ;
  Fait
```

Fait

Fin

3. Parcours du squelette et son codage :

Après l'opération de re-échantillonnage, il faudra parcourir le nouveau squelette afin de le coder en arbre étiqueté binaire tel que chaque nœud de l'arbre porte pour étiquette une direction ; la racine portera une étiquette spéciale. On utilisera pour cela le code de Freeman dans lequel une forme quelconque sera approchée de façon discrète par une suite de vecteurs de tailles élémentaires et de directions choisies dans un ensemble fini (8) (voir figure IV.5).

La première opération de parcours du squelette consiste à détecter, par un balayage horizontal de l'image, le premier point du squelette qui est le point le plus haut à droite de l'image. Ce point sera le sommet du caractère ainsi que le sommet de l'arbre qui va coder le squelette.

On cherche ensuite si l'un des 8-voisins du premier point détecté est allumé ; ce qui revient à faire une sorte de *balayage circulaire* autour du point considéré. Le sommet du squelette est étiqueté comme *ancien point* et on se déplace vers le nouveau point détecté. On cherche dans son voisinage les points allumés en ignorant les anciens points. On se déplace ainsi de proche en proche, en parcourant entièrement l'amas de pixels du squelette. Si au voisinage d'un point aucun pixel n'est allumé, mis à part les anciens points, cela voudra dire qu'on a atteint l'extrémité du squelette.

Si au voisinage d'un point, il y a deux pixels allumés, il s'agit d'un coin.

Si au voisinage d'un point, il y a trois pixels allumés, il s'agit d'une intersection ; dans ce cas les coordonnées de l'intersection (*les coordonnées du pixel courant et le nœud courant de l'arbre*) sont

mémorisées dans une pile. On suit l'un des deux chemins jusqu'à son extrémité, ensuite on revient à l'intersection pour parcourir le deuxième chemin.

Ainsi :

- Le sommet du squelette va constituer le sommet de l'arbre.
- Les différents traits du squelette vont constituer le chemin de l'arbre.
- Les primitives du code de Freeman (direction prise lors du passage d'un point du caractère à un autre) seront les nœuds de l'arbre.

L'exemple de la Figure IV.6 donne le contour du caractère, son squelette ainsi que le suivi du squelette. La Figure IV.7 quant à elle donne l'arbre du caractère considéré (ce qui constitue son code).

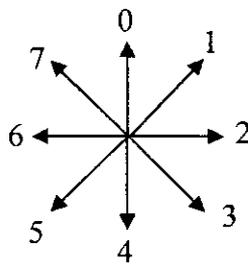


Figure IV.5 : Les huit directions de Freeman

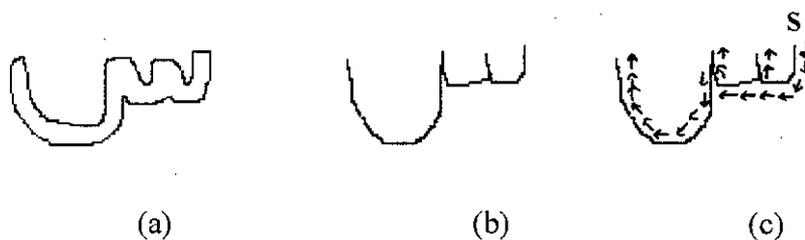


Figure IV.6 : Contour (a), Squelette (b) et Suivi du squelette (c) du caractère « sin »

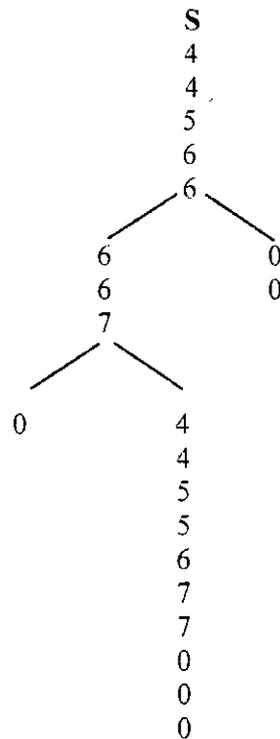


Figure IV.7 : Arbre du caractère « Sin »

Algorithme de parcours du caractère et son codage ;

P_c : point courant

P_a : ancien point

S : sommet de l'arbre

B, intersection : Booléen

N_c : Nœud courant de l'arbre

Début

- Détecter le sommet ;
- *P_c* := sommet
- *B* := vrai ;
- intersection := faux ;

Tant que B est vrai

Faire

-Tester les 8-voisins de *P_c*

Si nombre de pixels allumés égal 2

Alors

Faire

- Empiler les coordonnées d'intersection de P_c et de N_c
- $P_a := P_c$
- $P_c :=$ Premier pixel allumé détecté

Fait

Sinon

Faire

Si aucun pixel allumé

Alors

Faire

Si la pile est vide

Alors $B :=$ faux

Sinon

Faire

- Dépiler les coordonnées de l'intersection (P_c, N_c)
- Mise à jour du nœud courant de l'arbre et du pixel courant
- Intersection := vrai

Fait

Fait

Sinon

Faire

$P_c :=$ pixel détecté ; $P_a := P_c$;

Fait

Fait

Si intersection = vrai **Alors**

Si $P_a =$ Sommet **Alors** coder (P_a, P_c) Par S

Sinon

- Coder (P_a, P_c) suivant le code de Freeman
- Créer un nœud étiqueté par le code (P_a, P_c)
- Mise à jour de l'arbre.
- Intersection := vrai ;

Fait

Fin

III.2 Apprentissage

Dans cette méthode, chaque caractère prototype sera représenté dans le dictionnaire par un vecteur de caractéristiques structurelles.

Comme la méthode de codage du caractère arabe adoptée (algorithme de parcours du caractère) ne code que le corps du caractère ; il s'est avéré nécessaire de construire pour chaque caractère inconnu à reconnaître un vecteur de caractéristiques structurelles qui indique :

- 1- Le nombre de points diacritiques ou la présence de la hamza.
- 2- La position des points ou de la hamza.
- 3- La position du caractère dans le mot (cette information est fournie dans la phase de segmentation).
- 4- Un pointeur vers l'arbre du caractère.

N_p représentera le nombre de points diacritiques :

$N_p = 1$ Si le nombre de points est égal à 1

$N_p = 2$ Si le nombre de points est égal à 2.

$N_p = 3$ Si le nombre de points est égal à 3.

$N_p = 4$ indique la présence d'une hamza.

P_p déterminera la position des points diacritiques ou de la hamza :

$P_p = 0$ les points ou la hamza se trouvent au dessus du caractère.

$P_p = 1$ les points ou la hamza se trouvent en dessous du caractère.

P_c représente la position du caractère dans le mot :

$P_c = 0$ Le caractère est en position *début*.

$P_c = 1$ Le caractère est en position *milieu*.

$P_c = 2$ Le caractère est en position *finale*.

$P_c = 3$ Le caractère est en position *isolée*.

Deux critères ont été établis pour différencier entre la hamza et un point diacritique :

- un point isolé ou la hamza ont une forme presque carrée ; nous avons considéré le rapport de la largeur sur la hauteur, du cadre de cet amas, compris entre 0.8 et 1.2
- Le nombre maximal de transitions verticales (0-1 et 1-0) dans un point isolé est égal à deux et pour la hamza il est supérieur à deux.

Algorithme de construction du vecteur de caractéristiques :

Début

- Détecter le sommet du caractère.
- Parcourir tout le caractère.
- Détecter le nombre d'amas existants.

Si le nombre > 1 alors

- Classer les amas suivant leur type
- Déterminer le nombre de points ou l'existence de la hamza.
- Déterminer la position des points ou celle de la hamza.

Sinon

- Nombre de points = 0
- Position des points = 0
- Appliquer l'algorithme de squelettisation
- Appliquer l'algorithme de re-échantillonnage
- Coder le squelette du caractère en arbre
- Créer un pointeur vers l'arbre
- Construire le vecteur des caractéristiques du caractère à identifier.

Fin

L'apprentissage consiste, en fait, à répartir en classes un grand nombre de caractères prototypes construisant ainsi le dictionnaire. Les caractères arabes sous leurs différentes positions sont au nombre de 116. Parmi eux, il existe certains caractères qui possèdent le même code (arbre) mais des caractéristiques structurelles différentes; pour cela, une phase de classification des prototypes est indispensable. Cette classification permettra de comparer le caractère inconnu avec un nombre réduit de caractères prototypes d'une classe donnée.

La classification des prototypes se fait suivant les caractéristiques structurelles suivantes :

- 1- existence de points diacritiques
- 2- existence d'une hamza
- 3- nombre de points diacritiques
- 4- position des points diacritiques ou de la hamza.

Cette classification conduit à 8 classes, chacune d'elles étant subdivisée en 4 sous-classes suivant la position du caractère dans le mot. Ces classes sont réparties de la sorte :

- 1^{ère} classe* : Celle des caractères ne possédant pas de points.
- 2^{ème} classe* : Celle des caractères possédant un point en dessous.
- 3^{ème} classe* : Celle des caractères possédant deux points en dessous.
- 4^{ème} classe* : Celle des caractères possédant une hamza en dessous.
- 5^{ème} classe* : Celle des caractères possédant un point au dessus.
- 6^{ème} classe* : Celle des caractères possédant deux points au dessus.
- 7^{ème} classe* : Celle des caractères possédant trois points au dessus.
- 8^{ème} classe* : Celle des caractères possédant une hamza au dessus.

Nous pouvons ainsi regrouper sous forme de tableau, les vecteurs de caractéristiques associés à chaque classe (voir Tableau IV.1)

N° de classe	Vecteur des caractéristiques structurelles			
	A	B	C	D
1	0	0	x	p
2	1	0	x	p
3	2	0	x	p
4	4	0	x	p
5	1	1	x	p
6	2	1	x	p
7	3	1	x	p
8	4	1	x	p

- A : existence de points ou de la hamza.
- B : position des points ou de la hamza.
- C : position du caractère prototype dans le mot.
- D : pointeur vers l'arbre du caractère.

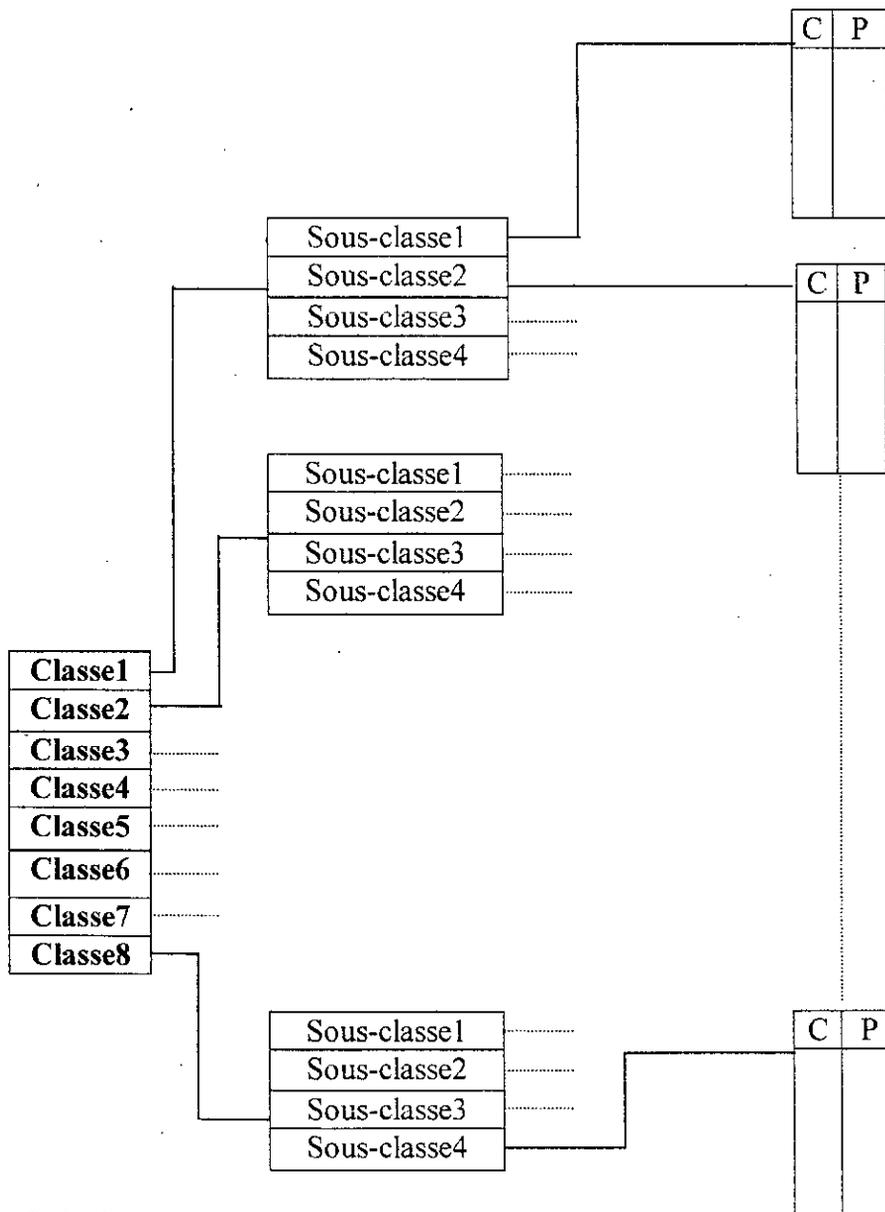
Tableau IV.1 : vecteurs de caractéristiques associés aux différentes classes

A l'issue de cette organisation, un caractère prototype sera représenté dans le dictionnaire par le caractère type (appelé descripteur prototype) mis en mémoire et auquel sera comparé l'arbre qui décrit le caractère inconnu .

Structure du dictionnaire :

La représentation d'un arbre en mémoire est une liste chaînée ; nous avons donc associé à chaque caractère une liste chaînée. La structure de données que nous avons adopté pour le dictionnaire est une structure de tableau. A chaque sous-classe du dictionnaire, nous avons associé un tableau

possédant deux champs tel que: Le premier champs représente le code du caractère et le deuxième champs représente un pointeur vers la liste chaînée du caractère (voir figure IV.8).



C : Code du caractère prototype
 P : Pointeur vers la liste chaînée du caractère prototype

Figure IV.8: Structure du dictionnaire

Mode d'accès au dictionnaire :

Pour améliorer le temps d'accès au dictionnaire, nous avons introduit une fonction de calcul d'adresse. Cette fonction détermine le numéro de la classe et celui de la sous-classe à laquelle

appartient le caractère à reconnaître. Ce numéro sera calculé en fonction des caractéristiques structurelles de ce caractère.

Algorithme de la fonction « calcul d'adresse »:

Donnée : Vecteur des caractéristiques structurelles.

Résultat : n_1 numéro de classe, n_2 numéro de sous-classe.

Nbr : nombre de points ou présence de la hamza.

Pos : position des points ou de la hamza.

Début

Si Nbr = 4 alors $n_1 = \text{Nbr} \times (\text{pos} + 1)$;

$n_2 =$ position du caractère dans le mot ;

 sinon $n_1 = \text{Nbr} + (\text{pos} \times 3) + 1$;

$n_2 =$ position du caractère dans le mot ;

Fin

A l'aide de cette fonction on obtient le numéro de classe n_1 et celui de la sous-classe n_2 .

L'identification se fait en comparant l'arbre du caractère inconnu avec les arbres des caractères de cette sous-classe.

III.3 Reconnaissance et Décision :

Après avoir déterminé le vecteur des caractéristiques du caractère à reconnaître dans la phase de paramétrisation, nous réalisons l'étape de reconnaissance et de décision qui se fait par consultation d'une sous-classe de prototypes. Nous utilisons pour cela la distance entre arbres.

L'intérêt de définir une distance entre arbres est de pouvoir se munir d'une métrique qui servira à comparer deux formes codées par un arbre et à décider si une forme appartient à une famille de formes.

Dans le cas qui nous intéresse, tous les caractères de l'alphabet arabe ont été codés sous forme d'arbres et stockés en mémoire dans un dictionnaire. Comme nous l'avons vu précédemment, après la première étape qui consiste à affecter le caractère à reconnaître à l'une des sous-classes du dictionnaire, celui-ci va être ensuite comparé à chacun des caractères de cette sous-classe. Nous avons utilisé pour cela la distance de LU [109]. Le caractère à reconnaître sera associé au caractère du dictionnaire le plus ressemblant, c'est à dire celui dont la distance au caractère considéré est la plus petite.

III.3.1 Règle de Décision :

Nous devons en premier lieu faire un parcours d'arbre. procédure qui fournit en résultat la liste des nœuds. La règle de décision est basée sur la distance minimale entre arbres déterminée par le coût minimal nécessaire pour transformer l'arbre «a» en l'arbre «b» par utilisation d'une série de transformations comme : l'insertion ou la suppression d'un nœud ou encore la substitution. A chacune de ces opérations est associé un coût qui ne dépend que de cette opération (par exemple, l'insertion n'importe où de n'importe quel nœud coûte toujours la même chose). L'ordre de parcours de l'arbre est l'ordre post-fixé [27].

1. Parcours d'arbre :

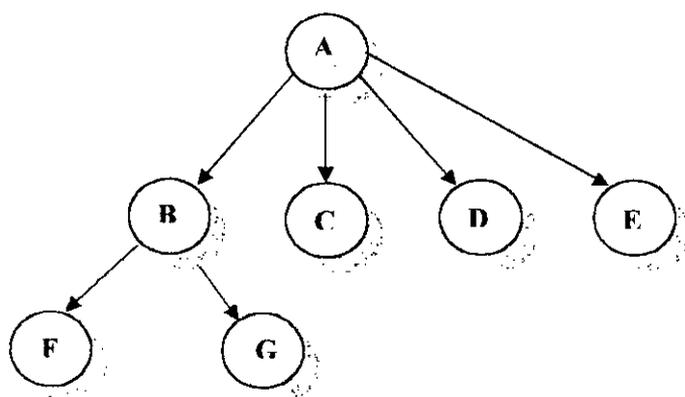
Le parcours d'un arbre s'effectue à l'aide d'un algorithme de parcours. C'est une procédure qui donne la liste des nœuds de l'arbre sans oubli ni répétition. Cette procédure ordonne totalement l'ensemble des nœuds de l'arbre.

On possède deux relations d'ordre partiel entre les nœuds d'un arbre : l'ordre de filiation et l'ordre d'aînesse entre les fils d'un même père. Il existe des relations d'ordre total, mais elles impliquent un certain arbitrage. Parmi ces relations d'ordre total, les plus habituelles sont l'ordre post-fixé (ou post-ordre) et l'ordre préfixé (ou pré-ordre). Dans notre travail, nous utilisons l'ordre de parcours post-fixé qui suit les étapes suivantes :

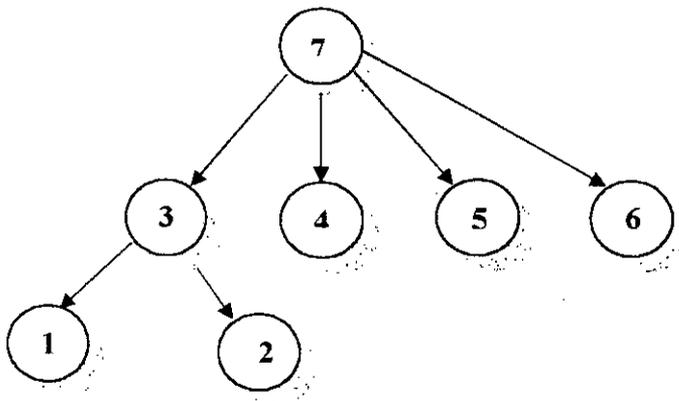
- Parcourir en ordre post-fixé le premier sous-arbre de la racine.
- Parcourir en ordre post-fixé le dernier sous-arbre de la racine.
- Examiner la racine.

Exemple

Soit l'arbre suivant :



Le parcours post-fixé de cet arbre est FGBCDEA, et l'arbre étiqueté suivant l'ordre de parcours est :



2. Distance entre arbres

On a vu que la technique la plus simple pour utiliser les structures d'arbres en reconnaissance des formes consiste à définir une métrique entre arbres. On pourra ainsi comparer deux arbres entre eux, et donc comparer les formes codées par ces arbres et décider si telle forme appartient à une famille de formes. Il existe plusieurs algorithmes de calcul de distances entre arbres. Nous expliquerons l'algorithme de Lu que nous avons utilisé.

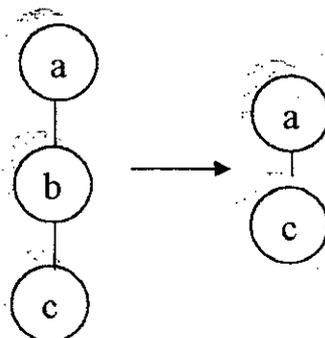
Algorithme de Lu :

Dans cet algorithme, le calcul de distances entre arbres est fondé sur des transformations élémentaires. A chacune de ces transformations est associé un coût, qui ne dépend que de l'opération. Par exemple l'insertion ou la suppression n'importe où de n'importe quel nœud coûte toujours la même chose.

Soient α et β deux arbres. Une transformation $T : D\beta \rightarrow D\alpha$ consiste en l'un des trois types de transformations suivantes :

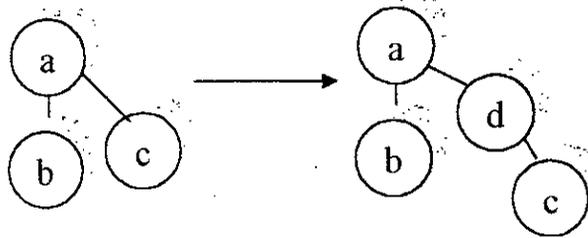
- Suppression d'un nœud n'importe où dans l'arbre.

$$\varepsilon(b) \rightarrow \lambda, p$$



➤ Insertion d'un nœud.

$$\Phi(\lambda) \rightarrow a, q$$



➤ Changement de l'étiquette d'un nœud.

$$\sigma(b) \rightarrow a, r$$

où $b \in D \beta$, λ est l'arbre vide (symbole nul), et p , q et r les coûts associés respectivement à ε , Φ et σ .

$\sigma(b) \rightarrow a, r$ provoque une substitution si les étiquettes des nœuds a et b ne sont pas identiques. Dans ce cas r est non nul, sinon $r = 0$.

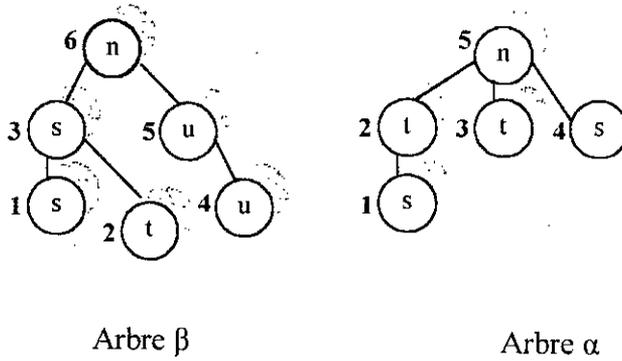
On définit alors la distance entre deux arbres α et β , notée $d(\alpha, \beta)$, comme la suite d'opérations la moins coûteuse transformant l'arbre β en l'arbre α en utilisant des séries de transformations T et vérifiant les conditions suivantes :

Soient deux nœuds A et B d'étiquettes respectives n et m . Ils sont transformés, par une suite d'opérations, en nœuds A' et B' d'étiquettes n' et m' :

- 1) $A=B \iff T(A) = T(B)$
- 2) La transformation T doit préserver l'ordre filial.
- 3) La transformation T doit préserver l'ordre post-fixé.

Si on affecte la valeur unité aux coûts associés aux transformations ε et Φ (suppression et insertion) et à la transformation σ (substitution), alors la distance devient le nombre minimal de suppressions, insertions et substitutions nécessaires pour obtenir un arbre d'un autre.

Exemple :



Transformation valide entre α et β :

Arbre β	{	arbre post-fixé	1	2	3	4	5	6	
		étiquette	s	λ	t	s	u	u	n
			↓	↓	↓	↓	↓	↓	
Arbre α	{	étiquette	s	t	t	λ	λ	s	n
		arbre post-fixé	1	2	3		4	5	

Pour cet exemple le coût est égal à quatre (une insertion, deux suppressions et une substitution)
 λ est l'étiquette d'un nœud ayant subi une opération de suppression ou d'insertion.

En d'autres termes, nous pouvons représenter les différentes transformations T_1 comme suit :

- $T_1 : D\beta \longrightarrow D\alpha$
- $\sigma(s) \longrightarrow s, 0$
- $\Phi(\lambda) \longrightarrow t, 1$ (insertion)
- $\sigma(t) \longrightarrow t, 0$
- $\varepsilon(s) \longrightarrow \lambda, 1$ (suppression)
- $\varepsilon(u) \longrightarrow \lambda, 1$ (suppression)
- $\sigma(u) \longrightarrow s, 1$ (substitution)
- $\sigma(n) \longrightarrow n, 0$

T_1 est la séquence de coût minimal qui transforme l'arbre β en l'arbre α . Donc $d(\alpha, \beta) = 4$.

III.4 Conclusion

Nous avons, par ce travail essayé d'apporter une amélioration au système OCR utilisant les moments statistiques. La méthode des arbres qui est une méthode structurale, descriptive utilise des descripteurs d'image puissants et efficaces que sont les arbres. Ces descripteurs se prêtent d'autant plus aux caractères arabes qui sont caractérisés par une structure qui possède beaucoup de *croisements*, de *courbures* et de *changements de directions*. Cependant l'opération de squelettisation que nécessite cette méthode a causé quelques problèmes. En effet, cette opération ne conserve pas toujours la connexité des formes. Nous avons résolu ce problème avec succès par utilisation d'une technique simple et rapide qui est le re-échantillonnage de l'image. La distance entre arbres que nous avons utilisée est efficace et rapide en temps de calcul et donne un bon taux de reconnaissance (92%). Cette méthode a été essayée sur quelques caractères arabes manuscrits et s'avère très prometteuse. Ce système est multiforme, il résoud donc un problème de la méthode statistique. Un problème important qui pourrait être envisagé serait l'étude de la variabilité de cette méthode à la taille et à l'orientation des caractères.

IV. METHODE STRUCTURELLE (Structure morphologique)

Nous présentons à présent, un système de reconnaissance de texte arabe imprimé utilisant une méthode structurale basée sur les caractéristiques morphologiques du caractère arabe. Cette méthode est différente de la précédente, dans le sens où elle doit améliorer ses performances en traitant des caractères omni fontes. Elle consiste en l'extraction des caractéristiques principales (boucles et concavités selon les quatre directions), permettant le calcul des différentes classes de caractères ainsi que les caractéristiques secondaires (forme du caractère, ponctuation etc.) qui différencient les caractères de chaque classe.

Cette nouvelle approche procède donc en deux niveaux :

- Le premier niveau est la Classification qui utilise les paramètres non métriques pour grouper l'ensemble des caractères en classes.
- Le second niveau est l'Interprétation, purement heuristique, qui permet de séparer les caractères d'une même classe.

Avant la reconnaissance proprement dite, il est au préalable nécessaire d'extraire les caractéristiques de tous les caractères prototypes : Ceci constitue la phase d'apprentissage. La dernière opération constituant l'étape de décision, consiste à identifier le caractère par un calcul de distances spécifiques.

IV.1 Classification et interprétation des caractères

IV.1.1 Classification des caractères

Elle prend comme critères les concavités dans les différents sens, les boucles et leurs nombres qui représentent les caractéristiques principales (qui sont morphologiques).

Le choix de ces caractéristiques permet au système de travailler avec des caractères multi-tailles sans avoir recours à une normalisation éventuelle (qui pourra causer une déformation du caractère et un temps de traitement supplémentaire).

Ces caractéristiques nous permettent de calculer la classe de chaque caractère de la façon suivante[104] :

$$\text{Classe} = 4^0 \times H + 4^1 \times L + 4^2 \times R + 4^3 \times U + 4^4 \times B$$

avec: H: nombre de boucles (Hole)

L: nombre de concavités vers la gauche (Left)

R: nombre de concavités vers la droite (Right)

U: nombre de concavités vers le haut (Up)

B: nombre de concavités vers le bas (Below)

Exemple:

Caractères	H L R U B	Classe
ذ ز ر د 7	0 1 0 0 0	4
8 B ها	2 0 0 0 0	2
س ش	0 0 0 3 0	192
5 2 s z ح خ ج	0 1 1 0 0	20

Note: Pour les caractères arabes, les concavités vers le bas sont en général introuvables, si elles existent elles sont faibles en surface ; ce type de concavité n'est donc pas pris en compte mais il est très intéressant de le garder pour des systèmes OCR bilingues (Arabe-Latin).

IV.1.2 Interprétation et identification

La première étape permet la classification des caractères, chaque classe pouvant contenir un ou plusieurs caractères (voir tableau de l'exemple précédent). Pour reconnaître un caractère, il faudra le comparer à tous les caractères de sa classe en se basant sur les caractéristiques secondaires qui sont purement heuristiques.

Les caractéristiques secondaires utilisées dans notre système sont les suivantes :

1- La forme du caractère : Elle peut être carrée, allongée ou bien debout selon le rapport (largeur / hauteur) du cadre du caractère.

Exemple:

- Caractère carré comme : ذ، ء، د
- Caractère allongé comme : ت، ب، ف، ث
- Caractère debout comme : ع، أ، ك، ل، لا

On définit pour cela une variable **Forme** comme suit :

- Si le caractère est allongé : **Forme = 1**
- Si le caractère est carré : **Forme = 2**
- Si le caractère est debout : **Forme = 3**

2- Le taux de remplissage : Des quatre coins du plus petit rectangle qui contient le caractère (rectangle détecté après cadrage du caractère) : cette caractéristique nous donne une idée sur la répartition du caractère aux quatre coins (haut-droit, haut-gauche, bas-droit et bas-gauche). (voir figure IV.9)

On définit pour cela une variable **Corvar** de la façon suivante :

Si le coin (i) est rempli (i = 1... 4) alors:

$$\text{Corvar} = \sum_{i=1}^4 S_i 2^{i-1}$$

S_i (remplissage du coin i) = 0 ou 1.

