

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université 8 Mai 1945 Guelma
Faculté des Sciences Et De La Technologie
Département D'électronique et Télécommunications



THESE

Présenté pour obtenir le Diplôme de

DOCTORAT EN SCIENCES

Options : Génie Electrique

Par

ZAGHDOUDI RACHID

Les techniques de reconnaissance de Formes

Application : à la reconnaissance de l'écriture Arabe

Soutenu le : 02/05/2018

Devant le Jury composé de :

Président	: Mr Abdelkrim MOUSSAOUI	Professeur	Université de Guelma
Rapporteur	: Mr Hamid SERIDI	Professeur	Université de Guelma
Examineurs	: Mr Nadir FARAH	Professeur	Université d'Annaba
	Mr Tarek KHADIR	Professeur	Université d'Annaba
	Mr Tahar BOUHDADA	Professeur	Université d'Annaba
	Mr Mohamed NEMISSI	M.C.A	Université de Guelma

REMERCIEMENTS

Avant toute chose, je remercie dieu tout puissant de m'avoir aidé et éclairé le chemin pour la réalisation de cette thèse.

En premier lieu, je tiens à exprimer ma reconnaissance et mon profond respect à mon directeur de thèse Monsieur **Hamid SERIDI** professeur à l'université de Guelma de m'avoir proposé ce sujet et diriger constamment de près mon travail. J'apprécie la confiance qu'il m'a témoignée et les conseils avisés qu'il m'a prodigués. Qu'il trouve ici mes sincères reconnaissances.

Je souhaite adresser mes sincères remerciements à **Mr. MOUSSAOUI Abdelkrim** professeur à l'université de Guelma, pour m'avoir fait l'honneur de présider le jury de ma soutenance.

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

J'adresse également mes sincères remerciements au **Mr. KHADIR Tarek** professeur à l'université d'Annaba d'avoir accepté de juger ce travail et d'apporter ses critiques tant constructives.

Je remercie chaleureusement le Professeur **BOUHDADA Tahar**, de l'université d'Annaba, pour avoir bien voulu examiner ce travail et d'apporter ces conseils congruents.

Je tiens également à exprimer mes sincères reconnaissances au **Mr. NEMISSI Mohamed** Maître de conférences à l'université de Guelma, et je le remercie d'avoir accepté de juger ce travail et d'apporter ses remarques aussi constructives et pertinentes.

Je tiens aussi à remercier **Mr. A. Hani Boukrouche** professeur à l'université de Guelma, pour son aide, ses inestimables conseils, ses explications claires et ses orientations précises. Qu'il trouve ici mes sincères reconnaissances.

Je remercie également tous les membres du laboratoire (**LabSTIC**) pour leur soutien moral et leur encouragement pour l'accomplissement de ce travail.

Un remerciement assez spécial à tout le personnel de l'unité de Recherche Appliquée en Sidérurgie et Métallurgie (URASM).

Enfin, je remercie toutes les personnes qui m'ont aidé de près ou de loin.

Dédicaces

Je dédie ce modeste travail à la mémoire de mes parents.

J'aurais tant aimé que vous soyez présents.

Que Dieu ait vos âmes dans sa sainte miséricorde.

A ma femme qui m'a beaucoup soutenu et encouragé.

A mon petit: Anes.

A mes chers frères.

A mes amis.

A toute ma famille.

ملخص

يحظي مجال التعرف الآلي على الكتابة العربية بخط اليد باهتمام بحثي مكثف خلال العقدين الماضيين، وتم اقتراح عدد كبير من التطبيقات العملية. على الرغم من هذه التطور، تبقى المشكلة مفتوحة للبحث والتجريب نظرا للخصائص المورفولوجية المعقدة للغة العربية التي هي في الأصل من أسباب فشل العلاجات. إن البحث المقدم في هذه الأطروحة يندرج ضمن هذا الإطار وهو إسهام في مجال التعرف على الكتابة العربية. أولا، نقترح طريقة إجمالية مبنية على أساس الدمج الموازي لمجموعة من المصنفات قصد التعرف على الكلمات العربية المكتوبة بخط اليد. ويتمحور النظام المتقدم حول أربع مراحل مختلفة : مرحلة المعالجة الأولية، استخراج الخصائص، التصنيف والتي تتم باستخدام نوعين من المصنفات: SVM و F-KNN والمرحلة الأخيرة تتمثل في الدمج الموازي للمصنفات.

لاختبار مدى نجاعة النظر المنجز، أجريت مجموعة من التجارب على قاعدة البيانات IFN/ENIT والتي بينت مدى فعالية النظام المقترح.

في الجزء الثاني، نقدم خوارزمية جديدة لتجزئة الكلمات العربية المكتوبة بخط اليد. هذه الخوارزمية هي نتيجة للتعاون بين مجموعة من القواعد المورفولوجية الخاصة بالكتابة العربية (AMS) ومصنف SVM فئة واحدة، المهمة الرئيسية لهذا الأخير تتمثل في قبول أو رفض نقاط التجزئة الأولية الناتجة عن وحدة (AMS). وأظهر النظام المقترح نتائج تجزئة جيدة على الكلمات المستخرجة من قاعدة بيانات المرجع (IFN/ENIT).

الكلمات الدالة

التعرف على الكتابة العربية بخط اليد، دمج المصنفات، مصنف SVM، مصنف FKNN، الرسم البياني للتدرجات الموجهة، تحويل جيب التمام المتقطع، التقطيع.