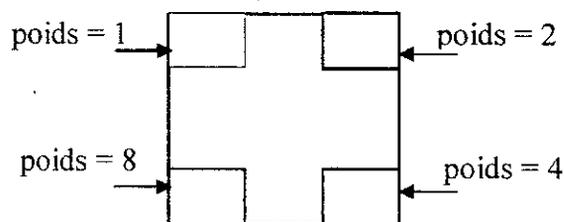


Figure IV.9: Corvar (taux de remplissage)

3- L'existence de points (y compris la "Hamza") : Nous définissons une variable **Expoint** qui prend la valeur "1" pour les caractères ponctués et la valeur "0" pour les caractères ne possédant pas de points diacritiques.

4- La position des points (y compris la "Hamza") : Nous définissons pour cela une variable **Pospoint** qui prend les valeurs suivantes:

- **Pospoint = 1** pour la position des points en haut du caractère.
- **Pospoint = 2** pour la position des points au milieu du caractère.
- **Pospoint = 3** pour des points en bas du caractère.

ك

ج

ي

Pospoint = 1

Pospoint = 2

Pospoint = 3

5- Le nombre de points défini par la variable Nombpoint qui prend les valeurs suivantes :

- **Nombpoint = 1** le caractère possède 1 point.
- **Nombpoint = 2** le caractère possède 2 points.
- **Nombpoint = 3** le caractère possède 3 points.
- **Nombpoint = 4** le caractère possède une Hamza.

Note: Si la 3^{ème} caractéristique secondaire est nulle (Expoin=0, alors la 4^{ème} et la 5^{ème} sont obligatoirement nulles (c'est à dire Pospoint=0 et Nombpoint=0).

Toutes ces caractéristiques permettent de définir un vecteur d'attributs qui caractérise chaque prototype de la même classe.

L'étape d'extraction des primitives principales et secondaires nous donne donc : la classe du caractère et un vecteur d'attributs V_i où :

- $V[1]$ = Forme.
- $V[2]$ = Corvar.
- $V[3]$ = Expoin.
- $V[4]$ = Pospoint.
- $V[5]$ = Nombpoint.

IV.1.3 Extraction des caractéristiques :

1- Caractéristiques principales

Ce sont les primitives qui nous permettent de regrouper les caractères en classes, donc leur choix est très important. On remarque très facilement que les caractères arabes sont formés d'un agencement de boucles et de concavités (sauf quelques cas particuliers) ; on estime que ce choix est convenable. Une concavité est une partie d'un caractère qui est courbée vers l'intérieur, elle peut être dirigée vers le haut, le bas, la droite ou la gauche.

Détection des caractéristiques principales :

Comme nous l'avons déjà noté, la difficulté du choix des primitives est de même grandeur que leur extraction ; il faudra donc utiliser une méthode d'extraction très efficace pour pouvoir trouver la classe du caractère.

Principe de la méthode :

Après avoir cadré le caractère (trouvé le plus petit rectangle qui le contient à partir de ses coordonnées obtenues pendant l'étape de segmentation) (voir figure IV.10), on procède à un balayage de toute la surface contenant le caractère horizontalement puis verticalement. Pendant ce balayage on sauvegarde les coordonnées des lignes (respectivement des colonnes) qui présentent quatre transitions au moins (passage de 0 à 1 ou de 1 à 0) car l'existence de 4 transitions ou plus signifie qu'on a croisé le caractère 2 fois ou plus, donc il y a une possibilité d'existence d'une concavité ou d'une boucle, (voir figure IV.11).

Si le caractère est bien cadré alors le nombre de transitions de chaque ligne (respectivement colonne) est pair, l'indice des transitions (0-1) est impair et l'indice des transitions (1-0) pair. On prend chaque ligne (respectivement colonne) sauvegardée suite aux balayages et on cherche le centre des segments $[T_{2n}, T_{2n+1}]$ tel que T_i représente la $i^{\text{ème}}$ transition.

A la fin de cette étape on obtient un ensemble de points (appelés points essentiels) qui seront à la base de la détection des concavités et des boucles (voir figure IV.12).



Figure IV.10: Encadrement du caractère

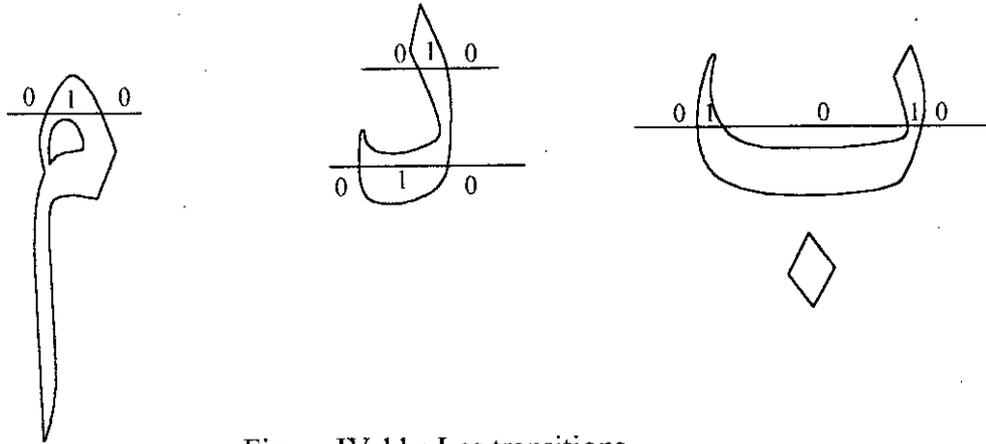


Figure IV.11 : Les transitions



Figure IV.12 : Les points essentiels

Le principe est le suivant :

A partir de chaque point essentiel on essaye de tirer une droite dans les huit sens :

Si on croise le caractère dans un sens, on peut conclure qu'il n'existe pas de concavité dans ce sens, et ainsi de suite.

Avec les huit sens (haut, haut-droit, haut-gauche, droit, gauche, bas, bas-droit, bas-gauche) , on construit une fonction booléenne liée à ce point et on décide à quel type de concavité appartient ce point.

Exemple:

- 1- Si on croise le caractère dans les huit sens, alors le point considéré appartient à une boucle.
- 2- Si on croise le caractère dans les sens : bas, bas-droit, bas-gauche alors on est en présence d'une concavité vers le haut.

Cette étape va nous donner le nombre de concavités dans les quatre sens et celui des boucles, mais il existe des points qui peuvent appartenir à une même concavité ou à une même boucle ; nous

avons prévu pour cela une troisième étape qui permettra de donner le nombre exact des boucles et des concavités. Dans cette étape, on essaie de faire une liaison entre chaque deux points essentiels [110] ; s'ils sont liés (absence de tout point allumé, appartenant au caractère, entre ces deux points) alors ils appartiennent à la même concavité, et si l'un d'eux représente une boucle et l'autre une concavité quelconque alors la boucle est liée avec la concavité donc il y a une ouverture entre eux, on conclut que la boucle détectée est une fausse boucle.

Pour cela, on relie d'abord les boucles entre-elles, ensuite les concavités de même type puis les concavités de types différents entre-elles et enfin les boucles avec chaque type de concavité.

A la fin de cette étape, nous obtenons le nombre réel de boucles et celui de chaque type de concavité (voir figure IV.13).

Cette méthode s'est avérée efficace et a donné de bons résultats, néanmoins elle donne parfois des résultats erronés qui sont dus non pas à la méthode mais à la forme du caractère ; on peut rencontrer ainsi :

- de fausses boucles difficiles à éliminer.
- Des concavités qui ne sont pas bien séparées.
- La difficulté de détection de très faibles concavités.

Nous avons remarqué aussi que cet algorithme est sensible au bruit ; nous avons prévu pour cela une procédure de bouchage de trous dans le caractère et d'élimination des points isolés par utilisation des opérateurs morphologiques (érosion et dilatation) prévus dans l'étape de pré-traitement.

2- Caractéristiques secondaires

L'extraction des caractéristiques secondaires est très importante car celles-ci servent à séparer les caractères de la même classe.

Nous avons choisi cinq caractéristiques secondaires citées précédemment. Nous donnons dans ce qui suit la méthode d'extraction de chacune d'elles.

➤ Extraction de la première caractéristique (*forme du caractère*) :

Pour extraire cette caractéristique, on mesure la largeur (W) et la hauteur (H) du caractère puis on calcule le rapport :

$$\text{Rapport1} = W/H$$

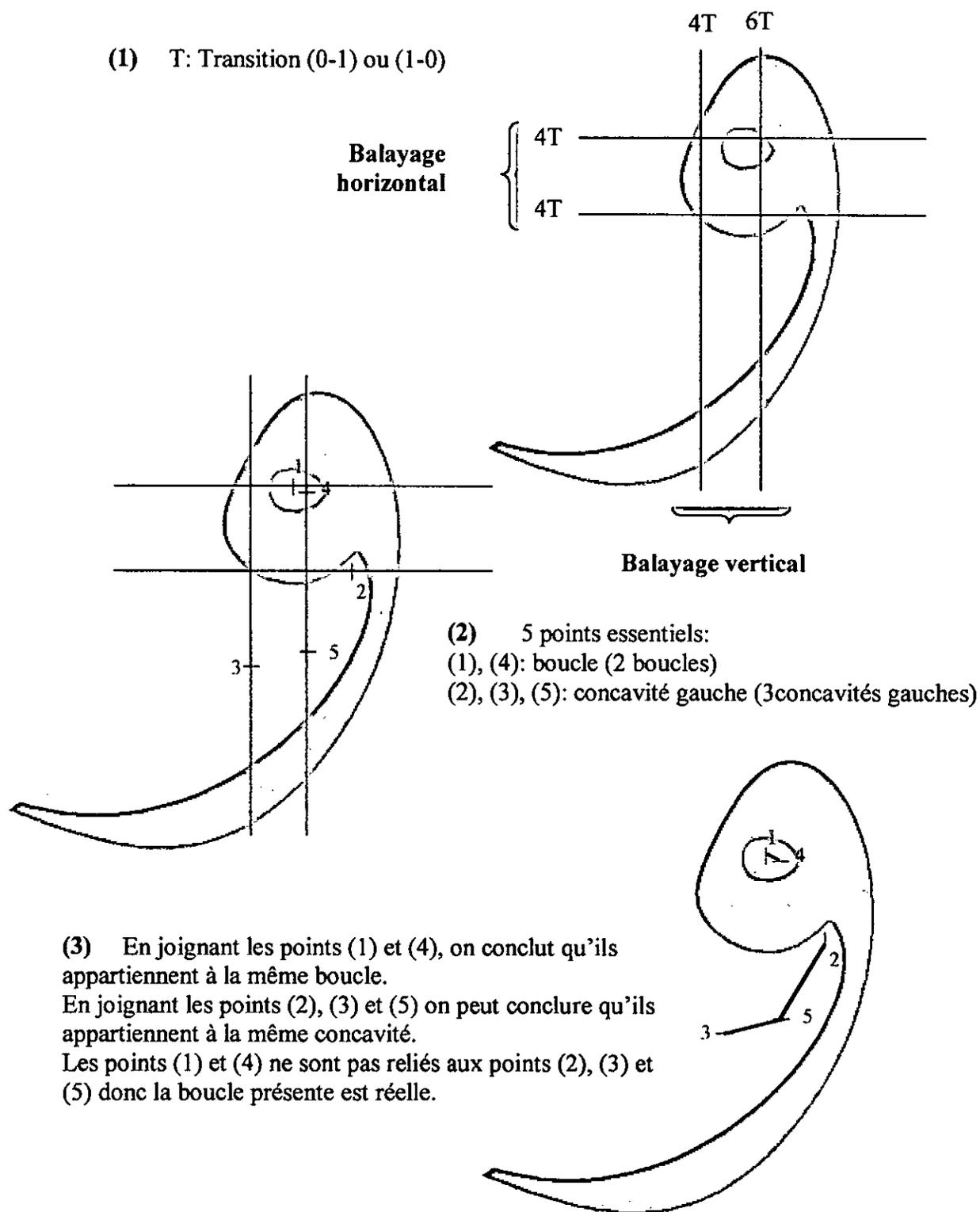


Figure IV.13 : Exemple de détection de concavités et boucles

- Si $Rapport1 > Seuil2$, alors on déduit que le caractère a une forme horizontale ou allongée donc **Forme = 1**
- Si $Seuil1 < Rapport1 < Seuil2$, alors le caractère a une forme carrée et **Forme = 2**
- Si $Rapport1 < Seuil1$, alors le caractère a une forme verticale ou debout et **Forme = 3**

Les seuils sont estimés après plusieurs expériences à : seuil 1 = 0.8 et seuil 2 = 1.2

➤ **Extraction de la seconde caractéristique (Taux de remplissage des 4 coins) :**

La variable à calculer est appelée **Corvar**, W et H étant la largeur et la hauteur respectives du caractère, on considère quatre carrés de dimensions :

- a) $(W / 4) * (W / 4)$ si la forme du caractère est debout.
- b) $(H / 4) * (H / 4)$ si la forme du caractère est allongée ou carrée.

Ces carrés sont pris dans chaque coin du rectangle qui encadre le caractère.

On calcule la surface de chaque carré notée S, ainsi que la surface pleine de chaque carré notée S_p .

Le taux de remplissage du carré considéré sera donné par **Rapport2** tel que :

$$\text{Rapport2} = S_p / S$$

Si $Rapport2 \geq \text{seuil3}$, on considère que le coin est plein et on attribue à une variable booléenne appelée Taux la valeur 1 d'où : **Taux = 1**

Sinon, il est vide et on aura : **Taux = 0**

Après plusieurs tests, nous avons pris $Seuil3 = 0.5$.

On effectue ceci pour les quatre coins, ce qui nous donne les quatre Taux suivants :

Taux (haut-gauche), Taux (haut-droit), Taux (bas-gauche) et Taux (bas-droit).

On évalue ensuite la variable **Corvar** de la façon suivante:

$$\text{Corvar} = \text{Taux (haut-gauche)} + 2 \times \text{Taux (haut-droit)} + 4 \times \text{Taux (bas-droit)} + 8 \times \text{Taux (bas-gauche)}$$

Cette variable sera comprise entre 0 et 15, d'où : $0 \leq \text{Corvar} \leq 15$

➤ **Extraction de la troisième caractéristique (existence de points ou de la hamza)**

Cette caractéristique comme les autres caractéristiques est très importante dans l'identification des caractères. Dans certains cas elle est très difficile à extraire, pour cela on procède en deux étapes : si

dans la première étape on détecte les points ou la hamza, alors on ne procède pas à la seconde étape sinon on la fait.

Dans la première étape on effectue un balayage horizontal (de droite à gauche et de haut en bas) de la surface du rectangle qui contient le caractère.

Dans la première ligne de balayage, on doit rencontrer au moins un pixel allumé appartenant au caractère, on continue le balayage jusqu'à la fin du caractère. Si on rencontre une ligne où tous les pixels sont éteints alors on est en présence d'un vide ; en continuant le balayage si on rencontre un deuxième amas de pixels, on pourra conclure que le caractère testé est composé de deux entités.

On peut rencontrer un vide à nouveau et donc le nombre d'entités devient trois, mais généralement un caractère arabe sans vocalisation ne peut contenir plus de deux entités, d'où :

$$\text{Nombre d'entités} = \text{nombre de vides} + 1$$

Si après balayage de tout le caractère on ne rencontre aucun vide (ou aucune ligne où tous les pixels sont éteints), on ne peut rien conclure sur le nombre d'entités du caractère car il se peut que le caractère recouvre les points ou la hamza (c'est le cas du caractère Djim (ج) où le point se trouve à l'intérieur du corps du caractère, ou encore celui du caractère Kef isolé (ك) pour lequel la hamza et le corps du caractère ne sont pas bien séparés.

Si le nombre d'entités détecté après la première étape est égal à un, on ne peut rien conclure, on passe alors à la deuxième étape. Dans cette dernière, on détecte l'existence des points à l'intérieur du caractère. On doit pour cela, balayer toute la surface du caractère par un carré ; si la périphérie du carré est éteinte et qu'il existe au moins un pixel allumé dans ce carré, alors on peut dire qu'il existe une entité séparée du corps du caractère et donc que le nombre d'entités est égal à deux ou encore que le caractère possède des points ou une hamza, sinon le caractère n'est pas ponctué.

On définit une variable appelée *Expoin* comme suit :

- Si le nombre d'entités est égal à deux, alors **Expoin = 1**
- Si le nombre d'entités est égal à un, alors **Expoin = 0**

➤ **Extraction de la quatrième caractéristique (*position des points ou de la hamza*)**

En comparaison avec la caractéristique précédente, la position des points est plus simple à détecter. Si lors de la première étape de détection des entités on en trouve deux, on calcule la surface de chacune d'elles et il est clair que l'entité la plus petite en surface représente les points ou la hamza alors que l'autre entité représentera le corps du caractère. Comme décrite plus haut, une variable *Pospoint* est définie et vaut 1 (points en haut), 2 (points au milieu) ou 3 (points en bas).

Si la plus petite entité (en surface) se trouve au dessus de l'autre entité alors les points ou la hamza se trouvent en haut du corps du caractère, nous aurons alors : **Pospoint = 1**

Dans le cas contraire nous aurons : **Pospoint = 3** , c'est à dire que les points diacritiques sont en dessous du corps du caractère.

Si la deuxième entité est détectée après la deuxième étape de détection d'entités, alors on tire à partir du milieu de cette entité une droite vers le haut et une autre vers le bas du corps du caractère ; si on croise le caractère dans le sens haut alors les points sont en bas du caractère et nous aurons : **Pospoint = 3** .

Si on croise le caractère dans le sens bas, alors les points sont en haut du caractère et nous aurons : **Pospoint = 1**.

Si on croise le caractère dans les deux sens, alors le point est au milieu du caractère d'où :

Pospoint = 2.

Si le nombre d'entités est égal à un après les deux étapes de détection d'entités, alors les points n'existent pas d'où : **Pospoint = 0**.

➤ **Extraction de la cinquième caractéristique (nombre de points ou la hamza)**

Cette cinquième caractéristique est à extraire si **Expoint = 1**.

La variable Nombpoint peut prendre les valeurs déjà citées précédemment c'est à dire 1, 2, 3 ou 4.

Cas de deux points :

Les critères donnés ci- dessous ont été évalués après plusieurs essais. On définit :

$$\text{Rapport3} = W_p / H_p$$

Où W_p et H_p sont respectivement la largeur et la hauteur de l'amas qui constitue les points.

On définit aussi une variable Maxtrv qui représente le nombre maximal de transitions verticales (0-1 ou 1-0) de l'entité amas de points.

Si (Rapport3 \geq 1.1) et (Maxtrv = 4) alors Nombpoint = 2

Cas de trois points :

On calcule les surfaces haute et basse de l'entité qui représente les points, elles sont notées respectivement SurfH et SurfB.

On calcule aussi la surface pleine SurfPl qu'occupe l'amas de points ainsi que la surface Surf du rectangle qui contient les points (après cadrage).

On définit les rapports suivants :

$$\text{Rapport4} = \text{SurfB} / \text{SurfH}$$

Ce rapport nous donne une idée sur la distribution des points en haut et en bas du rectangle qui les contient.

$$\text{Rapport5} = \text{SurfPl} / \text{Surf}$$

Ce rapport donne quant à lui, le taux de remplissage du rectangle déjà défini.

Les 3 points ont une distribution de surface plus importante en bas qu'en haut du rectangle, et le taux de remplissage de ce rectangle est faible, d'où :

Si (Rapport4 \geq 1.7) et (Rapport5 \leq 0.6) alors Nombpoint = 3

Cas de la Hamza :

La hamza a une forme carrée presque comme le point, mais elle est caractérisée par un nombre de transitions verticales (0-1 ou 1-0) élevé, on calculera donc ce nombre noté NombTr.

Si (0.8 \leq Rapport3 \leq 1.2) et (NombTr \geq 4) alors Nombpoint = 4

Cas du point :

Dans le cas du point, le nombre de transitions verticales est égal à 2, c'est ce qui le différencie de la hamza.

Si (0.9 \leq Rapport3 \leq 1.1) et (NombTr = 2) alors Nombpoint = 1

Tous les seuils ont été fixés après plusieurs essais.

Nous résumons dans ce qui suit l'étape d'extraction des caractéristiques principales et secondaires:

- 1- On effectue la lecture du fichier contenant le caractère et on obtient une matrice IM[W,H] où W et H représentent respectivement la largeur et la hauteur de l'image.
- 2- On effectue un cadrage du caractère.
- 3- On extrait les caractéristiques principales du caractère et on calcule sa classe.
- 4- On fait un balayage horizontal du caractère, si on détecte une ligne vide alors il existe deux entités ; d'où Expoin = 1

Calcul des surfaces des deux entités SurfB et SurfH.

- Si SurfH > SurfB alors Pospoin = 3 (points en bas) sinon Pospoin = 1 (points en haut).

Identification des points (1point: Nombpoint=1; 2 points: Nombpoint =2 ; 3points: Nombpoint =3; Hamza: Nombpoint = 4).

- Sinon : On fait un balayage de toute la surface du caractère avec un carré.

Si on trouve une position du carré où son périmètre est complètement éteint et dans lequel au moins un pixel est allumé ; alors Expoint = 1.

- Si à partir des points vers le haut on ne rencontre pas le caractère, alors Pospoint = 1 (*haut*)
- Si à partir des points vers le bas on ne rencontre pas le caractère, alors Pospoint = 3 (*bas*)
- Sinon Pospoint = 2 (*milieu*)

Identification des points (1point : Nombpoint=1; 2points: Nombpoint=2 ; 3 points: Nombpoint=3 ; Hamza: Nombpoint = 4)

Sinon le caractère ne contient ni points ni hamza d'où : Expoint = 0

5- Calcul des taux de remplissage des quatre coins.

Corvar = coin (haut-gauche)+2*coin(haut-droit)+4*coin (bas-droit)+8*coin (bas-gauche).

6- Calcul de Rapport1 = W / H.

- Si Rapport1 > 1.2 alors Forme = 1 (caractère allongé).
- Si 0.8 < Rapport1 < 1.2 alors Forme = 2 (caractère carré).
- Si Rapport1 < 0.2 alors Forme = 3 (caractère debout).

7- On donne un nom au caractère (ou son code ASCII).

8- On donne la position du caractère (début, milieu, finale, isolée).

On obtient à la fin un vecteur de caractéristiques $V(i)_{i=1\dots 8}$ qui a la forme suivante :

V(1)	V(2)	V(3)	V(4)	V(5)	V(6)	V(7)	V(8)
------	------	------	------	------	------	------	------

où :

V(1)=Forme, V(2)=Corvar, V(3)=Expoint, V(4)=Pospoint, V(5)=Nombpoint, V(6)=NomCar, V(7)=PosCar, V(8)=Classe.

IV.2 Apprentissage

L'apprentissage utilisé est supervisé, c'est à dire que des caractères prototypes de référence sont rangés dans un dictionnaire, sous forme de vecteurs contenant leurs caractéristiques et leurs noms ou codes (voir Figure IV.14). Une des flexibilités de ce logiciel est la possibilité d'ajouter au dictionnaire de nouveaux caractères (apprentissage dynamique).

La méthode citée précédemment pour l'extraction des caractéristiques est utilisée.

Un dictionnaire est alors établi, il est composé de deux parties :

- L'entête qui est formée de groupes de 4 octets tel que :

- le premier octet contient le numéro de classe,
- le second contiendra le nombre de prototypes (Npr) que contient cette classe,
- le troisième et le quatrième octet contiennent respectivement les poids faible et fort du pointeur vers le début de la classe considérée dans le fichier et cela à partir du début du dictionnaire.

➤ Les classes : cette partie contient les classes où chaque prototype est décrit sur huit octets.

Ces octets représentent les éléments du vecteur caractéristiques V.

- Le premier octet contient la variable Forme = {1,2,3} ;
- le 2^{ème} octet, la variable Corvar = {0,1,...,14,15} ;
- le 3^{ème} octet, la variable Expoint = {0,1} ;
- le 4^{ème} octet, la variable Pospoint = {0,1,2,3} ;
- le 5^{ème} octet, la variable Nombpoint = {0,1,2,3,4} ;
- le 6^{ème} octet, la variable NomCar c'est à dire le nom du caractère ou son code ASCII ;
- les 7^{ème} et 8^{ème} octets sont réservés.

Dés la fin de l'entête, commence la deuxième partie qui contient les classes.

Cette classification permettra de comparer le caractère inconnu avec un nombre très réduit de prototypes (ceux appartenant à la même classe que le caractère inconnu).

IV.3 Reconnaissance et Décision

C'est la dernière étape dans la chaîne d'un système OCR, elle représente le bloc décisif qui donne une des trois solutions possibles :

- le caractère est reconnu (identifié à un seul prototype),
- le caractère est rejeté (n'est identifié à aucun prototype),
- le caractère est ambigu (identifié à un plusieurs prototypes).

La règle de décision est basée sur la notion de distance calculée entre le caractère inconnu et les caractères prototypes ayant la même classe que lui. Le caractère inconnu est reconnu comme étant le caractère prototype qui donne la distance minimale, c'est à dire qui lui ressemble le plus.

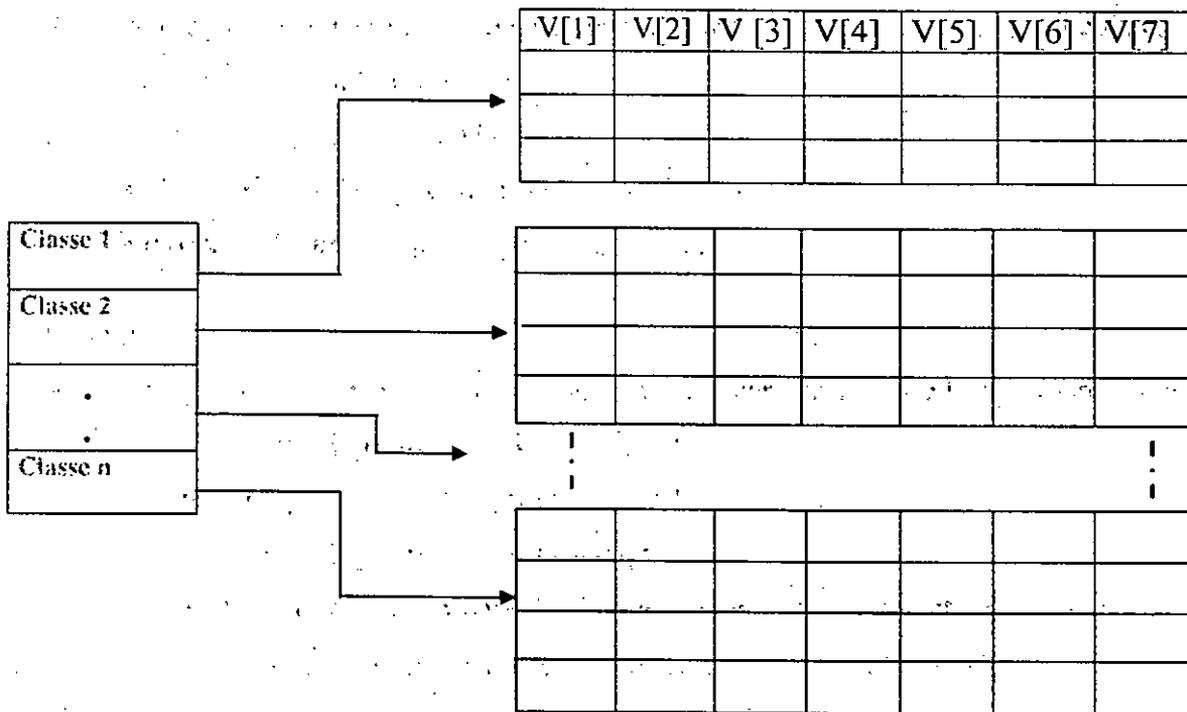


Figure IV.14 : organisation du dictionnaire

Nous utilisons deux distances D_1 et D_2 qui considèrent les deux premières caractéristiques secondaires (Forme et Corvar) telles que :

a) $D_1 = \text{abs}(V_x(1) - V_p(1))$

avec:

$V_x(i)$: élément d'ordre (i) du vecteur de caractéristiques du caractère x à reconnaître.

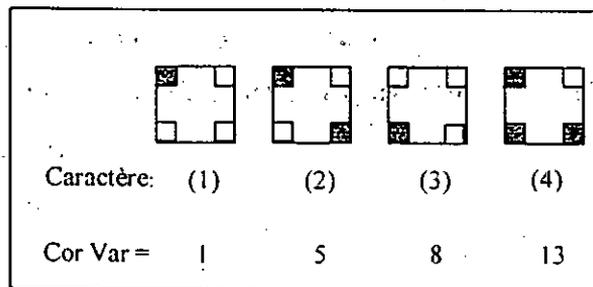
$V_p(i)$: élément d'ordre (i) du vecteur de caractéristiques du caractère prototype p.

Pour que la forme du caractère x ressemble à celle du prototype p il faut que : $D_1 \leq 1$.

(Ce qui signifie que la forme debout ou allongée peut ressembler à la forme carrée mais la forme allongée ne peut ressembler à la forme debout)

b) la deuxième caractéristique est CorVar ; Dans ce cas, la distance est binaire. considérons

la Figure suivante :



<i>Paramètres</i>	<i>Valeur</i>
Nombre de neurones de la couche d'entrée	22
Nombre de neurones de la couche cachée	15
Nombre de neurones de la couche de sortie	205
Momentum (α)	0.75
β	0.6
P_e	0.8
β_e	0.04
<i>cste</i>	0.05
<i>Seuil</i>	0.00001

V.3 Architecture du réseau adopté

Le réseau adopté pourra être représenté schématiquement par la Figure IV.15 :

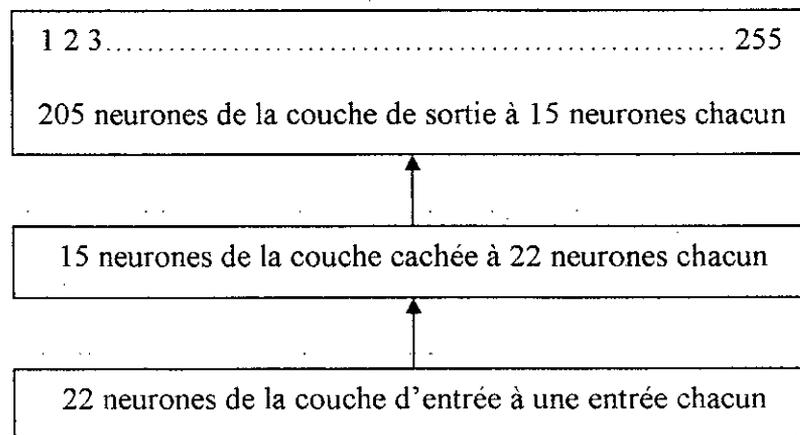


Figure IV.15 : Architecture du réseau de neurones adopté

V.4 Apprentissage

L'apprentissage consiste à présenter chaque caractère (sous forme de vecteur de caractéristiques) au réseau qui se chargera de faire le reste c'est-à-dire l'activation des neurones, le calcul des sorties et l'ajustement des poids synaptiques.

Nous donnons ci après la procédure de Reconnaissance élaborée.

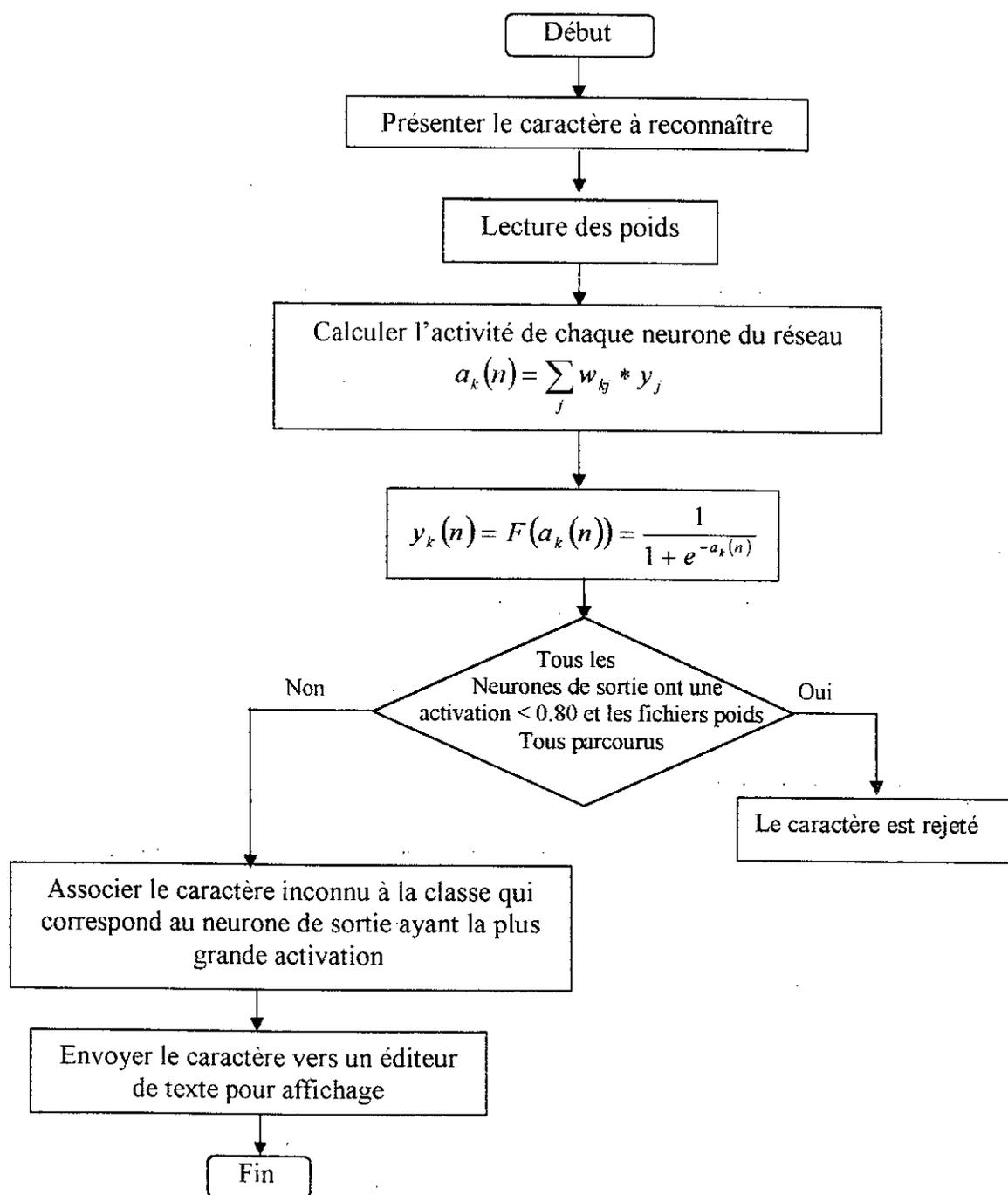


Figure IV.16 : Procédure de Reconnaissance par réseau de neurones

V.5 conclusion

Nous avons, par l'introduction des réseaux de neurones, essayé d'améliorer les résultats de la reconnaissance obtenus par les méthodes précédentes et notamment celle utilisant les paramètres morphologiques ; car la rétropropagation du gradient dans les réseaux multicouches corrige ou plutôt minimise l'erreur en la prenant en considération dans la prochaine itération jusqu'à la rendre minimale.

Les résultats obtenus par cette méthode ne sont pas très différents de ceux obtenus par la méthode qui la précède (taux de reconnaissance de 98.6% pour les mêmes textes que ceux considérés dans les méthodes précédentes). Ceci est lié au choix des paramètres du réseau qui restent pour la plupart expérimentaux. Dans cette dernière approche, l'apprentissage est très coûteux en temps de calcul mais en revanche, il permet de prendre des décisions plus rigoureuses.

CHAPITRE V

Résultats et interprétations

- I. Introduction**
- II. Comparaison des différentes méthodes étudiées**
- III. Quelques résultats obtenus en reconnaissance de documents**
- IV. Présentation du logiciel**
- V. Déroulement des différentes phases du logiciel**
- VI. Conclusion**

I. Introduction

L'objectif de notre travail vise la réalisation d'un système de reconnaissance de texte arabe imprimé. Dans cette application nous utilisons des images texte scannées en noir et blanc avec une résolution de 200dpi.

A partir d'un document composite, il s'agira de segmenter les différents blocs puis de reconnaître les blocs texte pour les acheminer vers le système de reconnaissance de caractères que nous avons réalisé par utilisation de différentes approches.

Nous commencerons dans ce présent chapitre par donner un tableau récapitulatif des caractéristiques de chaque méthode implémentée, puis nous présenterons les résultats de la segmentation et reconnaissance des blocs de document. Une étude descriptive de l'interface OCR sera suivie des résultats obtenus par le système de reconnaissance retenu, à savoir celui utilisant la description morphologique du caractère.

II. Evaluation des différentes méthodes étudiées

Un tableau récapitulatif des caractéristiques des différentes approches étudiées permettra de donner une évaluation des trois méthodes implémentées à savoir :

- *la méthode statistique* utilisant les 7 moments invariants, comme descripteurs de contours, et qui nécessite une étape de normalisation de la taille du caractère segmenté. La détection de contours et la normalisation sont deux étapes de prétraitement supplémentaires qui ralentissent le processus de reconnaissance. En plus cette méthode ne peut traiter qu'une seule fonte à la fois, le logiciel se trouve ainsi limité.
- *la méthode structurelle utilisant la structure d'arbres* et nécessitant une étape de squelettisation puis d'échantillonnage du squelette pour permettre un suivi correct du tracé du caractère. Ces deux étapes de prétraitement ralentissent le processus de reconnaissance. Cette méthode est multi fonte, elle nécessite donc l'apprentissage des différentes fontes que l'on désire reconnaître et par conséquent l'espace mémoire utilisé par la base de données est important.
- *la méthode structurelle syntaxique* qui utilise des caractéristiques morphologiques pour la classification et d'autres caractéristiques heuristiques qui permettent de différencier les caractères de la même classe. Les caractéristiques morphologiques choisies permettent de rendre le système bilingue par simple apprentissage d'une fonte latine. Ce système est omnifonte car nous ne sommes pas obligés d'apprendre à notre système toutes les fontes, une seule suffit car les caractéristiques sont descriptives et ne tiennent compte ni de la taille ni du style du caractère.

Nous présentons sur le tableau suivant les taux de reconnaissance obtenus par les différentes méthodes pour le même texte comportant 82 caractères.

Approches	Type	Taux de Reconnaissance	Observations
Méthode Statistique	mono fonte	99.3% (Fonte apprise)	<ul style="list-style-type: none"> - Nécessite une normalisation de la taille. - Temps d'apprentissage très court - Ne convient pas au manuscrit.
Méthode des arbres	multi fonte	94.2% (Fonte apprise)	<ul style="list-style-type: none"> - La squelettisation et le re-échantillonnage ralentissent la reconnaissance. - base de données très importante. - convient au manuscrit
Méthode morphologique	omnifonte	98.9% (Fonte non apprise)	<ul style="list-style-type: none"> - S'adapte au bilingue et au manuscrit. - Invariante au style et à la taille. - La base de données supporte une seule fonte prototype
Méthode connexionniste	omnifonte	96.3% (Fonte non apprise)	Mêmes observations que la méthode précédente mais, demande un temps d'apprentissage très important.

Tableau V.1 : Caractéristiques des différentes méthodes implémentées

Nous pouvons constater qu'aucune méthode n'est vraiment meilleure que l'autre car les taux de reconnaissance ne diffèrent pas tellement d'une approche à l'autre.

Si nous avons à considérer un document administratif qui utilise la même fonte, l'approche statistique est conseillée car elle est très rapide et a un taux de reconnaissance compétitif.

Par contre, si nous traitons plusieurs fontes à la fois la méthode morphologique est conseillée avec ou sans utilisation des réseaux de neurones.

Le taux de reconnaissance obtenu par utilisation des réseaux de neurones en hybridation avec les caractéristiques morphologiques pourrait être considérablement amélioré par un meilleur choix des paramètres du réseau qui sont restés dans notre application purement expérimentaux.

III. Quelques résultats obtenus en reconnaissance de documents

Les figures V.1 à V.5 illustrent les différentes étapes de segmentation et de reconnaissance d'un document complexe.

La figure V.1 représente un document scanné avec une résolution de 200dpi.

La figure V.2 représente la segmentation du document en ses différents blocs (ici blocs de texte et bloc image) non encore reconnus.

La figure V.3 représente la segmentation de chaque bloc séparément, à cet instant nous pouvons détecter ou reconnaître les blocs texte grâce à la segmentation en lignes du paragraphe de texte.

La figure V.4 reproduit chaque type de bloc reconnu et lui affecte un niveau de gris différent ; dans notre exemple nous avons deux types de blocs : texte et image.

La figure V.5 procède à la deuxième étape de segmentation du texte qui n'est pas nécessaire à ce niveau (ARD).

IV. Présentation du Logiciel

IV.1 L'environnement de programmation

Notre application a été réalisée dans un environnement Windows, en langage Visual C++ version 5 et Delphi 4, la puissance de ces deux langages se révèle dans les aspects suivants :

- Delphi 4 est un langage qui facilite la construction des interfaces utilisateur conviviales (menus, boîtes de dialogue, etc.)
- Visual C++ permet de réaliser des applications avec des résultats pointilleux par leur précision. Le réseau de neurones (apprentissage et reconnaissance) a été implémenté dans ce langage.

Pour la reconnaissance, on s'intéresse seulement au code du caractère reconnu qui permettra d'afficher le caractère ou de l'envoyer sur un éditeur.

Les prétraitements ont été réalisés en langage delphi 4.

إن نظام المشتريات الحالي والذي يتعامل مع لقاحات تستخدم منذ وقت طويل يؤكد على ابتياج بضائع فائضة. وهو يستفيد من كل البواعث المتاحة لدى منتجي اللقاحات كي يبيعوا منتجاتهم بأسعار تساوي أو تقترب، أو حتى أدنى، من التكلفة الحقيقية للإنتاج. وتتخذ النجاحات المرموقة التي يحققها البرنامج صفة الاستمرار، وما إن يتم وضع أنظمة الإيصال في مكانها تماما، حتى يصبح بمقدور البرنامج الموسع للتمنيع أن يحقق الهدف في إيصال اللقاحات الستة الحالية إلى كل أطفال العالم. وعند هذه النقطة سيقدّم البرنامج الموسع للتمنيع ما يزيد عن بليرين جرعة سنويا.

من المؤكد أن البرنامج الحالي يواجه تحديات. إن المسؤولين عن البرنامج الموسع للتمنيع ومن الحملة من أجل حماية الطفل يركزون على حل مشكلات تتعلق بكيفية القيام بحملات في مناطق لم يعط الأطفال فيها بعد اللقاحات الستة الكائنة وعن كيفية ضمان استمرار وصول هذه اللقاحات إلى كل جيل جديد من الأطفال. وذلك بعد أن تكون خدمات البرنامج قد شملت في النهاية كل أنحاء العالم. كما أن البرنامج الموسع للتمنيع يواجه باستمرار خطر ارتفاع الأسعار وعلاوة على ذلك، فإن اللقاحات المتوفرة حاليا تعاني بعض العيوب. فمثلا تفقد هذه اللقاحات فاعليتها عندما تتعرض للحرارة، وهو خطر يبرز في البلدان النامية حيث يكون التبريد في معظم الأحيان غير كاف. فالبرنامج الموسع للتمنيع في حاجة ملحة للقاحات تتحمل تأثير الحرارة ولاخرى قادرة أن تزيد من فاعلية وعن كفاءة البرنامج بطرائق أخرى، فلقاح مضاد للحصبة مثلا يمكن أن يعطى عند الولادة أو بعدها بقليل بحيث يمنع الوليد حمى مبركة.

وعلى الرغم من أنها حاسمة، فالعضلة التالية عن كيفية توفير لقاحات أخرى لم تول إلا اهتماما ضئيلا حتى عهد قريب. وبقينا، فإنها هي التي نحتاج بشدة إلى التفكير المدع والاستراتيجيات الجديدة. وإن تؤدي اللقاحات الجديدة إلى إنقاذ عدد لا يحصى من الأرواح فحسب، بل إن إتاحتها ستساعد على ديمومة انخفاص القادة السياسيين في الاقطار النامية لترسيخ النظم اللازمة التي تتلاقى مع أهداف البرنامج



مروماني شرق باكستان تم تحزته كصنوبر للكوئيرا (البيضة). وفي كل مكان في العالم الناس نجد ان الماء الذي يؤخذ من البحيرات والغدران والأنهار، والموت بالفضلات البشرية والمصادر الأخرى لعدد من التلوثات الضامة، والذي يستعمل في الشرب والطهو، يساعد على انتشار الكوئيرا وعند اخرون الأمراض كالتهاب الكبد والفيروسات المستعمرة وباء الشيكلات، والحمى التيفية. كما أن المستوى المنخفض والسكن المتكاثرة والحشرات الحاملة للأمراض (نقلات الجرثيم)، كلها عوامل تعزز انتشار المرض. ويمكن توفي عدد كبير من الأمراض بتلقيح برامج تناول كمية مصادر مياه الشرب وتحسين المستوى الصحي وشروط السكن والتخلص من جماعات الحشرات الناقلات للعوامل المرضية. نصف إلى ذلك أن برامج المعالجة غالبا ما تحول دون حدوث الإعاقات أو الموت. بيد أن كاتبي الطاقة يشيران إلى أن توسيع البرامج الخاصة بالتمنيع هي، على المدى القصير، أقل البرامج تكلفة وليس بالطرائق لإنقاذ حياة الناس.

موجودة في كندا وأوروبا واليابان؛ ذلك أن السوق المحلية في الولايات المتحدة على درجة من الضخامة بحيث لا تجد شركات الأدوية الأميركية حاجة ملحة للمنافسة في جهات أخرى أو أن تعمل مع البرنامج الموسع للتمنيع. وأيضا فإنها أثبتت عن عزيمتها لبيع اللقاحات لهذا البرنامج بكلفة منخفضة لأن الكونغرس قد انتقدها لعدم عرضها أسعارًا منخفضة مماثلة في موطنها.

واقدمتُ نظام استرجاع العروض عام 1988 اليونيسيف ومنظمة الصحة للاقطار الأميركية من شراء اللقاحات بسعر متوسط منخفض فرق العادة بلغ خمسة سنتات للجرعة الواحدة، وهذا يعني تكلفة خمسين سنتا لشراء كل الجرعة لكل الأمراض الستة. ويتفق البرنامج الموسع للتمنيع أكثر من عشرة أضعاف هذا المبلغ لكل طفل على البنود الأخرى غير اللقاح؛ كأجور نقل ورواتب الأطر (الكوادر) التي ستعطي اللقاح وتكلفة التجهيزات الخاصة بالتبريد والتي تحمي اللقاح في (سلسلة باردة) cold chain.

وعلى الرغم من أن البرنامج الموسع للتمنيع قد استطاع أن يقيد من الدوافع المتفاوتة لعدد كبير من منتجي اللقاحات كي يحصلوا على أسعار منخفضة قبل عام 1983، فقد نجح البرنامج في ذلك العلم في تخفيض نفقته كذلك باستهلاله نظاما لاستدراج العروض، وعمل الأخص (اليونيسيف) UNICEF، الذي يبتاع معظم اللقاحات التي تستخدم في أفريقيا وبعض اللقاحات التي تستخدم في آسيا، و(منظمة الصحة للاقطار الأميركية) Pan American Health Organization (PAHO) التي تشتري اللقاحات لأميركا اللاتينية، لتلقيح عرضا لكل لقاح على حدة من عديد من المنتجين، وتبرمان عقد شراء مع المنتج المؤهل الأرخص سعرا. (وتركز منظمة الصحة العالمية اهتماماتها حاليا على النواحي الطبية والإدارية وعلى تدريب الأطر أكثر من تركيزها على عمليات الحصول على اللقاحات).

منذ استهلال برنامج استدراج العروض كانت كل الشركات المهمة والمنتجة للقاحات

Figure V.1 : Image originale (document scanné de El-Ouloum : Scientific-American en Arabe, vol. 6, n°5, Mai 1989). (Résolution = 200 dpi, dimension = 2083x1590 pixels).

إن نظام المشتريات الحالي والذي يتعامل مع لقاحات تستخدم منذ وقت طويل يؤكد على ابقاء بضائع فائضة، وهو يستفيد من كل البواعث المتاحة لدى منتجي اللقاحات كي يبيعوا منتجاتهم بأسعار تساوي أو تقترب، أو حتى أدنى، من التكلفة الحدية للإنتاج. وتتخذ النجاحات المرموقة التي يحققها البرنامج صفة الاستمرار، وما إن يتم وضع أنظمة الإيصال في مكانها تماماً، حتى يسمح بعقدور البرنامج الموسع للتمنيع أن يحقق الهدف في إيصال اللقاحات السنة الحالية إلى كل أطفال العالم. وعند هذه النقطة سيقدّم البرنامج الموسع للتمنيع ما يزيد عن بلون جرعة سنوية.

من المؤكد أن البرنامج الحالي يواجه تحديات. إن المسؤولين عن البرنامج الموسع للتمنيع وعن الحملة من أجل حماية الطفل يركزون على حل مشكلات تتعلق بكيفية القيام بحملات في مناطق لم يعط الأطفال فيها بعد اللقاحات السنة الكائنة وعن كيفية ضمان استمرار وصول هذه اللقاحات إلى كل جيل جديد من الأطفال، وذلك بعد أن تكون خدمات البرنامج قد شملت في النهاية كل أنحاء العالم. كما أن البرنامج الموسع للتمنيع يواجه باستمرار خطر ارتفاع الأسعار. وعلاوة على ذلك، فإن اللقاحات المتوفرة حالياً تعاني بعض العيوب. فمثلاً، تفقد هذه اللقاحات فاعليتها عندما تتعرض للحرارة، وهو خطر يبرز في البلدان النامية حيث يكون التبريد في معظم الأحيان غير كاف. فالبرنامج الموسع للتمنيع في حاجة ملحة للقاحات تتحمل تأثير الحرارة ولاخرى قادرة أن تزيد من فاعلية ومن كفاءة البرنامج بطرائق أخرى، فلقاح مضاد للحصبة مثلاً يمكن أن يعطى عند الولادة أو بعدها بقليل بحيث يعفح الوليد حماية

ملاحظة

وعلى الرغم من أنها حاسمة، فالحملة التالية عن كيفية توفير لقاحات أخرى لم تول إلا اهتماماً ضئيلاً حتى عهد قريب. ويقينا، فإنها هي التي تحتاج بشدة إلى التقدير والمدح والاستراتيجيات الجديدة. وإن تؤدي اللقاحات الجديدة إلى إنقاذ عدد لا يحصى من الأرواح فحسب، بل إن إتاحتها ستساعد على ديمومة اندفاع القادة السياسيين في الأقطار النامية لترسيخ النظم اللازمة التي تتلاقى مع أهداف البرنامج



حرماني شرق باكستان مع معرفة كمصدر للكوليرا (الحمية). وفي كل مكان في العالم، التامي نجد أن الماء الذي يؤخذ من البحيرات والعيان والأنهار، والمثوث بالفضلات البشرية والمصائر الأخرى لعدد من التعضيات الضامجة، والتي يستعمل في الشرب والطهو، يساهم على انتشار الكوليرا وعدد آخر من الأمراض كالتهاب الكبد والفيروسات المستديرة وداء الشيكلات، والحمى التيفية. كما أن المستوى الصحي الكندي والسكن المكتظ ووفرة الحشرات الحاملة للأمراض (نقلات الجرثوم). كلها عوامل تعزز انتشار المرض. ويمكن توقي عدد كبير من الأمراض بتنفيذ برامج تناول نقيحة مصادر مياه الشرب وتحسين المستوى الصحي وشروط السكن والتخلص من جماعات الحشرات الناقلة للعوامل المرضية. أضف إلى ذلك أن برامج المعالجة غالباً ما تحول دون حدوث الإصابة أو الموت. بيد أن كاتبي المقللة يشيرون إلى أن توسيع البرامج الخاصة بالتمنيع هي، على المدى القصير، أقل البرامج تكلفة وأبصر الطرائق لنقل حياة الناس.

موجودة في كندا وأوروبا واليابان؛ ذلك أن السوق المحلية في الولايات المتحدة على درجة من الضخامة بحيث لا تجد شركات الأدوية الأميركية حاجة ملحة للتنافس في جهات أخرى، أو أن تعطل مع البرنامج الموسع للتمنيع. وأيضا فإنها أثبتت عن عزيمتها لبيع اللقاحات لهذا البرنامج بكلفة منخفضة لأن الكونغرس قد انتقدها لعدم عرضها أسعاراً منخفضة مماثلة في موطنها.

ولقد مكن نظام أستدراج العروض عام 1988 البرنامج ومنظمة الصحة للأقطار الأميركية من شراء اللقاحات بسعر متوسط منخفض فوق العادة بلغ خصمة سننات للجرعة الواحدة، وهذا يعني تكلفة خمسين سنتاً لشراء كل الجرعة لكل الأمراض السنة. وينفق البرنامج الموسع للتمنيع أكثر من عشرة أضعاف هذا المبلغ لكل طفل على البنود الأخرى غير اللقاح، كاجور نقل ورواتب الأطر (الكوادر) التي ستغطي اللقاح وتكلفة التجهيزات الخاصة بالتبريد والتي تحمي اللقاح في (سلسلة باردة) cold chain.

وعلى الرغم من أن البرنامج الموسع للتمنيع قد استطاع أن يفيد من الدوافع المتفاوتة لعدد كبير من منتجي اللقاحات كي يحصلوا على أسعار منخفضة قبل عام 1983، فقد نجح البرنامج في ذلك العام في تخفيض نفقته كذلك باستهلاله نظاماً لاستدراج العروض. وعلى الأخص (اليونيسيف) UNICEF، الذي يتتبع معظم اللقاحات التي تستخدم في أفريقيا وبعض اللقاحات التي تستخدم في آسيا، و(منظمة الصحة للأقطار الأميركية) Pan American Health Organization (PAHO)

التي تطوري اللقاحات أميركا اللاتينية لتلقيان عروضاً لكل لقاح على حدة من عديد من المنتجين، وتبرهان عقد شواء مع المنتج المؤهل الأخص سهرًا. (وتركز منظمة الصحة العالمية اهتماماتها حالياً على النواحي العلمية والإدارية وعلى تدريب الأطر أكثر من تركيزها على عمليات الحصول على اللقاحات). منذ استهلال برنامج أستدراج العروض كانت كل الشركات المهمة والمتنمة للقاحات

Figure V.2 : Séparation des blocs majeurs du document de la figure V.1

إن نظام المشتريات الحالي والذي يتعامل مع لقاحات تستخدم منذ وقت طويل يؤكد على ابتداء بوضوح فائضة. وهو يستفيد من كل البراعة المتاحة لدى منتجي اللقاحات كي يبيعوا منتجاتهم بأسعار تساوي أو تقارب، أو حتى أدنى، من التكلفة الحقيقية للإنتاج. وتتخذ النجاحات المرموقة التي يحققها البرنامج صفة الاستمرار، وما إن يتم وضع أنظمة الإيصال في مكانها تماماً، حتى يصبح بمقدور البرنامج الموسع للتنمية أن يحقق الهدف في إيصال اللقاحات السنوية الحالية إلى كل أطفال العالم. وعند هذه النقطة سيقيم البرنامج الموسع للتنمية ما يزيد عن بلون جرعة سنوياً



بمصر ما في شرق باكستان ثم معرفة مصدر الكوليرا (التهمة) وإن كل مكان في العالم الثاني نجد أن الماء الذي يتخذ من الصنوبريات والتفرايز والأنهار والظنون بالفضلات البشرية والصابون الأخرى لعقد من التخصيب الخاصة، والذي يستعمل في الشرب والظهو، يساعد على انتشار الكوليرا وعند آخر من الأمراض فتفوق الكتل الفيروسات المنتشرة بوجاء اللقاحات، والحصى التيفية، كما أن المستوي الصحي المنخفض والسكن المكتظ ووفرة الحشرات الجائعة للأمر آخر (بالذات الجرثومي)، كلها عوامل تعزز انتشار المرض ويمكن توفيق عدد كبير من الأمراض بشكل برنامج يتناول مشكلة مصادر مياه الشرب وتحسين المستوى الصحي وشروط السكن والتنظيف من جماعات الحشرات الخائفة للعوامل المرضية، أضف إلى ذلك أن برنامج العلاج عالمياً ما تحول مؤن حدوث الإعاقة أو الموت. بيد أن كشمي المقالة يشيران إلى أن توسيع البرامج الخاصة بتقديم عي على المدى القصير، أقل البرامج تكلفة وأيسر الطرق لتقليل حياة الناس

من المؤكد أن البرنامج الحالي يواجه تحديات. إن المسؤولين عن البرنامج الموسع للتنمية وعن الحملة من أجل حماية الطفل يركزون على حل مشكلات تتعلق بكيفية القيام بحملات في مناطق لم يعط الأطفال فيها بعد اللقاحات السنوية الكافية وعن كيفية ضمان استمرار وصول هذه اللقاحات إلى كل جيل جديد من الأطفال، وذلك بعد أن تكون خدمات البرنامج قد شملت في النهاية كل أنحاء العالم، كما أن البرنامج الموسع للتنمية يواجه باستمرار خطر ارتفاع الأسعار، وعلاوة على ذلك، فإن اللقاحات المتوفرة حالياً تعاني بعض العيوب. فمثلاً تفقد هذه اللقاحات فاعليتها عندما تتعرض للحرارة، وهو خطر يبرز في البلدان النامية حيث يكون التبريد في معظم الأحيان غير كاف. فالبرنامج الموسع للتنمية في حاجة ملحة للقاحات تتحمل تأثير الحرارة والآخرى بآثاره التي تزيد من فاعلية ومن كفاءة البرنامج بطرائق أخرى، فلقاح مضاد للحصبة مثلاً يمكن أن يعطى عند الولادة أو بعدها بقليل بحيث يفتح الوليد حماساً

بوجوده في كندا وأوروبا واليابان! ذلك أن السوق المحلية في الولايات المتحدة على درج من الضخامة بحيث لا تجد شركات الأدوية الأمريكية حاجة ملحة للمنافسة في جهات أخرى أو أن تعمل مع البرنامج الموسع للتنمية. وأيضاً فإنها أشتت عن عزمها لبيع اللقاحات لهذا البرنامج بكلفة منخفضة لأن الكونغرس قد اتخذها لعدم عرضها أسعاراً منخفضة مماثلة في موطنها

وعلى الرغم من أن البرنامج الموسع للتنمية قد استنظام أن يفيد من الدوافع المتفاوتة لعده كبير من منتجي اللقاحات كي يحصلوا على أسعار منخفضة قبل عام 1983، فقد نجم البرنامج في ذلك العام في تخفيض نفقته كذلك باستهلاكه نكلاً لاستدراج العروض. وعلى الأحمر (اليونيسيف) UNICEF، الذي يتناج معظم اللقاحات التي تستخدم في أفريقيا وبعض اللقاحات التي تستخدم في آسيا (منظمة الصحة للإقطار الأمريكية) Pan American Health Organization (PAHO) التي تشتري اللقاحات لأمريكا اللاتينية تتلقان عروضاً لكل لقاح على حدة من عديد من المنتجين، ويبرمان عقد شراء مع المنتج المؤهل الأرخص سعراً. (وتركز منظمة الصحة العالمية اهتمامها حالياً على النواحي القلعية والإدارية وعلى تدريب الأطر أكثر من تركيزها على عمليات الحصول على اللقاحات)

وعلى الرغم من أنها حاسمة، فالعضلة التالية عن كيفية توفير لقاحات أخرى لم تول إلا اهتماماً ضئيلاً حتى عهد قريب. وبقيتها فإنها هي التي تحتاج بشدة إلى التفكير المبدع والامتنان الجديرة. وإن تؤدي اللقاحات الجديدة إلى إنقاذ عدد لا يحصى من الأرواح فمسيب، بل إن إتاحتها ستساعد على ديمومة اندفاع القادة السياسيين في الأقطار النامية لتوسيع النظم اللازمة التي تتلاقى مع أهداف البرنامج

ولقد مكن نظام استدراج العروض عام 1983 اليونيسيف ومنظمة الصحة للإقطار الأمريكية من شراء اللقاحات بسعر متوسطاً منخفض فوق العادة بلغ خمسة سنتات للجرعة الواحدة، وهذا يعني تكلفة خمسين سنتاً لشراء كل الجرعة لكل الأمراض الستة. ويتفق البرنامج الموسع للتنمية أكثر من عشرة أضعاف هذا المنفق لكل طفل على البنود الأخرى غير اللقاح. كأجود نقل ورواتب الأطر (الكوادر) التي ستعطي اللقاح وبكافة التجهيزات الخاصة بالتبريد والتي تحمي اللقاح في (سلسلة باردة) cold chain

17

Figure V.3 : Séparation des lignes à l'intérieur des blocs texte du document de la figure V.1

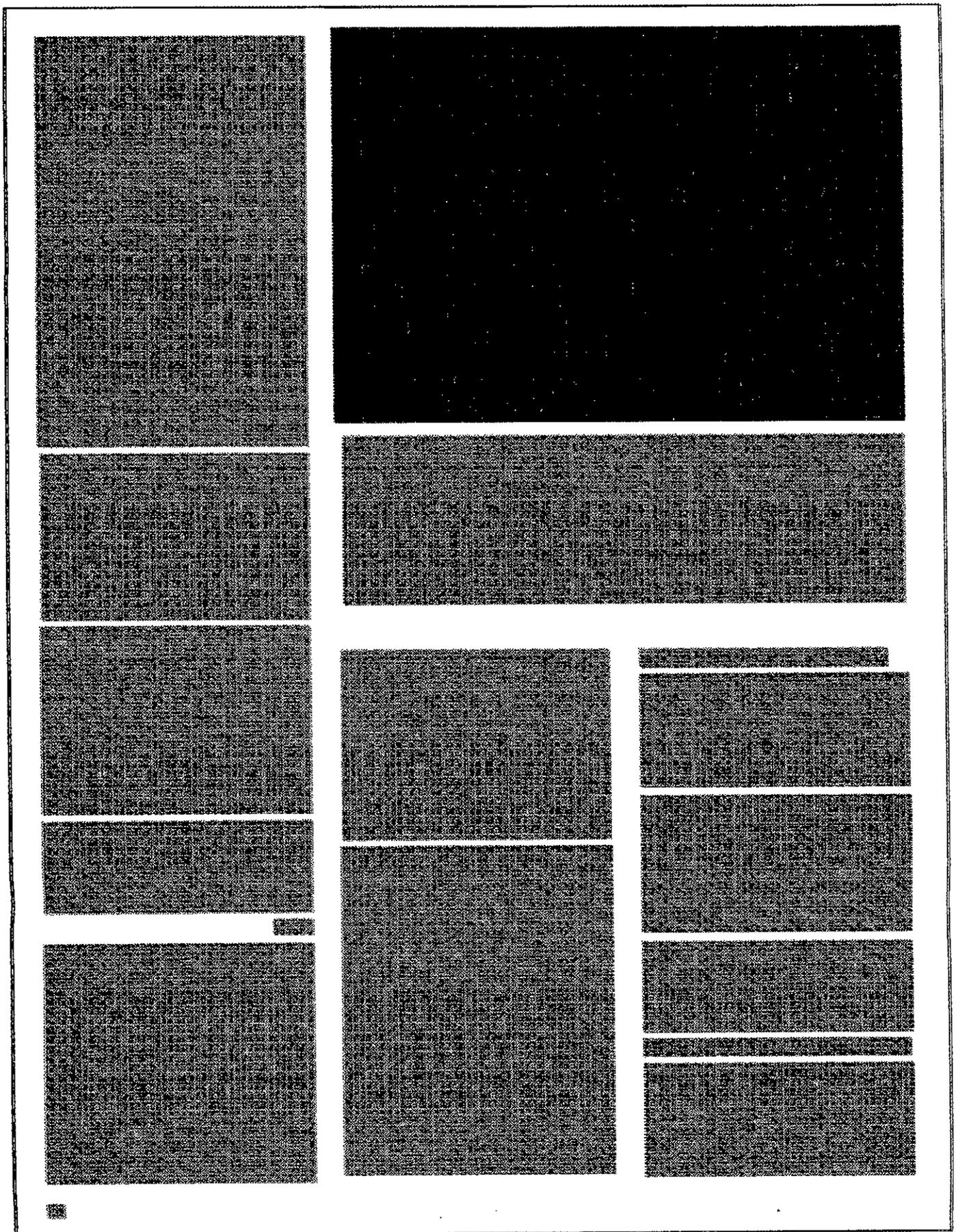


Figure V.4 : Reconnaissance des blocs du document de la figure V.1
(Les blocs en couleur claire sont les blocs texte et ceux en couleur sombre sont des photographies).

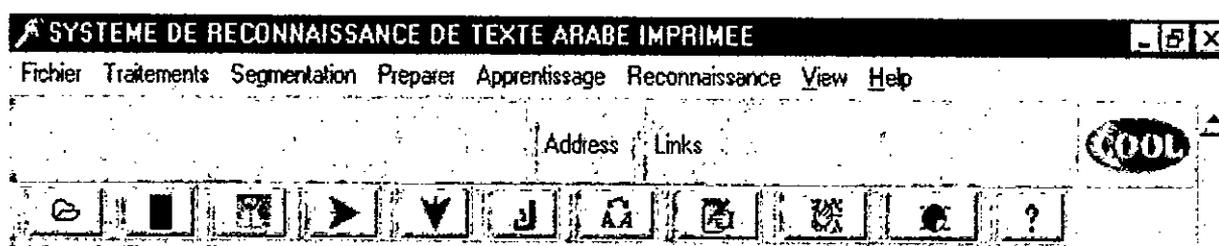
VI-2- Le Logiciel :

Notre logiciel exploite dans sa réalisation tous les moyens offerts par Windows à savoir :

- Un système de menus déroulants donnant un accès simple à toutes les fonctions du système,
- L'utilisation indifférente de la souris ou du clavier pour les différentes manipulations,
- Une barre de boutons à manipulation pratique permettant un accès rapide aux principales fonctions du système.

VI-3- Utilisation du menu :

Le dernier logiciel implémenté possède le menu suivant :



➤ Le menu Fichier :



- **Ouvrir** : Cette fonction ouvre une boîte de dialogue qui permet de sélectionner un fichier image (image texte).
- **Enregistrer** : Enregistre le fichier après modification.
- **Enregistrer Sous** : Enregistre le fichier sous un autre nom.
- **Quitter** : permet de quitter l'application OCR.

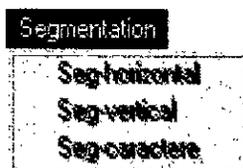
➤ Le menu Traitement :



Ce menu contient la fonction de traitement du fichier image, on y trouve le filtre retenu :

- **Filtre médian** : permet de filtrer l'image en enlevant les pixels isolés (bruit impulsionnel). Ce filtrage est optionnel.

➤ **Le menu Segmentation :**



Cette fonction permet de segmenter une image de texte en passant par trois étapes successives :

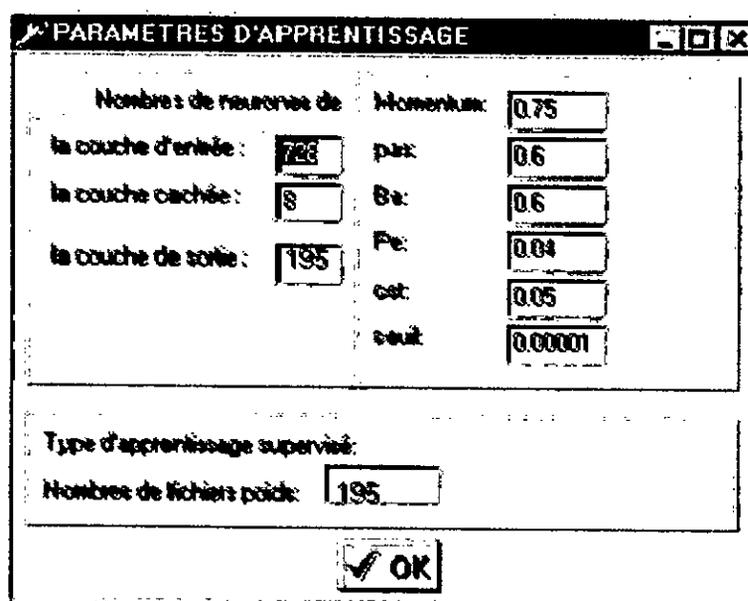
- **Seg-horizontal** :
Consiste à segmenter le texte en lignes (ligne1, ligne2, ..., ligneN)
- **Seg-vertical** :
Consiste à segmenter chaque ligne de texte en parties connexes (connexe1,...connexeM)
- **Seg-caractère** :
Consiste à segmenter chaque partie connexe en caractères isolés (car1, car2,..., carP) .

➤ **Le menu Apprentissage :**



Cette fonction permet de lancer l'opération d'apprentissage en procédant aux deux étapes suivantes :

- **Paramètres** : permet de fixer les paramètres via la boîte de dialogue ci-après :
(Ces paramètres sont statiques, ils ont été fixés après plusieurs tests)
- **Appt** : permet de lancer l'apprentissage via la boîte de dialogue suivante.



➤ Le menu préparer :



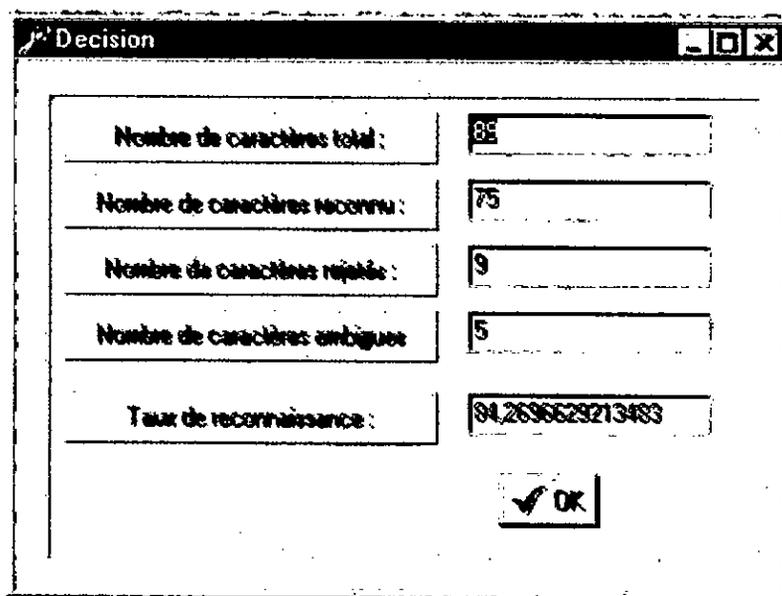
Cette fonction permet d'extraire les caractéristiques des caractères segmentés et de les envoyer à l'entrée du réseau de neurones afin d'être traitées puis de recevoir les codes des caractères reconnus.

➤ Le menu Reconnaissance :

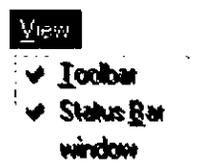
Cette fonction lance l'opération de reconnaissance :



- **Reco** : consiste à afficher les caractères reconnus après avoir reçu les codes de ces derniers dans le menu extraction et ceci à partir du fichier de caractères reçu par la fonction extraction du menu précédent.
- **Décision** : permet de visualiser les résultats de la reconnaissance comme le montre la boîte de dialogue suivante.



- **Le menu View** : permet de contrôler les barres de défilement de l'écran



Toolbar : contrôle la 1^{ère} barre de boutons.

Status Bar : contrôle la 2^{ème} barre de boutons.

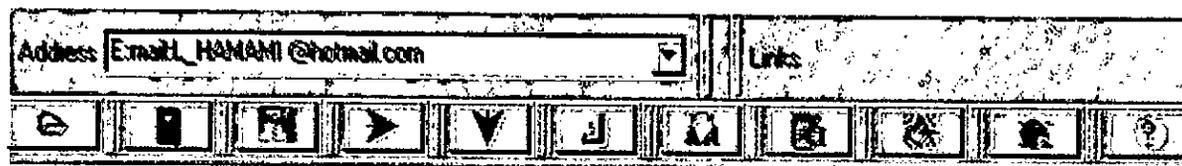
Window : contrôle la barre de bas de l'écran.

- **Le menu Help** :



- **Help** : affiche de l'aide concernant l'application.
- **About** : ouvre une Fenêtre donnant des informations sur l'application.

➤ La barre de boutons :



Description des boutons :

- | | |
|--|---|
|  Ouvrir une image existante |  extraction des caractères |
|  fermer l'application OCR |  Apprentissage |
|  Filtrer l'image bruitée |  Reconnaissance |
|  segmentation horizontale |  Décision |
|  segmentation verticale | |
|  Segmentation en caractères |  Aide |

V. Déroulement des différentes phases du logiciel

Nous allons dans ce qui suit présenter un exemple sur lequel nous ferons dérouler les différentes phases aboutissant à la reconnaissance d'un paragraphe de texte. Les différents résultats y seront donnés.

V.1 Résultats de l'acquisition et du prétraitement :

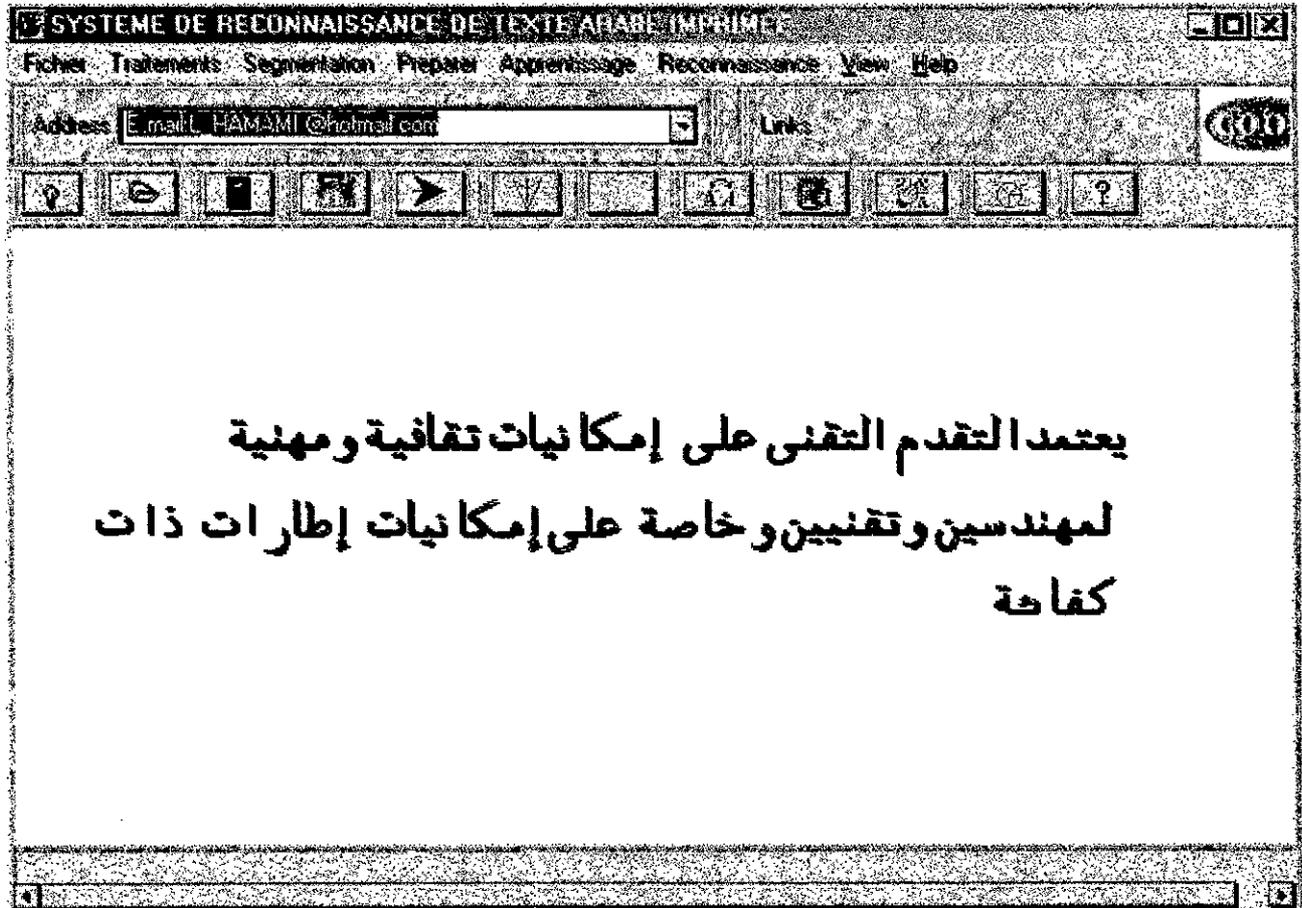


Figure V.6 : Ouverture d'un fichier

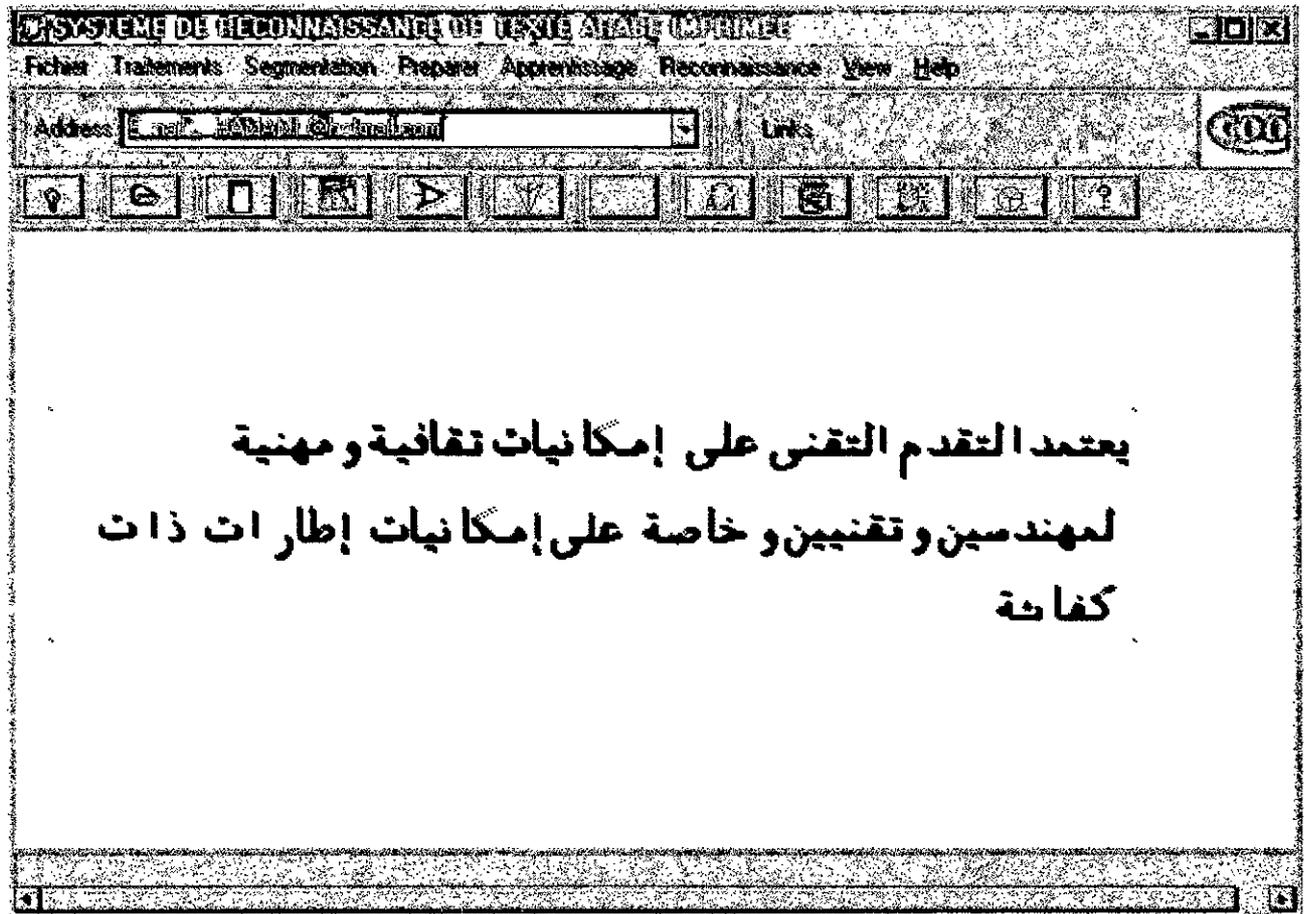


Figure V.7 : Résultat du filtrage médian

Nous remarquons que le filtrage agit sur la forme des caractères, les petites boucles se referment et deviennent plus difficiles à être détectées. Les traits obliques (cas du kaf) s'amincissent et ceci pourrait aussi être gênant pour la paramétrisation. C'est pour cette raison que nous avons préféré garder le filtrage optionnel, il dépendra de la nature du document à traiter.

V.1 Résultats de la Segmentation d'un texte:

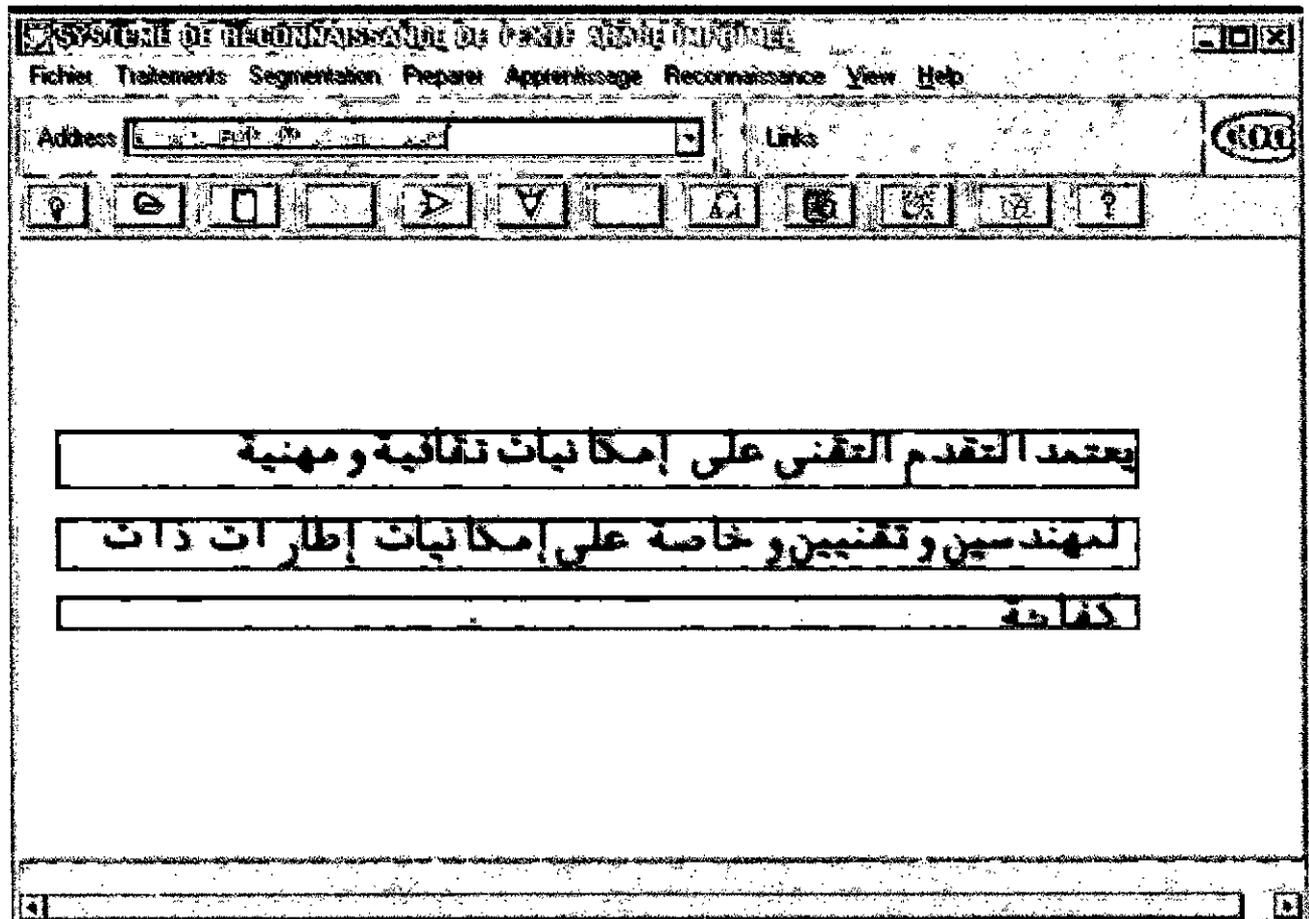


Figure V.8 : Résultat de la segmentation horizontale

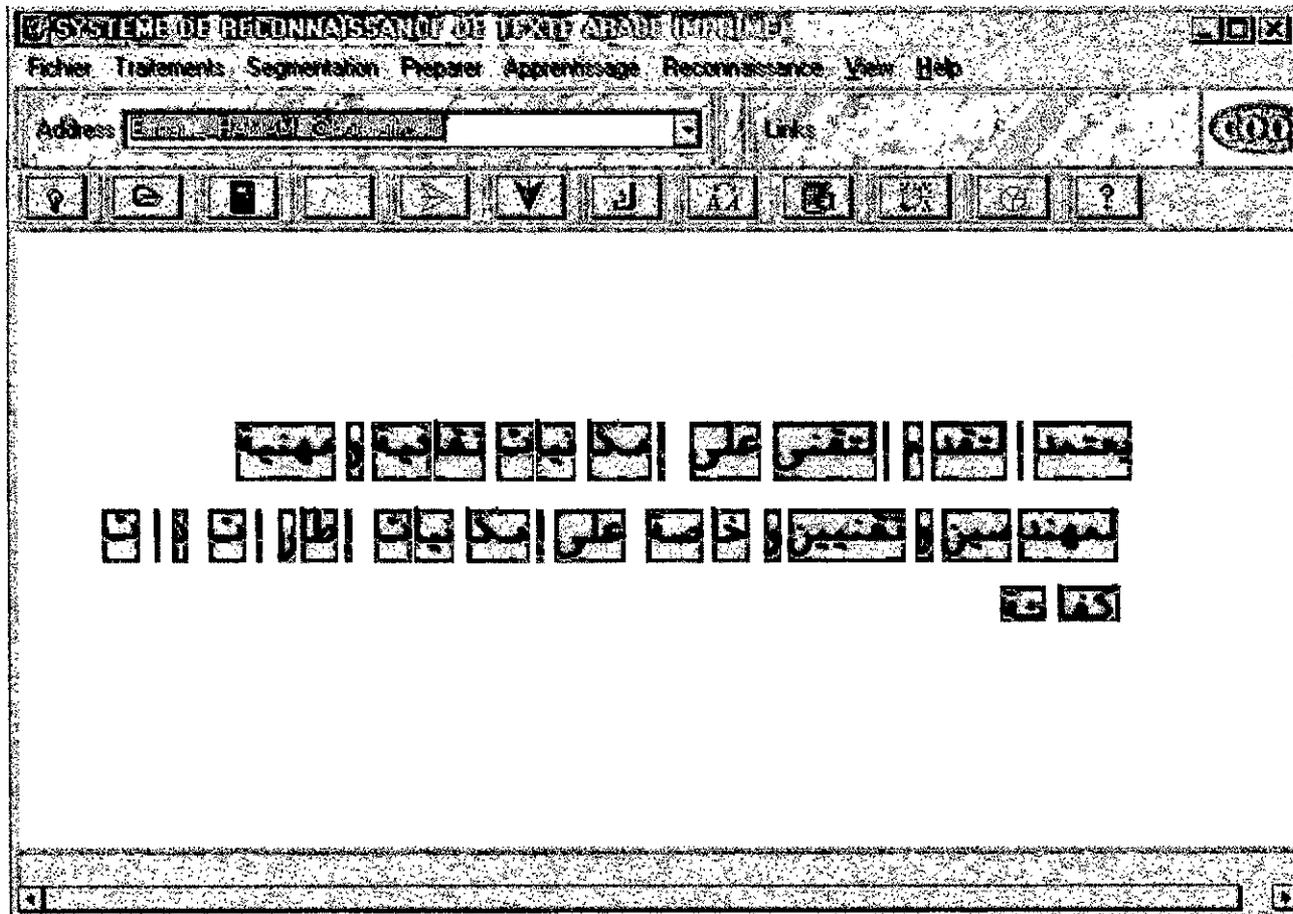


Figure V.9 : Résultat de la segmentation verticale

Les segmentations horizontale et verticale sont effectuées correctement sans aucune erreur.

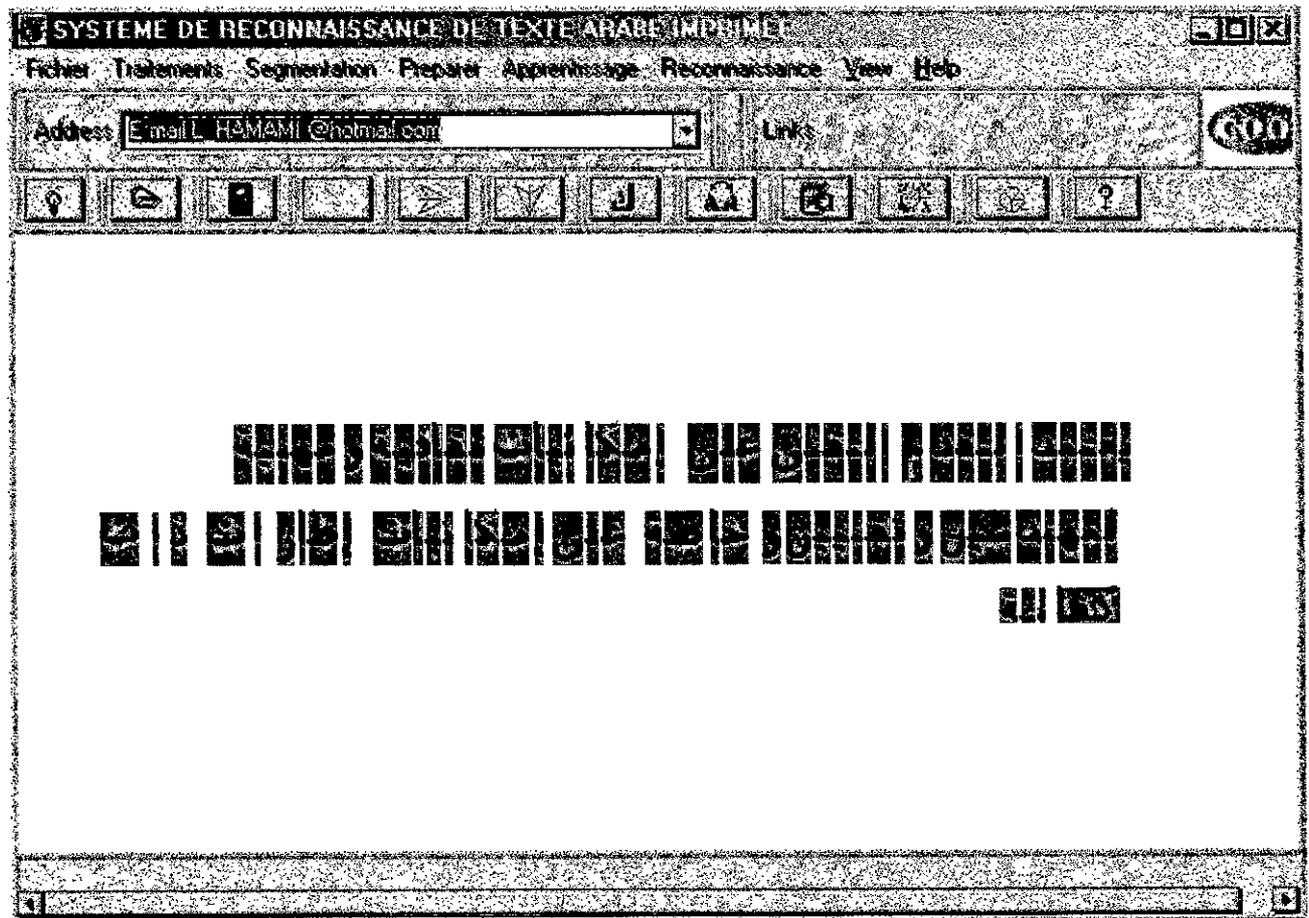


Figure V.10 : Résultat de la segmentation en caractères

Nous remarquons que cinq caractères n'ont pas été segmentés et qu'un autre a été sur segmenté. Ceci ne devrait pas arriver. Le résultat de cette segmentation va influencer sur la reconnaissance.

V.1 Résultats de la Reconnaissance:

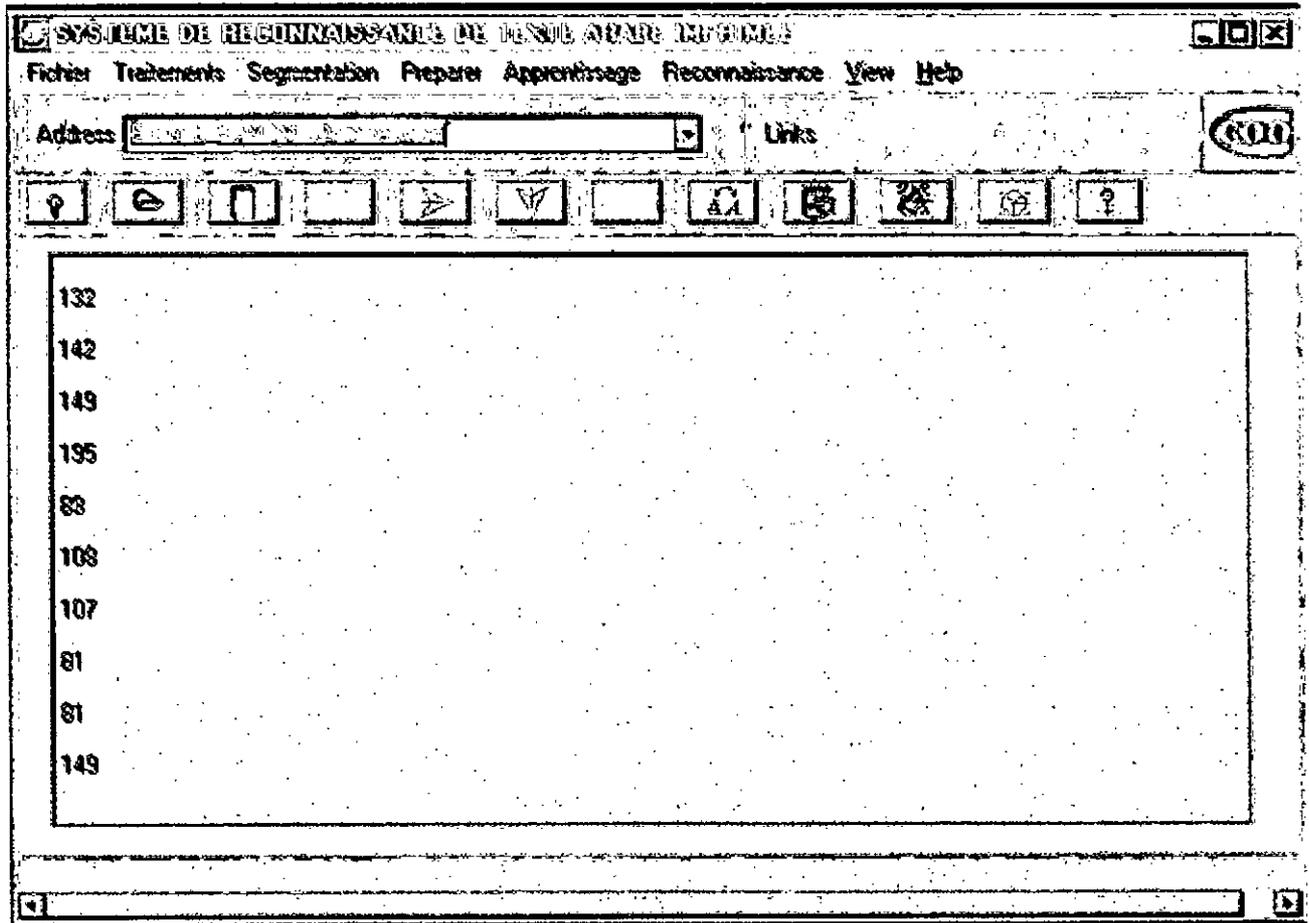


Figure V.11 : Une partie des codes des caractères reconnus

Ces codes sont générés par la fonction extraction du menu préparer.

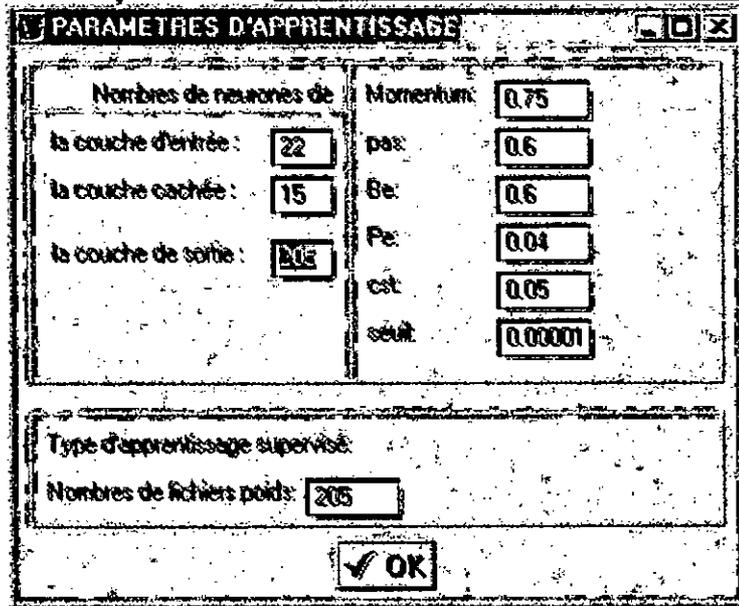


Figure V.12 : Boîte de dialogue permettant de fixer les paramètres d'apprentissage

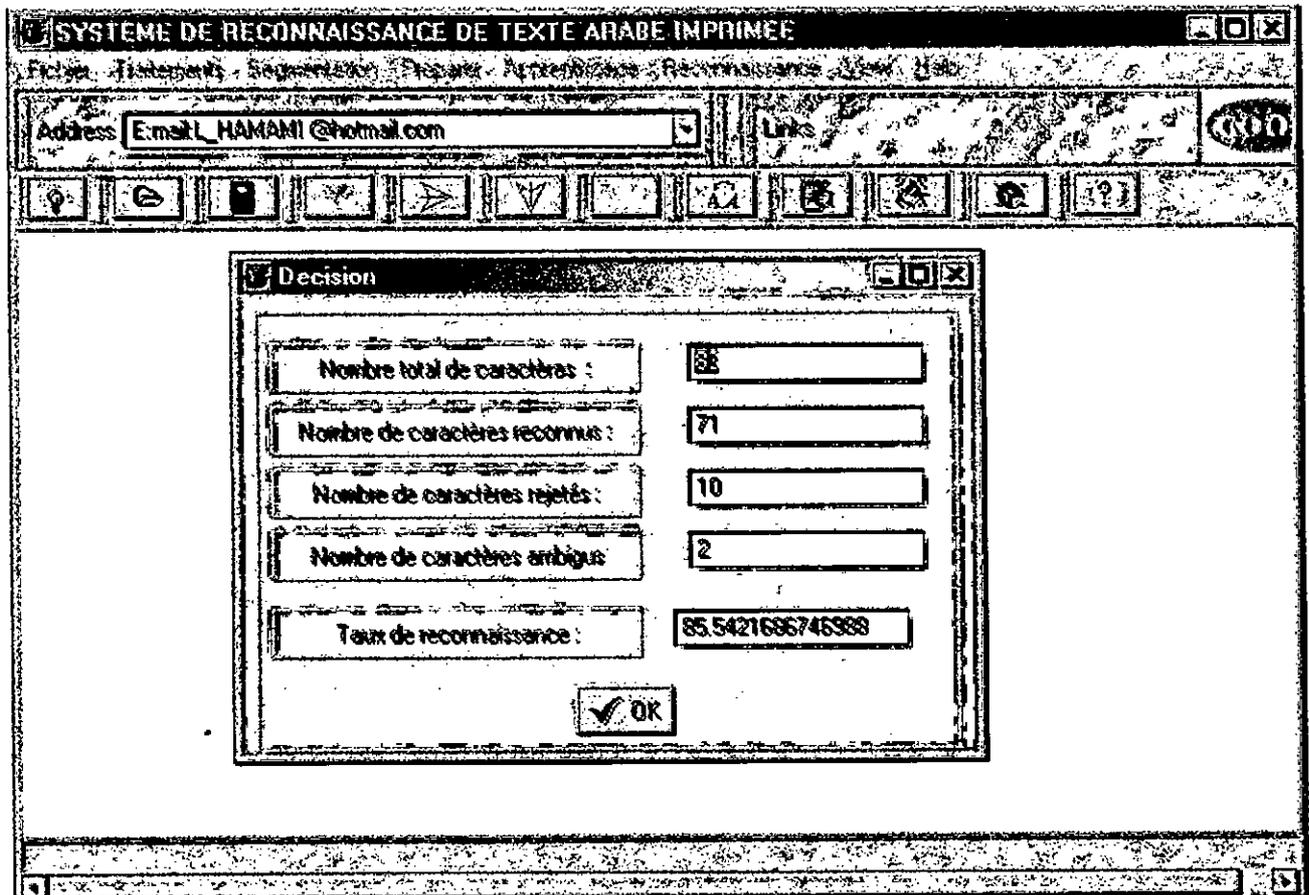


Figure V.13 : Résultats de la reconnaissance

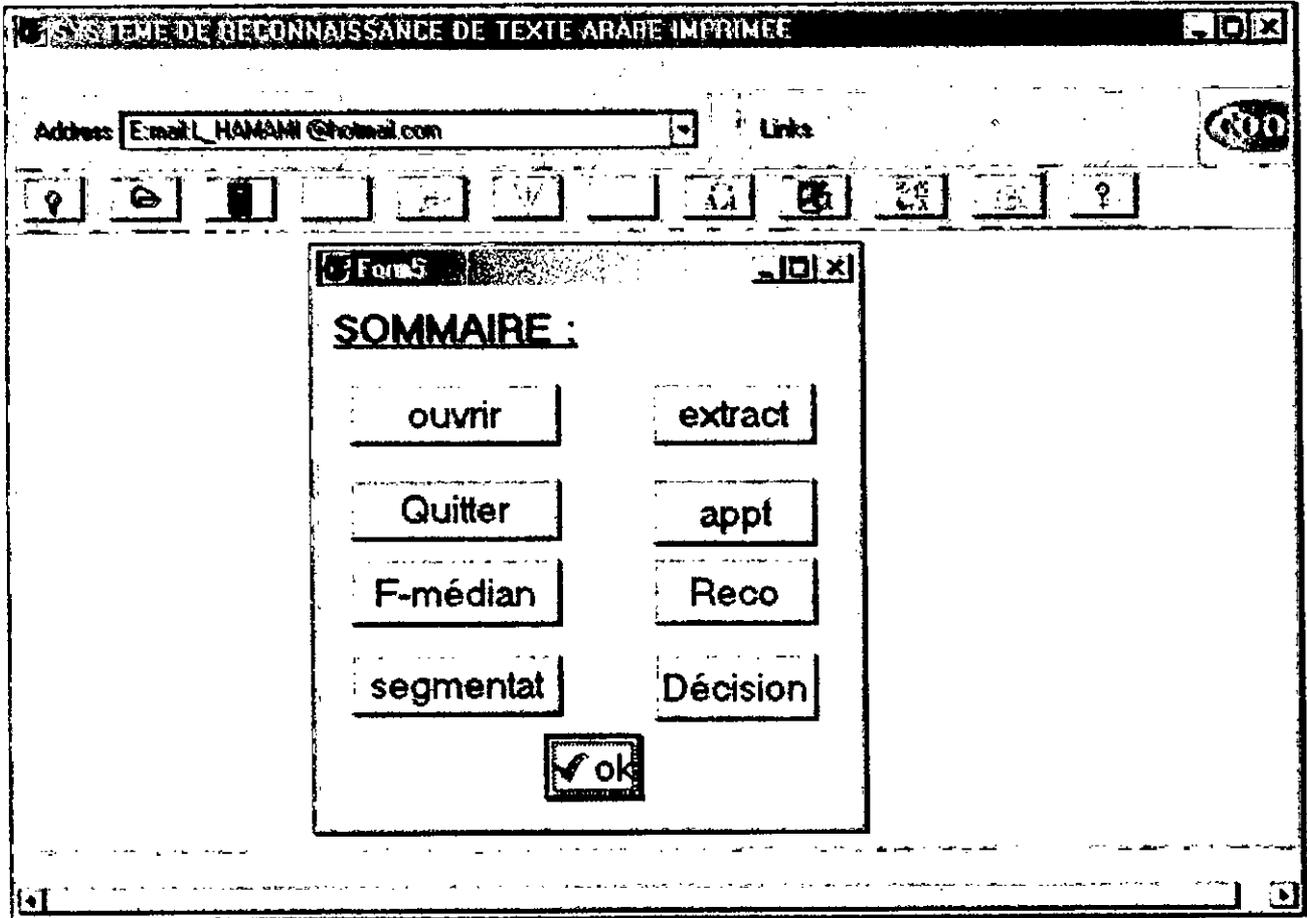


Figure V.14 : Menu d'aide

Ce menu explique le rôle et le fonctionnement de chaque bouton de la barre des outils.

V. Conclusion

Les résultats de la reconnaissance ne dépendent pas uniquement de l'approche utilisée. Ils sont étroitement liés aux résultats obtenus lors de la segmentation et même du filtrage (comme il a été remarqué plus haut).

Le taux de reconnaissance peut être amélioré par un meilleur choix des paramètres d'apprentissage qui restent jusque là purement expérimentaux. Nous pouvons penser aussi à utiliser quatre réseaux de neurones, chacun étant dédié à une classe de caractères regroupant ceux ayant la même position dans le mot (position début, milieu, finale et isolée du caractère).

La base d'apprentissage conçue comporte 205 caractères prototypes créés sous forme de vecteurs à cinq caractéristiques.

Conclusion

CONCLUSION

La lecture optique constitue un sujet important pour la recherche, à cause de la diversité des problèmes qu'elle doit résoudre, en fonction du contenu, de la forme et de la qualité des documents. Elle ne se limite, en général, pas seulement à la reconnaissance du texte ; selon le contenu du document elle doit encore être en mesure de traiter la présence d'images, de graphiques ou de plans [1, 2], voire d'effectuer une analyse partielle du contenu. La reconnaissance de table, contenant par exemple des nombres, constitue une autre généralisation de la lecture de textes au sens usuel.

La reconnaissance des caractères est sans doute le problème clé de la lecture optique. La diversité des alphabets (latin, arabe, grec, chinois) et des types d'écritures (écriture normalisée dactylographiée, imprimée ou manuscrite) nécessite des méthodes souvent fort différentes.

Notre contribution à ce vaste domaine d'actualité, consiste en la réalisation d'un système de reconnaissance de l'écriture arabe imprimée possédant un ordre de difficulté très proche du manuscrit latin. Par ailleurs, nous nous sommes efforcés de concevoir un logiciel que l'on pourrait facilement rendre bilingue en ayant pris la précaution de considérer des caractéristiques communes aux deux écritures Arabe et Latine. L'approche que nous avons retenue pourra être utilisée pour la lecture de textes arabes manuscrits.

Nous procédons, après reconnaissance d'un bloc de texte, aux trois étapes conventionnelles d'un système de reconnaissance de formes ; à savoir le prétraitement, la segmentation puis la reconnaissance des caractères.

L'image du document est saisie par un capteur optique (scanner) qui produit une image numérisée de la page. Selon les besoins, nous faisons subir à cette image brute un pré-traitement dans le but d'améliorer sa qualité. Il s'agit notamment de corriger l'inclinaison (due au mauvais positionnement du document dans le scanner), d'éliminer les bruits et d'effectuer un seuillage pour obtenir une image à deux niveaux, jugée suffisamment riche en informations pour la lecture automatique. Quant à la résolution, elle dépend du type et de la taille des caractères et varie entre deux cents et six cents points par pouce ; son choix résulte souvent d'un compromis entre la fiabilité de la reconnaissance et la rapidité de traitement.

Une étude et implémentation de plusieurs filtres susceptibles d'être utilisables pour des images de textes ont été faites et le choix a porté sur le filtre médian qui a donné les meilleurs résultats.

Nous effectuons ensuite la segmentation du document qui constitue la première étape importante du processus de reconnaissance. Son rôle est de subdiviser chaque page en blocs.

Le processus de segmentation, d'un bloc de texte en caractères, que nous avons implémenté procède en trois étapes qui consistent à segmenter le texte en lignes puis les lignes en sous-mots ou parties connexes (car certains caractères arabes ne se rattachent pas à leurs prédécesseurs ou à leurs successeurs et de ce fait divisent le mot en deux ou plusieurs sous-mots) et enfin les parties connexes en caractères. Cette dernière étape est la plus complexe et doit tenir compte des caractéristiques morphologiques de l'écriture arabe. Nous avons considérablement amélioré le taux de segmentation (en résolvant les cas des caractères sous segmentés ou sur segmentés) [10,11], néanmoins certains problèmes apparaissent aléatoirement et qu'il nous faudra reconsidérer.

Chaque forme isolée par la segmentation constitue un échantillon dont l'identification revient à un problème de classement. C'est dans ce domaine que les méthodes décrites dans la littérature sont les plus variées. Schématiquement elles procèdent presque toujours en deux phases : l'extraction de traits caractéristiques de l'échantillon, puis leur comparaison avec les prototypes de références.

La constitution des prototypes nécessaires à la phase d'identification est un problème en soi, connu sous le nom d'apprentissage. Là aussi les méthodes varient en fonction des applications : parfois

l'apprentissage est réalisé dans une phase initiale indépendante, mais souvent il est intégré à la phase d'identification.

Nous avons implémenté la phase de reconnaissance en utilisant trois approches différentes puis en hybridant la plus performante avec un réseau de neurones multicouches à propagation du gradient.

Un tableau récapitulatif des caractéristiques de ces quatre approches est donné au chapitre V.

La méthode statistique donne des taux de reconnaissance très élevés mais possède l'inconvénient d'être monofonte; elle pourra être utilisée pour des documents standards utilisant la même fonte et par conséquent l'apprentissage devra se faire sur cette seule fonte.

La méthode structurelle utilisant les arbres comme descripteurs donne de bons résultats mais est multifonte et par conséquent demande un temps d'apprentissage ainsi qu'un espace mémoire importants.

La méthode morphologique est omnifonte et considère des paramètres communs aux caractères arabes et latins [120], ceci lui donne la possibilité d'être bilingue. D'autre part, les caractéristiques morphologiques permettent de considérer des caractères manuscrits.

Le réseau de neurones doit être reconsidéré en améliorant les paramètres d'apprentissage. En perspective, nous proposons d'utiliser quatre réseaux de neurones, chacun traitant une position différente du caractère.

Les performances d'un système de lecture optique sont généralement évaluées au moyen des taux de reconnaissance, de rejet et d'erreur sur l'ensemble des caractères analysés. Cette mesure est à relativiser puisqu'elle dépend davantage de la qualité des documents que du lecteur lui même.

En ce qui concerne la langue latine, les meilleurs taux de reconnaissance, rapportés atteignent 99,8% dans le domaine de la lecture mono-fonte, dactylographiée ou standardisée. Dans le cas de textes imprimés, un taux de reconnaissance de 99,5% peut être considéré comme suffisant. Quant à la reconnaissance de caractères manuscrits indépendants le taux de reconnaissance se situe rarement au delà des 95% et se dégrade considérablement en écriture cursive [140].

Quant à l'écriture arabe qui est cursive, même dans le cas de textes imprimés ou dactylographiés, le taux de reconnaissance des caractères indépendants (c'est à dire isolés) demeure insuffisant. De nombreuses investigations seront encore nécessaires pour venir à bout de ces problèmes. Les taux obtenus, pour un même texte par les différentes méthodes implémentées, sont donnés dans le tableau V.1.

Les efforts entrepris en lecture optique promettent de rationaliser la saisie de textes bruts, mais leur intégration à d'autres logiciels et leur fiabilité intrinsèque posent souvent encore quelques problèmes.

Ces problèmes résultent du fait que la lecture optique repose sur des modèles de documents trop pauvres qui ne tiennent pas compte des informations typographiques disponibles à la source, pourtant essentielles pour une interprétation intelligente des documents saisis.

La solution que nous proposons en perspective, valable pour des documents structurés de bonne qualité typographique, s'appuie sur une modélisation des documents empruntée au formatage et à la composition. Dans ce modèle nous distinguons fondamentalement la structure logique de la structure physique d'un document. La structure logique est décrite, de façon générique, par un formalisme semblable à une grammaire. Cette description formelle est complétée par des règles de présentation dont le but est de définir les caractéristiques typographiques et de mise en page des divers constituants du document.

Il serait intéressant d'envisager des méthodes d'inférence grammaticale qui permettraient la constitution des descriptions de documents par apprentissage, à partir de documents caractéristiques.

Nous proposons, pour cette perspective, une méthode d'analyse basée sur les techniques de reconnaissance syntaxique.

Jusqu'à présent, le développement de la lecture optique et celui de la production de documents sont restés presque totalement indépendants. Pourtant leur lien semble évident, aussi bien du point de vue pratique des applications bureautiques que du point de vue théorique de la modélisation.

La convergence récente de l'informatique et du monde de l'édition implique que les documents sont de plus en plus voués à une double existence : l'une matérielle, sur papier par exemple, l'autre abstraite dans les bases de données des ordinateurs. Si certaines gens prédisent une disparition progressive du document sur papier, nous estimons, quant à nous, que les deux formes sont complémentaires et qu'elles vont continuer à coexister.

Un objectif est donc de concilier la lecture optique avec l'approche moderne de la présentation de documents.

Cette approche présente de nombreux avantages pour les applications. Ainsi devrait-elle permettre une meilleure intégration de la lecture optique dans les nouveaux logiciels de production de documents. Elle permettrait notamment de récupérer intelligemment d'anciens textes (imprimés) dans le but de les inclure dans de nouveaux documents (électroniques).

Bibliographie

BIBLIOGRAPHIE

- [1] H.Akroum et L. Hamami, Prétraitement et segmentation d'images de documents, Proceedings 1st National Workshop on Signal & Image Processing, **NWSIP'98**, 1^{er} Décembre 1998 Sidi Bel Abbés – Algérie, pp. 24-27.
- [2] H. Akroum, L. Hamami et D. Berkani, Prétraitement, Segmentation et Reconnaissance d'images de documents, Proceedings du Séminaire sur l'Imagerie et la Transmission **SIT'99**, CDTA Alger-Algérie, 30Nov.et 1^{er} Déc.1999, pp.30-37
- [3] L. Hamami-Mitiche et D. Berkani, Système de Reconnaissance de l'écriture Arabe Imprimée, 2^{ème} Conférence Internationale d'Electronique sur les Signaux, Systèmes et Automatique, **SSA2'99**, 10-12 Mai 1999 Blida – Algérie, Volume I pp.112-116.
- [4] L. Hamami-Mitiche, Système de Reconnaissance de l'écriture Arabe Imprimée Utilisant une Méthode Statistique, **A.J.O.T (Algerian Journal Of Technology)**, International Publications of Engineering Science, Série B, Vol.14, n°1, 1999, pp.86-93
- [5] L. Hamami-Mitiche et D.Berkani, Système de Reconnaissance de l'écriture Arabe Imprimée par utilisation de la Méthode des arbres, Proceedings du Séminaire sur l'Imagerie et la Transmission **SIT'99**, CDTA Alger-Algérie, 30Nov.et 1^{er} Déc.1999, pp.43-47
- [6] L. Hamami-Mitiche, Recognition System of writing printed Arabic multi-font and multi-size, Proceedings of the 9th International Conference on Computer Theory and Applications, **ICCTA'99 (IEEE Alexandria Chapter)**, 28-30 August 1999 Alexandria, Egypt, Vol. II pp. 274-277.
- [7] L. Hamami-Mitiche, Recognition System of Arab Text Paragraph Printed, 16th **IMACS WORLD CONGRESS 2000** on Scientific Computation, Applied Mathematics and Simulation 21-25Août 2000, Ecole Polytechnique Fédérale de Lausanne -Suisse.
- [8] L.Hamami and D.Berkani, Recognition System for Printed multi-font and multi-size Arabic characters writing, **The Arabian Journal for Science and Engineering (AJSE)**, Vol.27, n°1B, April 2002
- [9] L.Hamami and D.Berkani, Recognition System for Printed Arabic characters writing using Neural Network, **Vision Computing**, Elsevier (soumis en 2001)
- [10] L. Hamami-Mitiche, Segmentation of an Arabic text printed in characters, Proceedings of the 8th International Conference on Computer Theory and Applications, **ICCTA'98 IEEE (Alexandria Chapter)**, 15-17 September 1998 Alexandria, Egypt, pp. IV.6-IV.8.
- [11] L.Hamami-Mitiche, Segmentation d'un paragraphe de texte imprimé en caractères, **COMAEI'98**, Bedjaia, Algérie
- [12] Olivier Monga et Radu Horaud, **Vision par ordinateur**, Editions Hermès, 2^e édition, 1995.
- [13] C.Remy, **L'intelligence artificielle**, Dunod, 1994.
- [14] L. Hamami, Système de classification des chromosomes par utilisation des Réseaux de Neurones, first Polytechnic Symposium on Signal Processing, 25 Avril 2002, ENP, Algiers- Algeria.
- [15] F. Guibadj, Reconnaissance et classification des chromosomes humains, thèse de Magister soutenue à l'INI en 2002 et encadrée par L. Hamami.
- [16] L.Hamami et N.Lassouaoui, Application de la Segmentation Multifractale dans la Classification des cellules biologiques, **Fractales'2000** ; 11-12 Nov. 2000, Constantine (Algérie)
- [17] L.Hamami and N. Lassouaoui, An Approach Fractal and Analysis of Variogram for Edge Detection of biomedical images, 6th International Work Conference on Artificial and Natural Neural Networks, **IWANN2001**, Granada, Spain, June 13-15 2001, Proceedings, Part II, pp.336-344, published for Springer LNCS 2085 Bio-Inspired Applications of Connectionism

- [18] N. Lassouaoui and L.Hamami, Segmentation des Images de Cellules Biologiques par une Approche Multifractale, 9^{ème} Conférence Internationale IPMU 2002, Annecy- France, 1^{er} au 5 Juillet 2002.
- [19] Olivier Pavie, Le starter photos numériques, Scans et retouches, 1999
- [20] A.Belaid , "Reconnaissance de formes (Méthodes et Application), 1995.
- [21] J-P.Cocquerez, "Analyse d'image : filtrage et segmentation", 1995.
- [22] J.J. Toumazet, traitement de l'image sur micro-ordinateur, Edition Sybex, France, 1987.
- [23] Kaddour Chakib, "Généralités sur le traitement d'images, chapitre 1 ", 1999.
- [24] M. Kunt , Traitement d'images, Edition 1993.
- [25] Jean FRUITET, Outils et méthodes pour le traitement des images par ordinateur, 1997.
- [26] J.P Haton, Intelligence artificielle et Reconnaissance de Formes, techniques de l'ingénieur, traité informatique, H1900, 15p, 1991.
- [27] MICLET, Méthodes structurelles de reconnaissance de formes, Edition Eyrolles, 1984.
- [28] K. Hornik, M. Stinchcombe and H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feed forward networks, Neural Networks, 1989, pp. 359-368.
- [29] M. Stinchcombe and H. White, Universal approximation using feed forward networks with non sigmoid hidden layer activation function, Proceedings of IJCNN, Washington D.C Vol.1, pp. 613-617, 1989.
- [30] A.Amin, Un système pour la reconnaissance et la compréhension de l'Arabe écrit et imprimé, Thèse d'état, Université de Nancy1, pp 104-144, Déc.1985.
- [31] J.Kim and C.C. Tappert, Handwriting recognition accuracy versus tablet resolution and sampling rate, 7th Int. conf. on pattern recognition, Montreal, 1984, 917-918.
- [32] F.Nouboud and R.Plamondon, On-line recognition of hand printed characters: Survey and beta tests, Pattern Recogn. 25,9 -1990- 1031-1044.
- [33] C.Tappert, C.Y.Suen and T.Wakahara, The state of the art in on-line handwriting recognition, IEEE Trans. Pattern Anal Machine Intell. PAMI-12, 1990, pp. 787-808.
- [34] R.H. Cheng, C.W. Lee and Z. Chen, Recognition of radicals in Handwritten Chinese characters by means of problem reduction and Knowledge Guidance, International Journal of Pattern Recognition and Artificial Intelligence, 10(6), pp.657-678, 1996.
- [35] H.J. Kim and S.K. Kim, On-line recognition of cursive Korean characters using Art-Based Stroke Classification, International Journal of Pattern Recognition and Artificial Intelligence, 10(7), pp.791-812, 1996.
- [36] H. Almuallim and S. Yamaguchi, A method of recognition of Arabic cursive handwriting, IEEE Transaction on Pattern Analysis and Machine Intelligence, n°5, pp.715-722, 1987.
- [37] T.S. El-Sheikh and R.M. Guindi, Computer Recognition of Arabic Cursive Script, Pattern Recognition, Vol.21, No.4, 1988, pp 293-302.
- [38] A.Amin and J.F.Mari, Machine Recognition and correction of printed Arabic text, IEEE Trans. Man Cybern., 9(1), 1989, 1300-1306.
- [39] A. Shoukry, A sequential algorithm for the segmentation of typewritten Arabic digitized text, Arabian J. Sc. and Eng. 16(4), 1991, 543-556.
- [40] V. Margner, SARAT-A system for the recognition of Arabic printed text, 11th Int. Conf. on Pattern Recognition, 1992, 561-564.
- [41] M.B Kurdy and A.Joukhadar, Multifont Recognition System for Arabic Characters, Proceedings of the 3rd International Conference and Exhibition on Multilingual Computing, Durham U.K. Dec.1992, pp.731-739.
- [42] A.Amin and H.B.Al-Sadoun, A new segmentation technique of Arabic text, 11th Int. Conf. on Pattern Recognition, 1992, 441-445.

- [43] B. Al-Badr and R. Haralick, Segmentation free word recognition with application to Arabic, 3rd Int. Conf. On Document Analysis and Recognition, Montreal, 1995, 355-359.
- [44] A. Amin, Arabic Character Recognition, Handbook of character recognition and Document Image Analysis, edited by H. Bunke and P.S.P. Wang, 1996.
- [45] D. Motawa, A. Amin and R. Sabourin, Segmentation of Arabic Cursive Script, 4th Int. Conf. On Document Analysis and Recognition, ICDAR'97, Germany, 1997.
- [46] M. Khemakhem, Reconnaissance de caractères imprimés par comparaison dynamique, these de Doctorat, Université de Paris XI, 1987.
- [47] M. Khemakhem and M.C. fehri,, Recognition of Printed Arabic Characters by dynamic comparison, First Kuwait Computer Conference, 1987, pp.448-462.
- [48] H.Y. abdelazim and M.A. Hashish, Interactive font learning for Arabic OCR, Proc. First Kuwait Computer Conference, 1989, pp.464-482.
- [49] R. Schwartz, C. Lapre, J. Makhoul, C. Raphael and Y. Zhao, Language independent OCR using a continuous speech recognition system, In 13th ICPR, Vol. C, pp.99-103 August 1996, Vienna, Austria.
- [50] N. Benamara and A. Belaid, Printed PAW Recognition Based on Planar Hidden Markov Models, In 13th International Conference on Pattern Recognition, Vol. B, 1996, Vienna, Austria.
- [51] F. El-Khaly and M. Sid-Ahmed, Machine Recognition of optically captured machine printed Arabic text, Pattern Recognition, vol. 23, n°11, 1990, pp. 1207-1214.
- [52] S. El-Dabi, R.Ramsis and A.Kamel, Arabic Character Recognition System: A Statistical Approach for recognizing cursive typewritten text, Pattern Recognition, pp.485-495, 1990.
- [53] H. Al-Yousefi and S.S. Upda, Recognition of Arabic characters, IEEE Trans. on Pattern Recognition and Machine Intelligence,PAMI-14, 1992, pp.853-857.
- [54] S. Saadallah and S. Yacu, Design of an Arabic character reading machine, Proceeding of Computer Processing of the Arabic language, Kuwait, 1985.
- [55] F. Haj Hassan and W. Haj Ali, Printed Arabic Text Recognition, The Arabic Journal for Science and Engineering, Vol. 16, n°4, oct.1991, pp. 511-518.
- [56] H.B. Al-Sadoun and A.Amin, a new structural technique for recognizing printed Arabic text, international journal of Pattern Recognition and artificial Intelligence, Vol. 1 n°9, 1995, pp.101-125.
- [57] M.Altuwajjri and M. Bayoumi, Arabic Text Recognition Using Neural Networks, International Symposium on circuits and systems- ISCAS'94, 1994, pp.415-418.
- [58] F. Bouslama, Machine Recognition of Arabic Characters by Threshold Tuning Techniques, International Conference on Computational Intelligence and Multimedia Application, ICCIMA'97, Gold Coast, Australia, 1997.
- [59] A. Amin and W. Mansoor, recognition of Printed Arabic Text using Neural Networks, 4th International Conference on Document Analysis and recognition, ICDAR'97, 1997, Germany.
- [60] Roberto Hamm, Pour une typographie arabe : contribution technique à la démocratisation de la culture arabe, la bibliothèque arabe Sindbad, Paris, 1975.
- [61] J.C. Bliss., A Relatively High-Resolution Reading Aid for the blind, Transactions on Machine System, 10, 1-9, 1969.
- [62] R.D. Badoux, Delta, Text Reading for the Blind, Proceedings of the 5th International Workshop on Computerized Production, pp.21-25, Winterthur, Switzerland, 1985.
- [63] G.V. Kondraske and A. Shennib, Character Pattern Recognition for a Telecommunication Aid for the Deaf, IEEE Transaction of Biomedical Engineering, 33, 366-370, 1986.
- [64] J. Sternberg, Automated Signature Verification using Handwriting Pressure, 1975 WOSCON Technical Paper, Western Electronic Show and Convention 19, San Francisco California, 1975.

- [65] R. Plamondon and G. Lorette, Automatic Signature Verification and Writer Identification-The State of the art, *Pattern Recognition*, 22(2), 107-131, 1989.
- [66] F.H. Murphy and E.A. Stohr, Optimal Check Sorting Strategies, *Bull. Oper. Res. Am.* 23, B/145, 1975.
- [67] H. Timm, Registering of Health Insurance Data Using the IBM 1288 Page Reader, *IBM Nachr.*, 23, 789-792, 1973.
- [68] C.W. Swonger, An Evaluation of Character Normalization, Feature Extraction and Classification Techniques for Postal mail Reading, *Proceedings Automatic Pattern Recognition*, pp. 67-87, Washington, U-S, 1969.
- [69] J. Serra, United States Postal Services, Advanced Technology Conference, November 5-7, 1990
- [70] P.F. Polizzano, OCR and Electronic Mail, *Computer Word*, 17, 49-52, 1983.
- [71] J.W.T. Smith and Z. Merali, Optical character Recognition: The technology and its application in Information Units and Libraries, Report 33, British Library, Boston Spa Wetherby, West yorks, England, 1985.
- [72] A.Gyarfas, Experiments Concerning the Inspection and Control of Car and Truck in France, *Koezlekedes Tud. Sz.* 24, pp. 85-91, 1974.
- [73] S. Kuprianov, Electronic Handwriting Analyzer, *Tekh. Misul. (Bulgaria)* 9, pp.7-13, 1972
- [74] G.L. Skalski, OCR in the Publishing Industry, data Processing XII, *Proceedings 1967 Int. data Process. Conf. And Business Exposition*, pp.362-366, Boston, Massachusetts, U.S.A, june 20-23, 1967.
- [75] H. Genchi, data Communication Terminal Apparatus, *Optical Character and Mark Reader, Denshi Tsuchin Gakkai Zasshi* 52, pp. 418-422, 1969.
- [76] M. L. Gronmeyer, Recognition of Hand printed Characters for Automated Cartography: a Progress Report, *Proceedings Soc. Photo-Opt. Inst. Engin.* , 205, pp. 165-174, U.S.A, 1979.
- [77] C.P. Joshi, Role of Electronic in Law Enforcement, *Indian Institute of Electrical and Telecommunication Engineering (India)* 20, pp.500-503, 1974.
- [78] Y. Nakamura and al., an Optical Character Recognition System for Industrial Application: Tosey-1000, *International Workshop on Industrial application of Machine Vision and Machine Intelligence, Seiken Symp.*, pp. 364-368, Tokyo, Japan, 1987.
- [79] Y. Hirayama, A Bloc Segmentation for Document Images with Complicated Column Structures, in *Proceedings of ICDAR'93: 2nd International Conference on Document Analysis and Recognition*, Tsukuba, Japan, Oct. 28-30 1993, pp. 91-94.
- [80] P. Lefèvre, C. Felter and P. Lobbrecht, "Reconnaissance de Documents : Passage du Document Papier à l'Information Electronique," in *épure, revue trimestrielle publiée par la Direction des Etudes et de Recherches, Electricité De France (EDF)*, n° 58, Avril 1998, pp. 15-25.
- [81] A. Dengel, "ANASTASIL: A System for Low-Level and High-Level geometric Analysis of Printed Documents", in H.S. Baird, H. Bunck and K. Yamamoto (Editors), *Structured Document Image Analysis*, Springer-Verlag Berlin Heidelberg, 1992, pp. 70-98.
- [82] A. Belaïd, *Analyse et Reconnaissance de Documents, Le Traitement Electronique du Document, Cours INRIA*, 3-7 Oct. 1994, Aix-en-Provence, pp.49-92.
- [83] K. Etemad, D. Doerman, and R. Chellappa, Multiscale Segmentation of Unstructured Document Pages Using Soft Decision Integration, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol. 19, n° 1, Jan. 1997.
- [84] P. Chauvet, J. Lopez-Krahe, E. Taflin, and H. Maitre, "System for intelligence Office Document Analysis, Recognition, and Description," *Signal Processing*, Vol. 32, 1993, pp.161-190.
- [85] G. Nagy, S. Seth, and M. Viswanathan, "DIA, OCR, and the WWW," *Draft for 8th International Conference on Image Analysis and processing*, San Remo, April 4, 1995.

- [86] R. Ingold and D. Armangil, A Top-Down Document Analysis Method for Logical Structure Recognition, in Proceedings of the 1st International Conference on Document Analysis and Recognition, Vol. 2, St-Malo, France, 1991, pp. 41-49.
- [87] A. Dengel, R. Bleisinger, F. Fein, R. Hoch, F. Hönes and M. Malburg, OfficeMAID : A System for Office Mail Analysis, Interpretation and Delivery,” in Proceedings of DAS'94: 1st IAPR Workshop on Document Analysis Systems, Kaiserslautern, Germany, 18-20 October, 1994, pp. 253-275.
- [88] A. Belaïd, J. C. Anigbogu and Y. Chenevoy, “Qualitative Analysis of Low-Level Logical Structures,” in Proceedings of the Fifth International Conference on Electronic Publishing, Document Manipulation and Typography, 13-15 April 1994, Darmstadt, Germany, pp. 435-446.
- [89] T. Watanabe, Q. Luo and N. Sugie, Knowledge for Understanding Table-Form Documents, IEICE Transactions on Information and Systems,” Vol. E77-D, No. 7, July 1994, pp. 761-769.
- [90] Kok-Lay Heng, Analyse d'images de lignes : des pixels aux primitives, thèse de doctorat d'état, paris 6, 1988, 90p.
- [91] Abdel.Belaïd and Y.Belaïd, « Reconnaissance de formes, Inter-Edition, Paris, 1992, 412p.
- [92] S.R. Ramesh, A generalized character recognition algorithm: A graphical approach, Pattern recognition, 22(4), 1989.
- [93] R.A. Wagner, order-n correction for regular languages, communication of the ACM, 17(5), may 1974.
- [94] A.Belaïd, K.Tombre, “Analyse de documents: de l'image à la sémantique”, colloque national sur l'écrit et le document, Nancy, juillet 1992.
- [95] T.L. Abegnoli, “Quatre phases de reconnaissance de caractères”, Revue Electronique n°29, France, juin 1993.
- [96] J.D. Becker, “Arabic word processing”, Communications of the ACM Vol. 30 n°7, Juillet 1987.
- [97] K.S.Fu, “Syntactic Pattern Recognition and Applications, Prentice Hall, Englewood Cliffs, New Jersey, 1982.
- [98] J.Mantas, “An overview of character Recognition Methodologies, Pattern Recognition, 19(6): 425-430, 1986.
- [99] S.N. Srihari, Computer Text Recognition and Error Correction, IEEE Computer Society Press, Silver Spring, MD, 1984.
- [100] V.K. Govindan and A.P. Shivaprasad, “Character Recognition- a Review”, Pattern Recognition, 23(7):671-683,1990.
- [101] James G. Février, Histoire de l'écriture, p. 273, Payot édition, Paris, 1959.
- [102] Roland Meynet, L'écriture Arabe en question, p.24, Edition Dar el Machreq, Beyrouth, Liban, 1971.
- [103] K.Bouhlila, M.K Hamrouni and N.Ellouze, Method of segmentation of Arabic text image into characters, First Kuwait Conference, Mars 1989.
- [104] M.B KURDY & K. WADOUH, Transfer the Arabic OCR to product (first stage), internal report (in arabic), Sept.1991.
- [105] R.C Gonzalez and P. Wintz, Digital Image Processing, Addison Wesley 1987.
- [106] M.K Hu, Visual pattern recognition by moment invariant, I.R.E Trans. Inf. Theory. IT, 8 Feb. 1962
- [107] S. Thome, Prétraitement du chiffre manuscrit, Actes du 1^{er} congrès AFCET : Reconnaissance des formes et intelligence artificielle, pp. 568-576, 1978
- [108] P.Marthon, A.Bruel et G.Biguet, Squelettisation par calcul d'une fonction discriminante sur un voisinage de 8 pixels, Actes du 2^{ème} congrès AFCET : Reconnaissance des formes et intelligence artificielle, pp.107-114, Toulouse, 1979.

- [109] S.Y. LU, A tree-to-tree distance and its application to cluster analysis, IEEE Transaction on PAMI Vol. PAMI.1 n°2 Avril 1977.
- [110] Michel Lucas, La réalisation des logiciels graphiques interactifs, édition, 1979.
- [111] A.Poor, Looking at the Tiff specifications from the inside, PC magazine, décembre 1991.
- [112] S.N. Srihari, E. Cohen, J.J. Hull and L. Kuan, A system to locate and recognize ZIP codes in handwritten addresses, International journal of research and engineering, postal applications, 1989.
- [113] C.Touzet, Les Réseaux de neurones, introduction au connexionnisme cours, exercice, TP, 1992
- [114] Alexandre Aupetit, "Les réseaux de neurones artificiels", Edition 1997.
- [115] E.Davalo et P.Naim, "Des réseaux de neurones", Edition Eyrolles, 1993.
- [116] J.Hérault, C.Jutten, "Réseaux neuronaux et traitement du signal", 1994.
- [117] L.Hamami-Mitiche, Système de Reconnaissance de l'écriture Arabe imprimée utilisant la méthode des arbres, COMAEI'96 Tlemcen, Algérie
- [118] L. Hamami-Mitiche, OCR System of writing Arabic, International Association of Science and Technology for Development IASTED'97, International Conference Signal and Image Processing SIP'97, Nov.1997, îles canaries – Espagne.
- [119] L.Hamami-Mitiche, Système de Reconnaissance de l'écriture arabe imprimée, Proceedings XVII^{èmes} JTEA'97 Conférence Internationale, Hammamet-Nabeul Tunisie, Nov.1997, pp.357-362.
- [120] L. Hamami-Mitiche, Recognition System of writing printed Arabic, International Conference on Graphics and Imaging CGIM'99, Oct. 1999, Los-Angeles (U.S.A).
- [121] I.S.I Abuhaiba, M.J.J. Holt and S. Datta, Processing of off-line handwritten text: Polygonal Approximation and Enforcement of Temporal Information, CVGIP: Graphical Models and Image Processing, vol.56, n°4, July 1994, pp.324-335.
- [122] I.S.I Abuhaiba, S.A. Mahmoud and R.J. Green, Recognition of handwritten cursive Arabic characters, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.16, n°6, 1994, pp.664-672.
- [123] Al-Emami and M. Usher, On-line recognition of handwritten Arabic characters, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.12, n°7, 1990, pp.704-710.
- [124] A.Alimi and O. Ghorbe, The analysis in an On-line recognition system; of Arabic Handwritten Characters, 3rd International Conference on Document Analysis and recognition, Canada, 890-893, 1995.
- [125] A. Amin and H. Al Sadoun, Handprinted Arabic character recognition system using an artificial neural network, Pattern Recognition, 29 (4), 1996, pp.663-675.
- [126] R.G. Casey and E. Lecolinet, Strategies in character segmentation: A survey, 0-8186-7128-9/95\$4.00,1995 IEEE, pp.1028-1033.
- [127] T.S. El-Sheikh and S.G. El-Taweel, Real-time Arabic Handwritten Character Recognition, Pattern Recognition, vol.23, n°12,1990, pp.1323-1332.
- [128] M.S. El-Wakil and A.A. Shoukry, On-line recognition of handwritten isolated Arabic characters, Pattern Recognition, vol.22, n°2, 1989, pp.97-106.
- [129] T.S. El-Sheikh and R.M. Guindi, Automatic recognition of isolated Arabic characters, Signal Processing, vol.14, 1988, pp. 177-184.
- [130] H.Goraine, M.Usher and S.Al-Emami, Off-line Arabic Character Recognition, Computer, Jul. 1992, pp.71-74.
- [131] F. Haj-Hassan, Arabic Character Recognition, Arab School (Computer and the Arabic language), 1985.
- [132] A.Hoekstra, A.Tholen, R.P.W.Dwin, Estimating the reliability of neural network classifications, pattern recognition group, Delft University of technology, The Netherlands, 1996.

- [133] S. Holder and J. Dengler, Font and Size invariant Character Recognition with Grey value Image Features, CH2614-6/88/0000/0252\$01.00,1988 IEEE, pp.252-254.
- [134] A. Khotanzad and Yaw Hua Hong, Invariant Image Recognition by Zernike Moments, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.12, n°5, May 1990, pp.489-497.
- [135] S.A. Mahmoud, I.S.I Abuhaiba, and R.J. Green, Skeletonization of Arabic characters using clustering based skeletonization algorithm (CSBA), Pattern Recognition, vol.24, n°5, 1991, pp.453-464.
- [136] M.A. Mahdjoub, Choix des paramètres liés à l'apprentissage dans la reconnaissance en ligne des caractères arabes par les chaînes de Markov cachées. In. Forum de la recherche en informatique, Tunis, juillet 1996.

- [137] A. Nouh, N. Ula and A. Sharaf Eldin, A Proposed Algorithm For the Thinning Binary Arabic Character Patterns, First Kuwait Computer Conference. Mars 1989.
- [138] K.Pakker and A.Chehikian, Reconnaissance de caractères alphanumériques multipolice par analyse structurelle hétéroarchique. Congrès AFCET-INRIA RFIA, Paris, 1985.
- [139] K.Romeo-Pakker and A.Ameur, Une méthode rapide de segmentation et de reconnaissance de caractères manuscrits arabes, 14^{ème} colloque GRETSI- Juan-les-pins, Sept. 1993.
- [140] R. Ingold, Une nouvelle approche de la lecture optique intégrant la reconnaissance des structures de documents, Thèse de Doctorat es sciences, EPFL, Lausanne Suisse, 1989.

Annexes

Description du format Tiff

La spécification Tiff (Tagged Image File Format), utilisée par les fichiers Tiff a été développée à partir de 1986 par Aldus Corp. et un groupe d'autres compagnies principalement Scanner Manufactories telles que DCST Corp., Hewlett Packard Co. Microsoft Corp, Microtek International, New Image Technology et Xerox Imaging Systems/Data copy [111].

1- Fichier Tiff :

La grande force du format Tiff est sa flexibilité. Les images peuvent être en couleurs, en niveaux de gris, d'une taille aussi grande que désirée.

Un seul fichier peut contenir plusieurs images (ou la même à des résolutions différentes). Cette flexibilité est due à l'utilisation systématique de pointeurs au sein du fichier. Ainsi un fichier Tiff ne présente pas une structure figée mais, de manière identique à celle des répertoires d'un disque, une organisation qui est elle-même décrite au sein du fichier.

2- Exploitation du fichier Tiff :

Tout type de fichier informatique a des règles qui gouvernent sa structure ; pour pouvoir exploiter ce type de fichier, il faudra connaître ses règles.

Le plus simple des fichiers Tiff est constitué de trois principales parties comme il est présenté dans la figure A1.

a) Première partie : l'entête du fichier composé de huit octets et contenant trois informations, à savoir :

- Les deux premiers octets 0 et 1 contiennent le code ASCII du caractère I pour Intel ou M pour Motorola.
- Les octets 2 et 3 contiennent le numéro de la version (42 pour une compatibilité éventuelle)
- Les octets 4, 5, 6 et 7 contiennent un pointeur vers la 2^{ème} partie, c'est à dire le premier IFD.

b) Deuxième partie : l'IFD (Image File Directory) qui constitue le répertoire du fichier et qui contient des informations sur l'image comme : les dimensions, les couleurs, la compression...

Au début d'un IFD, on trouve le nombre de tags ou champs que contient cet IFD, écrit sur deux octets ; juste après ces deux octets commencent les tags.

Un tag est un groupe de 12 octets qui représentent une information et qui sont divisés en 4 parties :

- Les octets 0 et 1 désignent le type de l'information représentée par le tag (largeur de l'image, sa hauteur, le nombre de bits par pixel, le format de compression, etc....).

Exemple : Si on trouve la valeur 256, alors ce champ contient la largeur de l'image.

- Les octets 2 et 3 contiennent l'information sur le type de la donnée que représente le tag.

Exemple : La donnée peut être de type byte ou en code ASCII ou encore de type short (16 bits non signés) ou long (32 bits non signés), elle peut être aussi de type rationnel.

- Les octets 4, 5, 6 et 7 contiennent le nombre de données, s'il s'agit par exemple de la longueur de l'image, ce nombre est égal à 1 ; si la donnée représente une chaîne de caractères alors le nombre de données sera la longueur de cette chaîne.
- Les octets 8, 9, 10, et 11 contiennent la donnée elle-même. Si la donnée ne tient pas sur 4 octets alors ces 4 octets contiennent un pointeur vers la donnée (bien évidemment en dehors de l'IFD).

Un IFD se termine par 4 octets qui contiennent un pointeur vers un autre IFD d'une autre image dans le même fichier. Un IFD peut contenir jusqu'à 45 tags, ce nombre dépend de la richesse de l'image en informations, l'ordre des tags n'est pas important.

c) ***Troisième partie : le Bitmap*** qui est l'image elle-même, toutes les informations sont dans la partie précédente. Notons que chaque octet de cette partie peut représenter M pixels avec : $M=8/Bpp$ où Bpp représente le nombre de bits par pixel.

En ce qui nous concerne les plus importantes informations sont :

Les dimensions de l'image (sa largeur et sa hauteur), le nombre de bits par pixel, le pointeur vers le Bitmap.

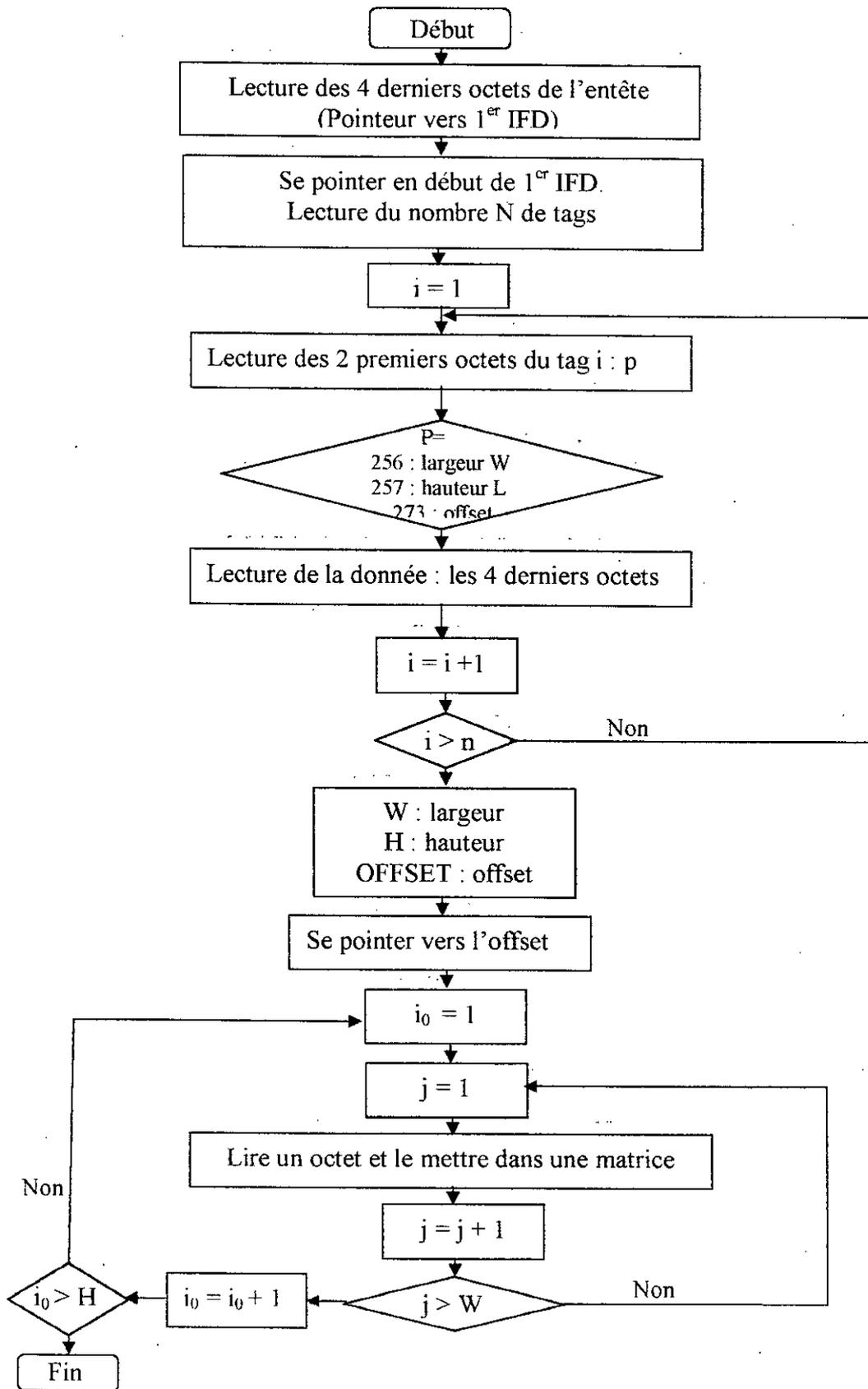


Figure A2 : Lecture d'un fichier Tiff

Technique de normalisation de la taille :

La taille des caractères peut varier d'une fonte à l'autre et même au sein d'une même fonte après agrandissement ou réduction, ce qui peut causer une instabilité des paramètres. Une technique naturelle de prétraitement consiste à ramener les caractères à la même taille. Nous allons donner un algorithme de normalisation de la taille.

Algorithme de Srihari [112]

Cet algorithme opère en deux étapes. La première normalise le caractère en hauteur et la seconde, en largeur. L'ordre de normalisation ainsi choisi évite que les caractères fins comme « 1 » et « l » ne se déforment par rapport à des caractères épais.

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

$$p = h / h_i \quad \text{et} \quad l' = p * l_i$$

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

- Si $l' < l$, alors l'image normalisée en hauteur est centrée dans une surface de dimension h, l .
- Si $l' > l$, alors l'algorithme balaye cette image et assigne à noir tout pixel de coordonnées $(x/l' * l, y)$ si le pixel (x, y) l'est aussi.

Algorithme de reconnaissance :

Une fois que les caractéristiques principales et secondaires du caractère à reconnaître sont extraites, on procède à l'identification suivant l'algorithme de reconnaissance suivant :

Les caractéristiques principales nous donnent la classe du caractère. On se place dans le dictionnaire au début de cette classe et on charge les vecteurs de caractéristiques de chaque prototype, on aura :

$CAi[j]$: $j^{\text{ème}}$ caractéristique du $i^{\text{ème}}$ prototype

Avec $j=1...5$; $i = 1..Nbpr$

$Nbpr$ = le nombre de prototypes de la classe

On cherche les prototypes qui ont les mêmes caractéristiques que le caractère inconnu et cela en calculant la distance $Dist$ qui est définie comme suit :

$$Dist [i] = \sum_{j=1}^5 abs(CAx[j] - CAi[j])$$

Avec: $i = 1...Nbpr$.

$CAx [j]$: $j^{\text{ème}}$ caractéristique du caractère à reconnaître

Les prototypes qui vérifient $Dist[i] = 0$ sont mis dans une liste appelée $Liste1$, qui contient k prototypes.

Si $k = 1$, le caractère est identifié

Si $k > 1$ alors :

Si tous les prototypes décrivent le même caractère, c-à-d ont le même nom

alors le caractère est identifié sinon il y a ambiguïté.

Si $k = 0$ alors on met dans une deuxième liste appelée $Liste2$ les prototypes qui vérifient : $D = 0$

$$D[i] = \sum_{j=3}^5 abs(CAx[j] - CAi[j])$$

Avec : $i = 1...k1$

$k1$: représente le nombre de prototypes de $Liste2$

Si $k1 = 0$ alors le caractère est rejeté

Si $k1 = 1$ alors on calcule les distances $Dist1$ et $Dist2$

Si $Dist1 \leq 1$ et $Dist2 \leq 1$ alors le caractère est reconnu

Sinon il est rejeté

Si $k1 > 1$ on met dans une liste appelée $Liste3$ tous les prototypes qui vérifient :

$$(Dist1 \leq 1) \text{ et } (Dist2 \leq 2)$$

On obtient k_2 caractères

Si $k_2 = 1$ alors le caractère est reconnu

Si $k_2 = 0$ alors le caractère est rejeté

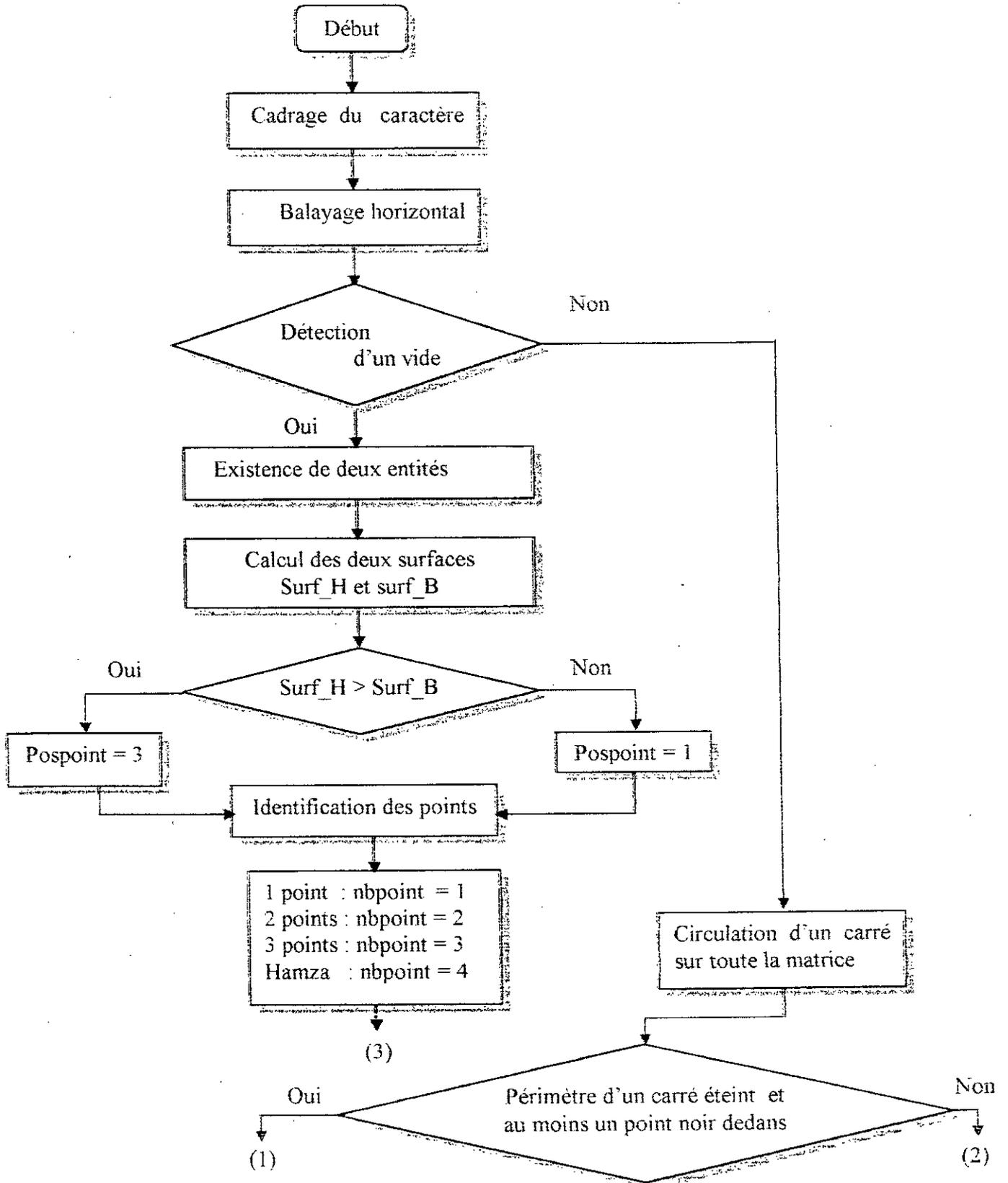
Si $k_2 > 1$ on calcule le nombre d'occurrences de chaque caractère.

S'il y a un seul prototype qui a la fréquence d'occurrence la plus élevée alors :

Le caractère est reconnu si non il y a une ambiguïté

Organigramme récapitulatif de l'étape de reconnaissance :

Nous avons jugé utile de donner l'organigramme de toutes les démarches de l'opération de reconnaissance :



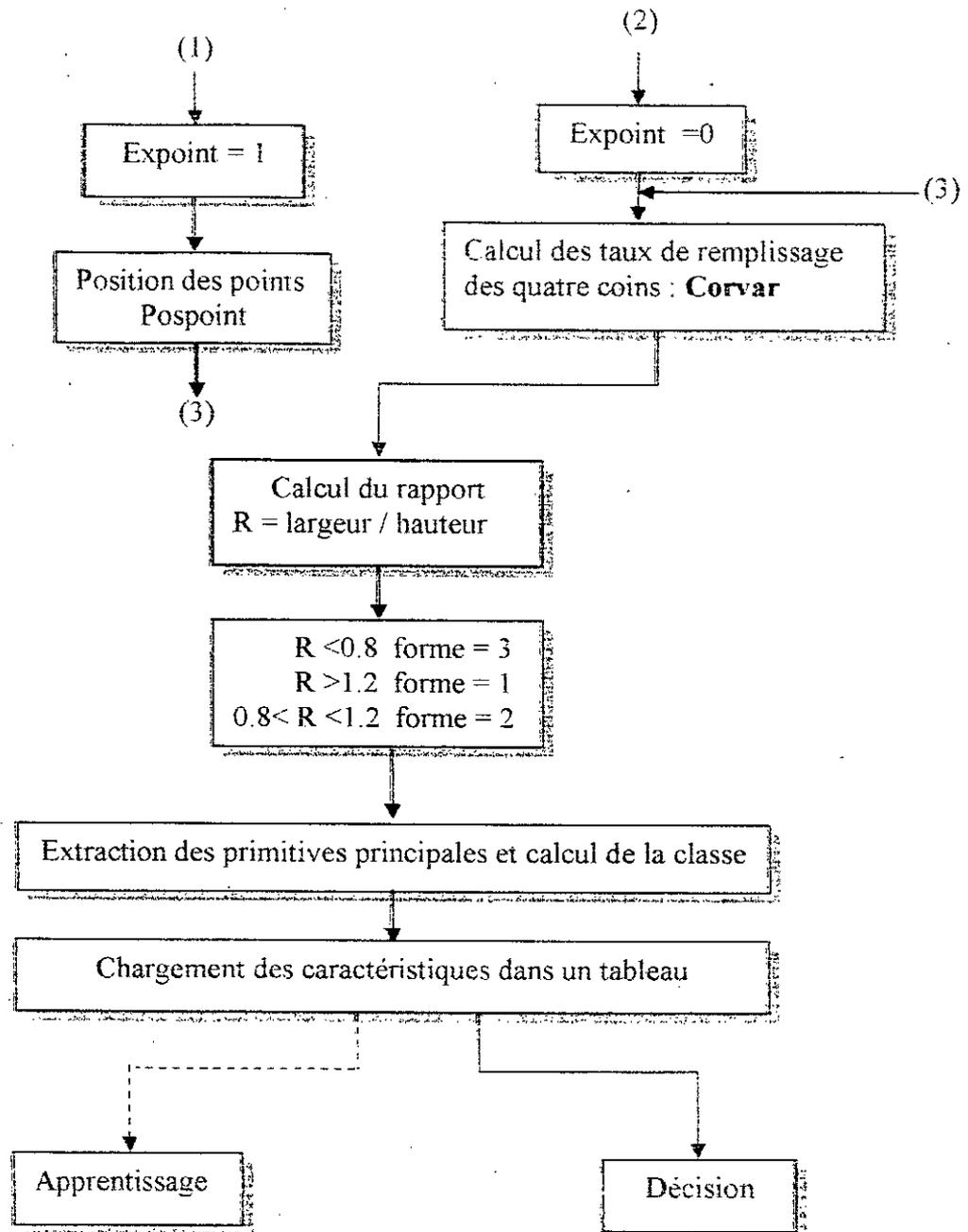


Figure C1: Organigramme de la reconnaissance

Etude des Réseaux de Neurones multi couches avec rétro propagation du gradient

- **Le neurone biologique [113]:**

La connaissance de la structure interne du cerveau et la compréhension de son fonctionnement a été d'une extrême importance pour le développement des réseaux de neurones artificiels. Nous allons, dans ce qui suit, décrire succinctement l'élément de base du système nerveux central : le neurone.

- **Définitions [113] :**

Le neurone est l'unité fonctionnelle de base du système nerveux. Le cerveau se compose d'environ 10^{12} neurones, avec 1000 à 10000 synapses (connexions) par neurone. Les neurones sont caractérisés par cinq fonctions de base qui sont :

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

- **Structure du neurone biologique [113] :**

Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone. L'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. La jonction entre deux neurones est appelée la synapse (FigureD1)

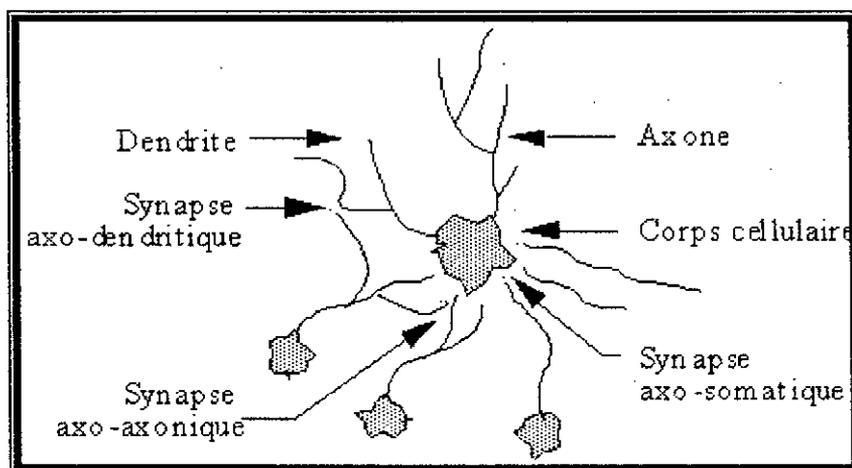


Figure D1 : Un neurone avec son arborisation dendritique [113].

ANNEXE D : Les réseaux de neurones

• Les réseaux de neurones artificiels :

Copier le cerveau restera pour longtemps une ambition exagérée, mais vouloir s'inspirer des architectures et des fonctions du système nerveux n'est pas un rêve inaccessible ; c'est une réalité.

Les réseaux de neurones biologiques réalisent facilement un certain nombre d'applications telles que la reconnaissance de formes, le traitement du signal, l'apprentissage par l'exemple, la mémorisation, la généralisation. Ces applications sont pourtant, malgré tous les efforts déployés en algorithmique et en intelligence artificielle, à la limite des possibilités actuelles. C'est à partir de l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base du cerveau que les réseaux de neurones artificiels se sont développés. Les réseaux de neurones artificiels sont des modèles, à ce titre ils peuvent être décrits par leurs composants, leurs variables descriptives et les interactions des composants.

• Qu'est-ce qu'un neurone artificiel ? Qu'est-ce qu'un réseau de neurones ?

Ce que l'on désigne habituellement par «réseau de neurones» ou réseau neuromimétique est un réseau de neurones artificiels basé sur un modèle simplifié du neurone. Ce modèle permet certaines fonctions du cerveau, comme la mémorisation associative, l'apprentissage par l'exemple, le travail en parallèle, etc. [114].

La Figure D2 montre la structure d'un neurone artificiel. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amonts. A chacune de ces entrées est associé un poids w abréviation de weight (poids en anglais) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque connexion est associé un poids [113].

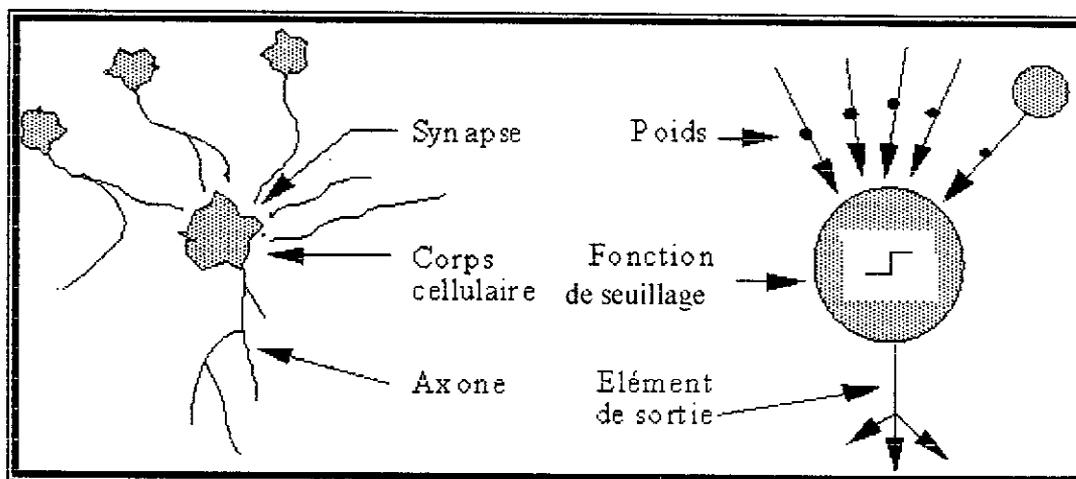


Figure D2 : Mise en correspondance neurone biologique/neurone artificiel [113].

ANNEXE D : Les réseaux de neurones

Ces réseaux sont capables de séparer les entrées en différentes classes, en leur présentant successivement des exemples, sans avoir besoin de lui apprendre des règles logiques ni de stocker des données.

- **Le neurone formel [115]:**

En 1943, Mc Culloch et Pitts inventèrent le premier modèle de neurone artificiel (Figure D3). Inspiré du neurone biologique, ce modèle n'était rien d'autre qu'un produit scalaire entre un vecteur d'entrées $X(x_1, \dots, x_i, \dots, x_p)$ et un vecteur poids $W(w_1, \dots, w_i, \dots, w_p)$, avec p : le nombre d'entrées, suivi d'une fonction f de seuillage.

Voici comment le neurone est modélisé mathématiquement :

$$S = f\left(\sum_{i=1, p} w_i * x_i\right) \quad \dots(1) \quad \text{Avec } f : \text{fonction de seuillage.}$$

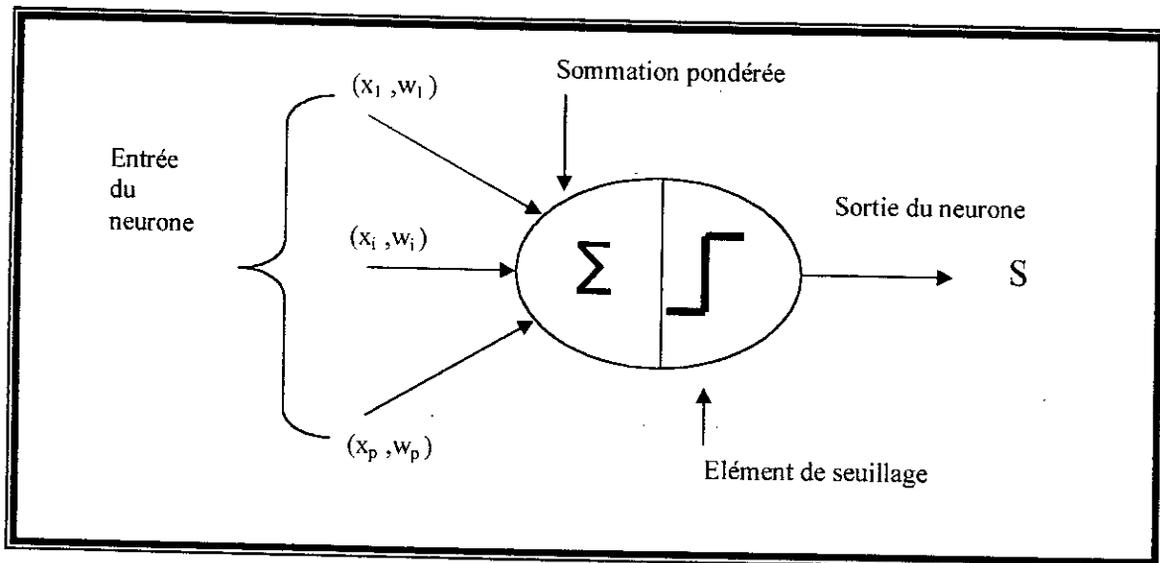


Figure D3 : Modèle du neurone formel de Mc Culloch et Pitts [115].

- **La fonction de seuillage [115]:**

La fonction de seuillage f (d'activation ou d'état) définit son état interne en fonction de ses entrées totales. La fonction d'activation peut être "Discrete" ou "Continue".

➤ **Fonction binaire à seuil (figures D4 et D5):**

Les premiers modèles de neurones étaient caractérisés par une fonction d'activation à seuil simple (binaire : 0=inactive, 1=active). Le déclenchement de l'activité intervient si la somme des excitations (activité des neurones excitateurs pondérée par les poids synaptiques) dépasse un certain seuil propre au neurone.

Fonction de Heaviside : $f(x) = 1$, si $x > \text{SEUIL}$.

$$f(x) = 0, \text{ sinon.}$$

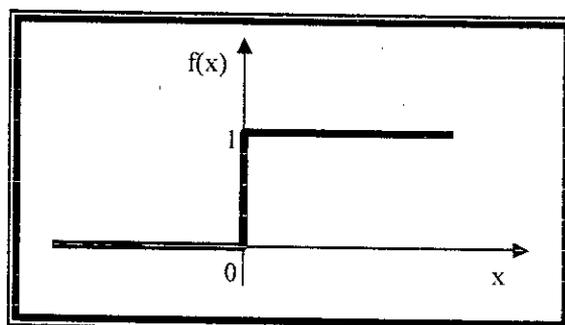


Figure D4 : Fonction de Heaviside.

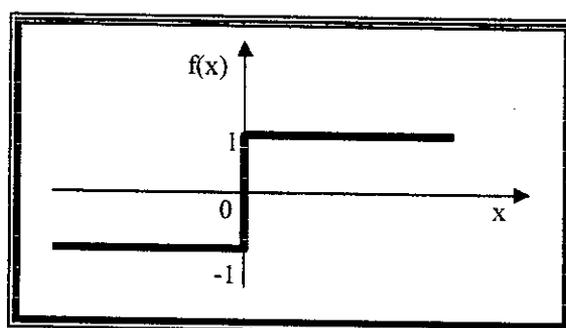


Figure D5 : Fonction signe.

Le modèle binaire a été rapidement perfectionné (fonctions sigmoïdes...), mais l'effet de seuil reste souvent recherché car il confère une certaine efficacité aux réseaux et parce qu'il semble correspondre à **un modèle proche de la réalité**.

➤ Fonction sigmoïde :

Lorsque la fonction d'activation est dérivable, il est possible d'utiliser des algorithmes d'apprentissage performants, basés sur la recherche d'un minimum de la fonction d'erreur, notamment la "rétropropagation du gradient" pour les perceptrons multicouches (voir plus loin).

Une fonction est "Sigmoïde" lorsqu'elle est monotone croissante, dérivable et bornée. Ce type de fonction combine l'avantage de l'effet de seuil et de la dérivabilité. Les fonctions sigmoïdes sont donc très utilisées, notamment dans le cas du perceptron multicouches.

Fonction sigmoïde exponentielle (ou logistique) : $f(x) = \frac{1}{1 + \text{EXP}(-x)}$

Dérivée : $f'(x) = f(x) \cdot (1 - f(x))$

Elle est la plus utilisée du fait que sa dérivée est simple à calculer.

Fonction sigmoïde tangentielle : $f(x) = \text{TANH}(x)$

Dérivée : $f'(x) = 1 - f(x)^2$

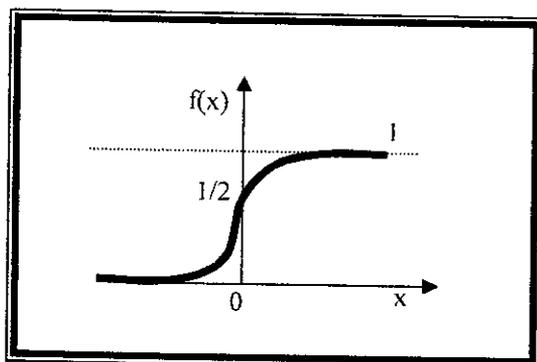


Figure D6 : Fonction logistique.

- **Un peu de vocabulaire [114, 115] :**

- **Apprentissage, mémoire et oubli :**

Une caractéristique des réseaux de neurones est leur capacité d'apprendre (par exemple à reconnaître une lettre, un son...). Mais cette connaissance n'est pas acquise dès le départ, ces réseaux apprennent par l'exemple de la même manière qu'un enfant apprend à reconnaître un oiseau à partir d'exemples d'oiseaux qu'il a vus. Ils ont donc une capacité à classer, généraliser, à mémoriser et à oublier. Un réseau de neurones reconnaîtra d'autant plus facilement un objet qu'il l'aura vu souvent.

Définition : L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré.

- **Connectivité :**

La connectivité des réseaux, est la manière dont les neurones sont connectés. Elle peut être totale (tous les neurones sont connectés entre eux) ou par couches (les neurones d'une couche ne sont connectés qu'aux couches suivantes). Il existe des réseaux multicouches (Figure D7) ou monocouche (Figure D8).

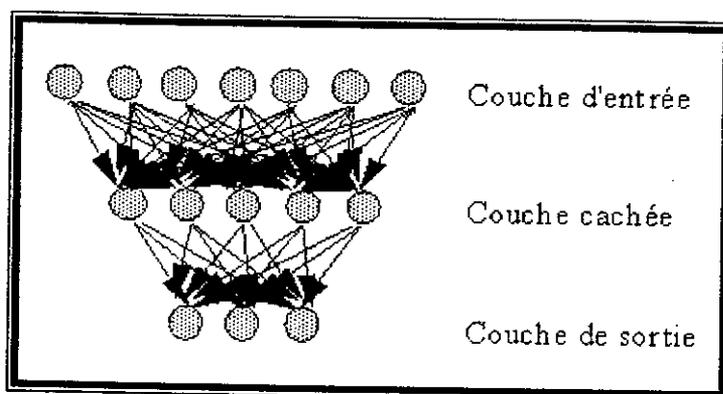


Figure D7 : Définition des couches d'un réseau multicouche.

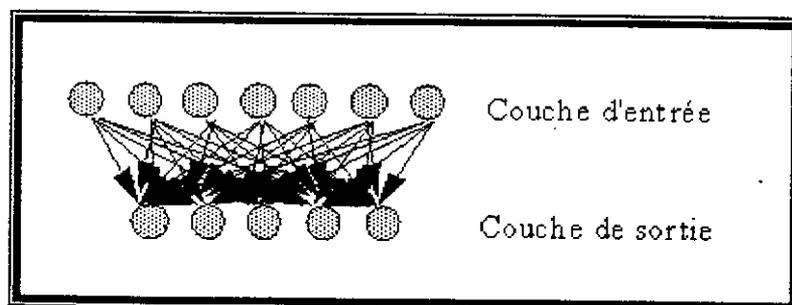


Figure D8 : Définition des couches d'un réseau monocouche.

➤ **Calcul des poids synaptiques :**

La rétropropagation est une méthode de calcul des poids pour un réseau à apprentissage supervisé qui consiste à minimiser l'erreur quadratique de sortie (somme des carrés de l'erreur de chaque composante entre la sortie réelle et la sortie désirée). D'autres méthodes de modification des poids sont plus "locales", chaque neurone modifie ses poids en fonction de l'activation ou non des neurones proches. C'est le cas des réseaux suivant la règle de Hebb ou les réseaux de Kohonen (apprentissage non supervisé).

➤ **Architecture des réseaux [115]:**

Les structures utilisées sont très variées. Si l'on se réfère aux études biologiques du cerveau, on constate que le nombre de connexions est énorme.

D'une manière générale, l'architecture d'un réseau définit son fonctionnement. Il existe trois types de réseaux à architectures différentes.

❖ **Réseaux statiques :**

Ce type de réseau est organisé en couches de neurones (Figure D9). Chaque neurone d'une couche reçoit ses entrées à partir des neurones de la couche précédente. Dans de tels réseaux il n'existe pas de retour d'information appelé «feed-back ». Ces réseaux peuvent être utilisés pour des problèmes de classification.

La première couche s'appelle couche d'entrée, la dernière couche de sortie, les autres s'appellent couches cachées.

❖ **Réseaux dynamiques :**

L'introduction d'une structure appelée feed-back entre les neurones a rendu le réseau dynamique (Figure D10), ce qui a nécessité l'étude de la stabilité de ce dernier.

Ces réseaux sont exploités pour l'identification ou la commande de systèmes dynamiques.

❖ **Réseaux à architecture évolutive et réseaux Auto Organisés :**

- Les réseaux sont dits évolutifs si la dimension du réseau change pendant l'apprentissage ou l'entraînement. Ainsi le nombre de neurones augmente ou diminue.

ANNEXE D : Les réseaux de neurones

- Les réseaux sont dits auto-organisés s'ils changent leurs structures internes pendant l'utilisation. Ainsi les neurones se regroupent topologiquement suivant la représentation des exemples issus de l'espace d'entrée.

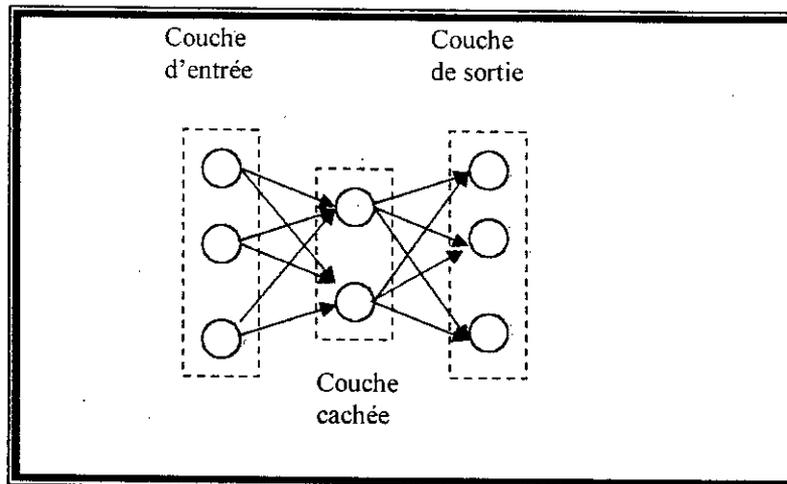


Figure D9 : Réseau statique (à couches) [115].

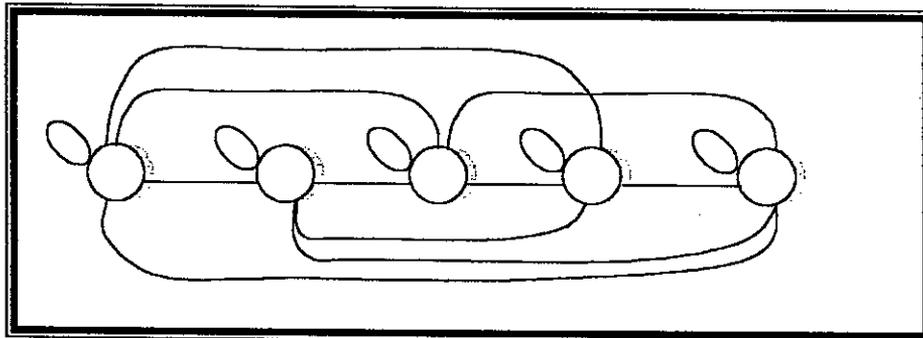


Figure D10 : Réseau dynamique [115].

- **Apprentissage supervisé et non supervisé :**

Les réseaux de neurones se divisent en deux principales classes, les réseaux à apprentissage supervisé et les réseaux à apprentissage non supervisé.

Pour les réseaux à apprentissage supervisé (Perceptron, Adaline, etc...), on présente au réseau des entrées, et en même temps les sorties que l'on désirerait pour cette entrée. Par exemple on lui présente en entrée des caractéristiques du caractère et en sortie son code ASCII. Le réseau doit alors se reconfigurer, c'est à dire calculer ses poids afin que la sortie qu'il donne corresponde bien à la sortie désirée.

Pour les réseaux à apprentissage non supervisé (Hopfield, Kohonen, etc...), on présente une entrée au réseau et on le laisse évoluer librement jusqu'à ce qu'il se stabilise.

❖ Réseaux à apprentissage supervisé :

Les réseaux à apprentissage supervisé représentent la classe des réseaux de neurones qui a fait l'objet du plus grand nombre de travaux de recherche et qui a connu une grande évolution depuis les années 1980.

L'histoire de ce type de réseaux commence par le perceptron de Rosenblatt en 1957 puis ce fut Widrow et Hoff avec l'ADALINE en 1960. Après une époque très difficile, c'est Rumelhart, Hinton et Williams qui ont ouvert une nouvelle ère avec l'algorithme de rétropropagation du gradient et les réseaux multicouches perceptron en 1986. C'est à partir de cette date que les réseaux à apprentissage supervisé ont connu un grand essor.

Règles d'apprentissage :

L'apprentissage d'un réseau de neurones supervisé revient à modifier les poids de ses connexions. En général, la modification d'un poids w_{ij} se fait de la façon suivante : $w_{ij} = w_{ij} + \Delta w_{ij}$. Dans la suite, nous allons expliquer comment se fait la réadaptation de ces poids (Règle de Widrow-Hoff et l'algorithme de rétropropagation du gradient), c'est à dire comment calculer le terme Δw_{ij} . Les calculs seront conduits de façon générale avec une fonction de seuillage f simplement supposée dérivable (on utilise fréquemment avec ce type de réseaux une fonction de type sigmoïde).

a) Règle de Widrow-Hoff [116]:

Ce type de règle est utilisé pour les réseaux à une couche. On considère un réseau constitué de n neurones recevant des vecteurs X à p composantes $\{x_1, \dots, x_p\}$ (figure D11). Les p entrées x_k du réseau sont distribuées sur tous les neurones. La sortie du neurone i vaut :

$$y_i = f(p_i) = f\left(\sum_{j=1}^p w_{ij} x_j\right) \dots\dots\dots(1)$$

Avec p_i le potentiel du neurone i .

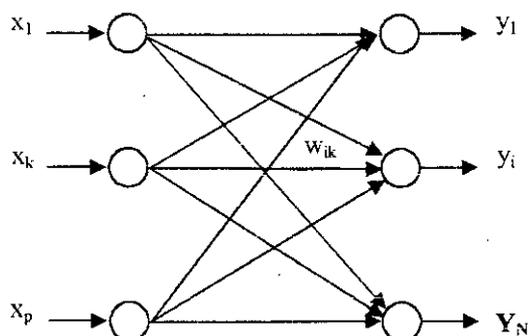


Figure D11 : Réseau à une couche.

A un vecteur d'entrée X, on veut associer un vecteur de sortie Yd (y_{d1}, \dots, y_{dn}) (sortie désirée). Si les poids w_{ik} ont des valeurs quelconques (ce qui est le cas initialement et avant la fin de l'apprentissage), le vecteur de sortie y observé est a priori différent de yd. On peut associer à cette différence l'erreur quadratique :

$$E = \frac{1}{2} \sum_{j=1}^n (y_j - y_{d_j})^2 \quad \dots\dots\dots(2)$$

Calculons le gradient de cette erreur par rapport à w_{ik} :

$$\frac{\partial E}{\partial w_{ik}} = \sum_{j=1}^n (y_j - y_{d_j}) \frac{\partial (y_j - y_{d_j})}{\partial w_{ik}} \quad \dots\dots\dots(3)$$

La dérivée partielle du membre de droite est nulle sauf pour $j=i$ car seul la sortie y_i est fonction du poids w_{ik} . De plus, la sortie désirée, fournie par le superviseur, ne dépend d'aucun poids.

On arrive finalement à :

$$\frac{\partial E}{\partial w_{ik}} = (y_i - y_{d_i}) \frac{\partial y_j}{\partial w_{ik}} = (y_i - y_{d_i}) x_k f'(p_i) \quad \dots\dots\dots(4)$$

où f' dérivée de f .

En posant $\delta_i = (y_i - y_{d_i}) \cdot f'(p_i)$, le gradient de l'erreur quadratique E s'écrit :

$$\frac{\partial E}{\partial w_{ik}} = \delta_i \cdot x_k \quad \dots\dots\dots(5)$$

et la mise à jour du poids, selon le principe du gradient, s'écrit :

$$\Delta w_{ik} = -a \cdot \delta_i \cdot x_k \quad \dots\dots\dots(6)$$

où a est un gain d'adaptation positif appelé **taux d'apprentissage**.

b) Algorithme de rétropropagation du gradient [116] :

Considérons le réseau multicouche (2 couches) de la figure D12. On veut comme précédemment associer un vecteur de sortie yd à un vecteur d'entrée x. La sortie observée est $y \neq y_d$. On peut y associer le terme d'erreur quadratique E comme précédemment.

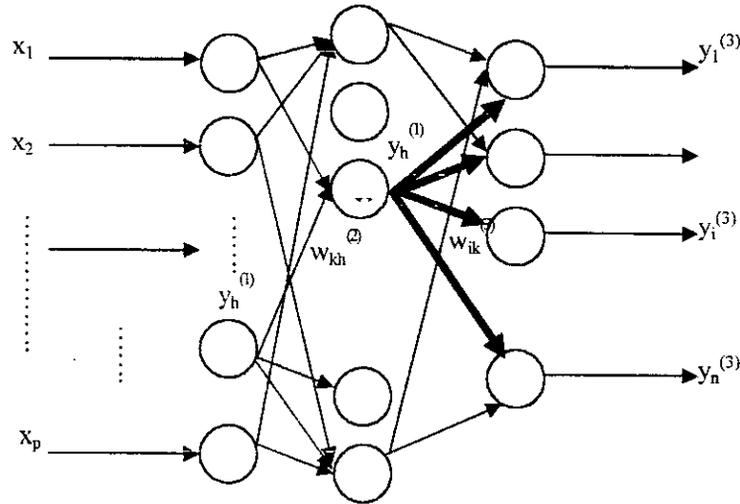


Figure D12 : Réseau de neurones multicouches
(Le numéro de la couche est indiqué en indice haut entre parenthèse).

Nous allons montrer que nous pouvons modifier les poids du réseau en minimisant cette erreur quadratique de façon analogue à celle de Widrow-Hoff. Cependant, si cette méthode est directement applicable pour ajuster les poids de la dernière couche, elle ne l'est pas pour les couches internes car on ne connaît pas la sortie désirée pour ces couches, et par conséquent on ne connaît pas directement le terme d'erreur associé à chaque couche interne. Il faut donc exprimer l'erreur à la sortie de chaque neurone d'une couche quelconque à partir de l'erreur de la dernière couche, seul erreur mesurable.

Ceci est possible par un calcul différentiel simple. On peut calculer l'erreur $\delta_k^{(j)}$ associée à un neurone k d'une couche quelconque j à partir des erreurs des neurones de la couche suivante(j+1) pondérées par les poids $w_{ik}^{(j+1)}$ des connexions partant du neurone k vers les neurones de la couche suivante (en gras sur la figure D12). L'erreur $\delta_k^{(j)}$ s'exprime par :

$$\delta_k^{(j)} = f'(p_k^{(j)}) \cdot \sum_{i \in \text{couche } j+1} w_{ik}^{(j+1)} \cdot \delta_i^{(j+1)} \dots\dots\dots(7)$$

Avec le potentiel qui s'exprime par : $p_k^{(j)} = \sum_{h \in \text{couche } j-1} w_{kh}^{(j)} y_h^{(j-1)}$

Pour plus de détail sur le calcul du terme $\delta_k^{(j)}$ voir l'annexe E.

En appliquant la méthode du gradient, la modification des poids se fait selon la règle :

$$\Delta w_{kh}^{(j)} = -a \cdot \delta_k^{(j)} \cdot y_h^{(j-1)} = -a \cdot \left[f'(p_k^{(j)}) \cdot \sum_{i \in \text{couche } j+1} w_{ik}^{(j+1)} \cdot \delta_i^{(j+1)} \right] \cdot y_h^{(j-1)} \dots\dots\dots(8)$$

La mise à jour de l'ensemble des poids d'une couche nécessite la connaissance des erreurs associées à chaque neurone de la couche suivante. On appliquera l'algorithme d'ajustement des poids en partant de la dernière couche (pour laquelle les erreurs sont connues) vers la première. D'où l'appellation de cet algorithme : **algorithme de rétropropagation du gradient de l'erreur**

Nous allons dans ce qui suit donner l'algorithme de rétropropagation du gradient pour un réseau à 2 couches.

1) Présenter un exemple X_l à l'entrée du réseau.

2) Calculer les potentiels des neurones de la couche cachée (couche numéro 2)

$$p_k^{(2)} = \sum_{h \in \text{couche.1}} w_{kh}^{(2)} x_h \quad \dots\dots\dots(9)$$

3) Calculer les sorties des neurones de la couche cachée (couche numéro 2)

$$y_k^{(2)} = f(p_k^{(2)}) \quad \dots\dots\dots(10)$$

4) Calculer les potentiels des neurones de la couche de sortie (couche numéro 3)

$$p_i^{(3)} = \sum_{k \in \text{couche.2}} w_{ik}^{(3)} y_k^{(2)} \quad \dots\dots\dots(11)$$

5) Calculer les sorties des neurones de la couche de sortie (couche numéro 3)

$$y_i^{(3)} = f(p_i^{(3)}) \quad \dots\dots\dots(12)$$

6) Calculer les termes d'erreur sur chaque unité de sortie :

$$\delta_i^{(3)} = (y_i^{(3)} - yd_i) \cdot f'(p_i^{(3)}) \quad \dots\dots\dots(13)$$

7) Calculer les termes d'erreur sur chaque neurone de la couche cachée :

$$\delta_k^{(2)} = f'(p_k^{(2)}) \cdot \sum_{i \in \text{couche.3}} w_{ik}^{(3)} \delta_i^{(3)} \quad \dots\dots\dots(14)$$

8) Calculer les valeurs de réadaptation des poids :

Pour la couche 3 : $\Delta w_{ik}^{(3)} = -\alpha \delta_i^{(3)} y_k^{(2)} \quad \dots\dots\dots(15)$

Pour la couche 2 : $\Delta w_{kh}^{(2)} = -\alpha \delta_k^{(2)} x_h \quad \dots\dots\dots(16)$

9) S'il s'agit d'un Data Adaptive Learning (voir définition plus loin) :

Aller à 11.

10) S'il s'agit d'un Block Adaptive Learning (voir définition plus loin) :

Si $l \neq M$ (M étant le nombre d'exemples d'entraînement) retourner à 1.

Sinon aller à 11.

11) Réadapter les poids : $w_{ij}^{(k)} = w_{ij}^{(k)} + \Delta w_{ij}^{(k)} \quad \dots\dots\dots(17)$

12) Répéter de 1 à 11 autant de fois que nécessaire jusqu'à ce que l'erreur quadratique E_i de chaque exemple X_i devienne acceptable pour tous les exemples (inférieure à un seuil d'erreur fixé) avec :

$$E_i = \frac{1}{2} \sum_{i \in \text{couche}_3} (y_i^{(3)} - yd_i)^2 \dots\dots\dots(18)$$

➤ **Data Adaptive Learning** : en utilisant cette technique, les poids sont réadaptés au passage de chaque exemple. De cette manière, le processus devient plus sensible à chaque exemple individuellement, ce qui le rend donc, facilement influençable par les bruits que peuvent contenir ses entrées durant l'entraînement. Cette technique est utilisée pour un apprentissage en temps réel.

➤ **Block Adaptive Learning** : dans ce cas, on ne réadapte les poids qu'après passage de tous les exemples d'entraînement. La réadaptation est plus «prudente». Le réajustement se fait suivant la moyenne de tous les exemples, ce qui rend la méthode moins sensible aux bruits que peuvent contenir les exemples. La méthode est plus robuste si l'application ne nécessite pas un apprentissage en temps réel.

c) Lissage de la règle d'adaptation :

L'algorithme proposé précédemment est rarement utilisé tel quel en pratique. La vitesse de convergence est assez lente. L'amélioration la plus couramment utilisée consiste à ajouter un terme de filtrage sur les incréments d'adaptation. Ce terme est appelé «momentum» dans la littérature [Hér 94]. La règle d'apprentissage $w_{ij}^{(k)} = w_{ij}^{(k)} + \Delta w_{ij}^{(k)}$ proposée précédemment devient

$$w_{ij}^{(k)}(t) = w_{ij}^{(k)}(t) + \Delta w_{ij}^{(k)}(t) + \gamma(w_{ij}^{(k)}(t) - w_{ij}^{(k)}(t-1)) \dots\dots\dots(19)$$

Avec γ le momentum, en pratique, on le prend inférieur et voisin de 1.

• **Quelques modèles de réseaux à apprentissage supervisé [114]:**

a) Le perceptron :

C'est un des premiers réseaux de neurones, conçu en 1958 par Rosenblatt. Il est linéaire et monocouche. Il est inspiré du système visuel. C'est un réseau de neurones très simple. Les sorties des neurones sont binaires, elles ne peuvent prendre que deux états (-1 et 1 ou 0 et 1). Le procédé d'apprentissage est supervisé, ce qui veut dire que les sorties du-réseau sont comparées avec des sorties désirées.

La règle de modification des poids utilisée est la règle de Widrow-Hoff : si la sortie (celle d'une cellule de décision donc) est égale à la sortie désirée, le poids de la connexion entre ce neurone et le neurone d'association qui lui est connecté n'est pas modifié. Dans le cas contraire le poids est modifié en fonction de l'entrée.

Le réseau est capable de résoudre des opérations logiques simples comme AND ou OR. En 1969,

Papert et Minsky (un des pères des systèmes experts) démontrent les limites du perceptron classique, incapable, par exemple de simuler la fonction ou exclusif (xor).

b) Le perceptron multicouches :

C'est le modèle le plus utilisé. Appelé aussi réseau de neurones à rétropropagation. Il est une amélioration du perceptron comprenant une ou plusieurs couches intermédiaires dites couches cachées. Il utilise, pour modifier ses poids, un algorithme de rétropropagation du gradient, qui est une généralisation de la règle de Widrow-Hoff. Il s'agit toujours de minimiser l'erreur quadratique, ce qui est assez simple quand on utilise une fonction f dérivable (la sigmoïde par exemple). On propage la modification des poids de la couche de sortie jusqu'à la couche d'entrée. Les perceptrons multicouches agissent comme un séparateur non linéaire et peuvent être utilisés pour la classification, le traitement de l'image ou l'aide à la décision. Il est très gros consommateur de temps de calcul sur des problèmes de grande taille.

❖ **Réseaux à apprentissage non supervisé :**

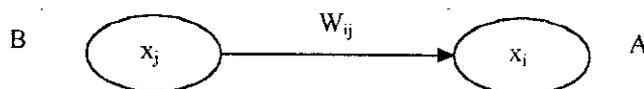
Les réseaux à apprentissage non supervisé traitent des échantillons en entrée sans avoir d'informations sur les sorties que ceux ci doivent générer. Ces réseaux présentent moins de complexité de calculs mais sont moins précis que ceux à apprentissage supervisé.

L'utilisation de ces réseaux est pratique dans les problèmes qui nécessitent un apprentissage rapide en temps réel. Après l'échec du perceptron linéaire et avant l'apparition de l'algorithme de rétropropagation du gradient assurant l'apprentissage supervisé des réseaux multicouches, c'est vers ces réseaux que beaucoup de travaux se sont orientés.

• **La loi de Hebb, le fondement de l'apprentissage non supervisé :**

La loi de Hebb (1949) est le fondement de l'apprentissage non supervisé. Elle s'exprime de la façon suivante :

Si un neurone A est stimulé à maintes reprises par un autre neurone B pendant qu'il est actif. Le neurone A devient plus sensible aux stimulations en provenance de B ; le lien synaptique de B vers A devient plus significatif. Ainsi B pourra stimuler A plus facilement à l'avenir.



Cette règle a été développée sous des aspects mathématiques qui traduisent le principe initial.

Ainsi si w_{ij} est le poids synaptique reliant le neurone B au neurone A, la réadaptation de ce dernier peut s'écrire sous la forme discrète suivante :

$$\Delta w_{ij} = s(x_i) s(x_j)$$

où x_i représente l'activité du neurone et $s(x_i)$ est sa sortie, représentée par une fonction non linéaire appliquée à son activité.

• **Quelques modèles de réseaux à apprentissage non supervisé :**

a) **Réseau de Hopfield :**

Un réseau de Hopfield réalise une mémoire adressable par son contenu. Il est capable de mémoriser de l'information que l'on peut ensuite retrouver non pas par son adresse comme dans une mémoire classique mais en fournissant des données même incomplètes ou bruitées relatives aux informations stockées. Il s'agit d'un réseau constitué de neurones de McCulloch et Pitts (à deux états -1 et 1 ou 0 et 1), dont la loi d'apprentissage est la règle de Hebb, qui veut qu'une synapse améliore son activité si et seulement si l'activité de ses deux neurones est corrélée (C'est à dire que le poids w_{ij} d'une connexion entre un neurone i et un neurone j augmente quand les deux neurones sont activés en même temps).

Ce type de réseau est utilisé dans les problèmes d'optimisation, tels le problème du voyageur de commerce (il s'agit de trouver le plus court chemin reliant N villes en passant une fois et une seule par une ville).

Les limites du réseau de Hopfield :

- Les prototypes doivent vérifier une condition d'orthogonalité assez contraignante et qui n'a pas de sens vis-à-vis d'un problème réel : comment peut-on assurer que des numéros de téléphone sont orthogonaux ?
- Oubli catastrophique : un réseau peut apprendre n exemples sans problème et les oublier tous dès lors que l'on souhaite lui en faire apprendre un de plus.

b) **Le modèle de Kohonen :**

Contrairement aux réseaux de Hopfield où les neurones sont modélisés de la façon la plus simple possible, on recherche ici un modèle de neurone plus proche de la réalité. Ces réseaux sont inspirés des observations biologiques du fonctionnement des systèmes nerveux de perception des mammifères. Une loi de Hebb modifiée (tenant compte de l'oubli) est utilisée pour l'apprentissage. La connexion est renforcée dans le cas où les neurones reliés auraient une activité simultanée, et diminuée dans le cas contraire.

Une loi d'interaction latérale (observée biologiquement) est aussi modélisée. Les neurones très proches (physiquement) interagissent positivement (le poids des connexions est augmenté quand une synapse est activée), négativement pour les neurones un peu plus loin, et pas du tout pour les neurones éloignés. Ceci crée un "amas" de neurones activés et contribue à spécialiser certains neurones : pour une entrée donnée, une sortie particulière sera activée et pas les autres.

On utilise aussi parfois des lois de concurrence entre les neurones (création et destruction de neurones selon certains critères).

Les réseaux de Kohonen permettent de résoudre les problèmes d'optimisation, tels le problème du voyageur de commerce (comment relier n villes par le chemin le plus court). Les réseaux de Kohonen ont des applications dans les problèmes difficiles de la reconnaissance des formes comme la reconnaissance de la parole où les perceptrons multicouches sont limités dans leurs performances.

- **Classification par réseaux de neurones :**

En quoi un réseau de neurones multicouche perceptron peut-il nous aider à résoudre un problème de classification ?

La topologie ou architecture du réseau doit être fixée avant l'apprentissage. Les seules variables pouvant ensuite être modifiées sont les valeurs des poids des connexions. Pour un problème de classification, le nombre de neurones d'entrée est fixé et est égal à la taille du vecteur donné en entrée.

Le nombre de neurones de sortie est aussi connu à l'avance, puisque chaque neurone de sortie est destiné à présenter une classe, donc le nombre de neurones dans la couche de sortie est égal au nombre de classes.

Reste à déterminer quelques paramètres très importants comme le nombre de couches cachées (théoriquement une seule suffit), la taille de la couche cachée, le pas d'apprentissage pour l'algorithme de rétropropagation du gradient, le momentum...

La manière la plus générale pour implémenter un modèle de classification multicouches est d'utiliser un neurone de sortie séparément pour chaque classe. Pendant la phase d'apprentissage, la présentation d'un vecteur d'entrée de la classe connue i , doit activer le neurone de sortie numéro i , c'est à dire que tous les neurones de sortie ont une activation égale 0 sauf le numéro qui représente la classe i doit être à 1 (figure D13).

- **Propriétés et limites des réseaux de neurones [115]:**

- ❖ **Propriétés :**

l'intérêt porté aujourd'hui aux réseaux de neurones tient sa justification dans les quelques propriétés fascinantes qu'il possède :

➤ **la capacité d'adaptation :** Elle se manifeste par la capacité d'apprentissage qui permet de tenir compte de nouvelles contraintes ou de nouvelles données du monde extérieur.

- **Le parallélisme :** Les réseaux de neurones sont considérés comme un ensemble d'entités élémentaires qui travaillent simultanément. Le parallélisme permet une rapidité de calcul supérieure mais exige de poser différemment les problèmes à résoudre
- **La capacité de généralisation :** La capacité de généralisation d'un réseau de neurones est sa capacité à donner une réponse satisfaisante à une entrée qui ne fait pas partie de son ensemble d'apprentissage

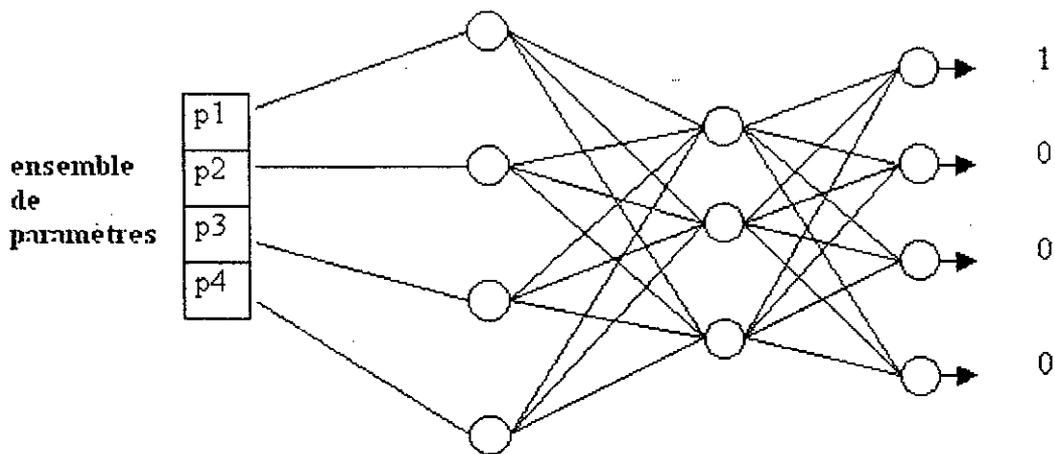


Figure D13 : Un réseau à trois couches pour la classification.

❖ **Limites :**

Les principales limites actuelles sont :

- La plupart des réseaux de neurones sont simulés sur des machines séquentielles. Ce qui entraîne des temps de calculs importants dès que la taille du problème devient grande.
- Incapacité d'expliquer les résultats qu'ils fournissent. Les réseaux de neurones se comportent comme des boîtes noires dont les règles de fonctionnement sont inconnues.

• **Conclusion :**

L'apparition des réseaux de neurones est l'une des révolutions les plus importantes que le monde de l'informatique a connu. Leur utilisation dans les différentes applications a augmenté durant ces dernières années grâce à l'augmentation de la vitesse des processeurs et de la mémoire. A travers ce chapitre, nous avons constaté que les réseaux de neurones sont une famille d'outils très diversifiés. Chaque modèle a ses propres caractéristiques et peut être destiné à des opérations spécifiques.

Règle de modification des poids (méthode de rétro propagation du gradient de l'erreur)

Nous allons dans ce qui suit détailler les calculs qui ont permis d'obtenir $\delta_k^{(j)}$, c'est à dire, la règle (7 de l'annexe D) de modification des poids selon la méthode de rétro propagation du gradient de l'erreur. En reprenant l'équation (2 de l'annexe D), l'erreur quadratique sur la sortie s'écrit : [116]

$$E = \frac{1}{2} \sum_{i=1}^n (y_i - yd_i)^2$$

calculons le gradient de cette erreur par rapport au poids $w_{kh}^{(2)}$ associé à la connexion venant du neurone h de la couche 1 vers le neurone k de la couche 2 (l'indice de la couche cible est indiqué en haut) :

$$\frac{\partial E}{\partial w_{kh}^{(2)}} = \sum_{i \in \text{couche.3}} \frac{\partial E}{\partial p_i^{(3)}} \frac{\partial p_i^{(3)}}{w_{kh}^{(2)}} \dots\dots\dots(1)$$

Calculons d'abord le premier terme :

$$\frac{\partial E}{\partial p_i^{(3)}} = \frac{\partial E}{\partial y_i^{(3)}} \frac{\partial y_i^{(3)}}{\partial p_i^{(3)}}$$

avec :

$$\frac{\partial E}{\partial y_i^{(3)}} = (y_i^{(3)} - yd_i) \dots\dots\dots(2)$$

De la relation $y_i^{(3)} = f\left(\sum_{j \in \text{couche.2}} w_{ij}^{(3)} \cdot y_j^{(2)}\right)$, on obtient :

$$\frac{\partial y_i^{(3)}}{\partial p_i^{(3)}} = f'(p_i^{(3)}) \dots\dots\dots(3)$$

on tire, en tenant compte de l'expression de E :

$$\frac{\partial E}{\partial p_i^{(3)}} = (y_i^{(3)} - yd_i) f'(p_i^{(3)}) \dots\dots\dots(4)$$

Calculons maintenant le second terme de (1). Le potentiel du neurone i de la couche 3 est la somme des sorties $y_j^{(2)}$ de la couche 2 pondérées par les poids $w_{ij}^{(3)}$ des neurones de la couche3. On peut donc écrire :

$$\frac{\partial p_i^{(3)}}{\partial w_{kh}^{(2)}} = \frac{\partial}{\partial w_{kh}^{(2)}} \left[\sum_{j \in \text{couche.2}} w_{ij}^{(3)} \cdot y_j^{(2)} \right] \dots\dots\dots(5)$$

ANNEXE E : Rétro propagation du gradient de l'erreur

Dans le second membre de cette expression, le seul terme de type $y_j^{(2)}$ dépendant du poids $w_{kh}^{(2)}$ est $y_k^{(2)}$, celui obtenu pour $j=k$. la relation (5) devient :

$$\frac{\partial p_i^{(3)}}{\partial w_{kh}^{(2)}} = w_{ik}^{(3)} \frac{\partial}{\partial w_{kh}^{(2)}} [y_k^{(2)}] \dots\dots\dots(6)$$

Dans laquelle on a fait sortir les poids $w_{ik}^{(3)}$.

En remplaçant $y_k^{(2)}$ (sortie du neurone k de la couche 2) par son expression, on arrive à :

$$\frac{\partial p_i^{(3)}}{\partial w_{kh}^{(2)}} = w_{ik}^{(3)} \cdot \frac{\partial}{\partial w_{kh}^{(2)}} \left[f \left(\sum_{m \in \text{couche 2}} w_{km}^{(2)} \cdot y_m^{(1)} \right) \right] \dots\dots\dots(7)$$

Le seul terme de la somme sur m dont la dérivée est non nulle est celui qui contient $w_{kh}^{(2)}$, car il correspond à $m=h$. En calculant la dérivée, on obtient :

$$\frac{\partial p_i^{(3)}}{\partial w_{kh}^{(2)}} = w_{ik}^{(3)} \cdot f'(p_k^{(2)}) \cdot y_h^{(1)} \dots\dots\dots(8)$$

On peut réécrire à partir des relations (4) et (8) le gradient de l'erreur

$$\frac{\partial E}{\partial w_{kh}^{(2)}} = \sum_{i \in \text{couche 3}} (y_i^{(3)} - yd_i) \cdot f'(p_i^{(3)}) \cdot w_{ik}^{(3)} \cdot f'(p_k^{(2)}) \cdot y_h^{(1)} \dots\dots\dots(9)$$

En posant $\delta_i^{(3)} = (y_i^{(3)} - yd_i) \cdot f'(p_i^{(3)})$ et en sortant les termes $y_h^{(1)}$ et $f'(p_k^{(2)})$ de la somme sur i :

$$\frac{\partial E}{\partial w_{kh}^{(2)}} = \left[\sum_{i \in \text{couche 3}} \delta_i^{(3)} \cdot w_{ik}^{(3)} \right] \cdot f'(p_k^{(2)}) \cdot y_h^{(1)} \dots\dots\dots(10)$$

Si on pose :

$$\delta_k^{(2)} = \left[\sum_{i \in \text{couche 3}} \delta_i^{(3)} \cdot w_{ik}^{(3)} \right] \cdot f'(p_k^{(2)}) \dots\dots\dots(11)$$

La relation (11) est l'expression de l'erreur affectée à un neurone k de la couche 2. Elle se généralise pour n'importe quelle couche (j) du réseau de neurones.

$$\delta_k^{(j)} = \left[\sum_{i \in \text{couche } j+1} \delta_i^{(j+1)} \cdot w_{ik}^{(j+1)} \right] \cdot f'(p_k^{(j)}) \dots\dots\dots(12)$$

الخلاصة

يصف هذا البحث نظام التعرف الأتوماتيكي على الحروف العربية. و تتمثل المعضلة الأساسية في القراءة الأتوماتيكية للنصوص العربية في إمكانية تجزئها إلى الحروف المكونة لها نجزيّ أولاً النص إلى أسطر، و نجزيّ كلمات كل سطر حسب حروفها المتصلة، ثم نقوم بتقسيم كلّ من هذه الجزاء إلى حروف. و هذا التقسيم يعتمد على خصائص بنية الكتابة العربية. الطريقة المتبعت للتعرف على الحروف أخذت بعد دراسة عدة تقنيات، فهي طريقة تركيبية بنوية تعتمد على استخراج الخصائص الأساسية - ثقوب و ثقعات في الإتجاهات الأربعة - بغية الترتيب مع الخصائص الثانوية للتمييز بين الحروف العربية و إزالة الإلتباس.

مفاتيح: التعرف على الحروف، الكتابة العربية، الحروف المطبوعة، تجزيء النص، الطريقة التركيبية.

RESUME

Ce travail décrit un système de reconnaissance automatique des caractères arabes imprimés. Le problème majeur de la reconnaissance automatique de l'écriture arabe cursive est la segmentation en ses différents constituants. Tout d'abord, nous segmentons le texte en lignes puis en parties connexes et enfin en caractères. Cette segmentation est basée sur les propriétés contextuelles de l'écriture arabe; notre méthode de segmentation apporte des solutions aux problèmes des caractères sur segmentés.

Après étude de différentes techniques de reconnaissance, nous optons pour une méthode structurale syntaxique basée sur la détection des boucles et des concavités dans les quatre directions (haute, basse, droite et gauche) qui permettent la classification des caractères. D'autres caractéristiques secondaires sont utilisées pour différencier entre les caractères appartenant à la même classe et réduire ainsi les confusions rencontrées pratiquement dans tout système multi fonte.

Mots clés: Reconnaissance de caractères optiques, écriture Arabe, caractères imprimés, segmentation de texte, méthode structurale multi fonte.

ABSTRACT

This work describes an automatic recognition system for Arabic writing. The major problem in automatic reading of Arabic cursive characters is the segmentation of their constituent elements. First, the text is segmented in lines, then in connected parts, and finally in characters. This segmentation is based on the contextual properties of Arabic writing and gives a solution to the problem of over segmentation of some characters.

After studying various recognition techniques, we choose a syntactic structural method based on the detection of holes and concavities in four directions (up, down, left and right), which permits the classification of characters. In addition, secondary characteristics are used in order to differentiate between the characters of each class and to reduce the confusions that are, in practice, meet in every multi font system.

Key words: Optical character recognition, Arabic writing, printed characters, text segmentation, structural method and multi font.