Résumé

Le domaine de la reconnaissance de l'écriture arabe manuscrite a reçu un intérêt de recherche intensif aux cours des deux dernières décennies et se concrétise aujourd'hui par un grand nombre d'applications opérationnelles. Malgré ces avancées, le problème reste ouvert à la recherche et à l'expérimentation à cause des caractéristiques morphologiques complexes du manuscrit arabe qui sont à l'origine de la défaillance des traitements. Les travaux de recherches présentés dans cette thèse s'inscrivent dans ce cadre et constituent une contribution au domaine de la reconnaissance de l'écriture arabe manuscrite. Dans un premier temps, nous proposons une approche globale basée sur une combinaison parallèle de classifieurs pour la reconnaissance hors ligne de mots arabe manuscrits. Le système développé s'articule autour de quatre modules distincts : prétraitement, extraction de primitives, classification basée sur deux types de classifieurs : SVM et k plus proches voisins flou et un module de combinaison. Des expériences ont été effectuées sur des images de la base de référence INF/ENIT ont montré l'efficacité du système proposé.

En second temps, nous présentons un nouvel algorithme de segmentation en caractères des mots arabes manuscrits. Cet algorithme est fruit d'une coopération entre un ensemble de règles morphologiques spécifiques à l'écriture arabe (AMS) et un classifieur SVM mono classe, dont la tâche principale consiste à accepter ou rejeter les points de segmentation primaires générées par le module AMS. Il a montré de bons résultats de segmentation sur des mots extraits de la base de données de référence IFN/ENIT.

***Mots clés :** reconnaissance de l'écriture arabe manuscrite, combinaison de classifieurs, séparateurs à vaste marge, k plus proches voisins flou, la transformée en cosinus discrète, histogrammes des gradients orientés, segmentation.*

Abstract

The field of Arabic handwritten recognition has received intense research interest during the last two decades and a large number of operational applications have been proposed. Despite these advances, the problem remains open to research and experimentation due the complex morphological characteristics of the Arabic script which are at the origin of the failure of the treatments. The research presented in this thesis falls within this framework and is a contribution to the field of Arabic handwritten recognition. At a first, we propose a global approach based on a parallel classifiers combination for the offline recognition of Arabic handwritten words. The developed system was divided into four steps: pre-processing, features extraction, classification based on two types of classifiers: SVM and fuzzy k nearest neighbors and a classifiers combination stage. Experiments were carried out on images of the INF / ENIT reference database showing the effectiveness of the proposed system.

In second time, we present a new algorithm for the segmentation of Arabic handwritten words. This algorithm is the result of cooperation between a set of morphological rules specific to Arabic writing (AMS) and one class SVM classifier, which his main task is to accept or reject the primary segmentation points generated by the AMS module. The proposed system showed a good segmentation results on words extracted from the IFN / ENIT reference database.

Keywords: *Handwritten Arabic word recognition; Classifier combination; Support vector machine; Fuzzy K-nearest neighbor; Discrete cosine transform; Histogram of oriented gradients, segmentation.*

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GENERALE

01

CHPAITRE I Généralités sur la Reconnaissance des formes

I.1 Introduction 04

I.2 Architecture d'un système de Reconnaissance des formes..... 04

 I.2.1 Acquisition..... 06

 I.2.2 Prétraitement..... 06

 I.2.3 Extraction des primitives 06

 I.2.4 Classification..... 07

I.3 Approches de reconnaissance de formes..... 08

 I.3.1 Approches structurelles 08

 I.3.2 Approches statistiques 09

I.4 domaines d'application de la reconnaissance des formes..... 09

 IV.1 Reconnaissance des signaux de communication 09

 IV.2 Reconnaissance des signaux d'origine naturelle..... 10

 IV.3 Reconnaissance des signaux d'origine artificielle..... 10

I.5. Conclusion 11

CHPAITRE II Reconnaissance de l'écriture manuscrite : Etat de l'art

II.1 Introduction..... 12

II.2 Différents aspects de la Reconnaissance Automatique de l'Ecriture..... 12

 II.2.1 Mode d'acquisition (en ligne ou hors ligne)..... 12

 II.2.1.1 La reconnaissance en ligne (on line) 12

 II.2.1.2 la Reconnaissance hors ligne (off line)..... 13

 II.2.2 Reconnaissance de l'imprimé ou du manuscrit 13

 II.2.3 Approches de reconnaissance..... 13

 II.2.3.1 Approche globale 13

 II.2.3.2 Approche analytique..... 14

II.3 Architecture d'un système de reconnaissance de l'écriture hors ligne..... 14

 II.3.1 Acquisition 15

 II.3.2 Prétraitement..... 16

II.3.2.1 Filtrage.....	16
II.3.2.2 Binarisation.....	16
II.3.2.3 Lissage	17
II.3.2.4 Extraction des contours	18
II.3.2.5 Extraction des composantes connexes.....	19
II.3.2.6 Squelettisation.....	19
II.3.2.7 Correction de l'inclinaison des caractères.....	20
II.3.2.8 Correction de l'inclinaison des lignes	20
II.3.2.9 Normalisation de la taille	21
II.3.3 Extractions des primitives.....	21
1. Les caractéristiques statistiques	22
2. Les caractéristiques structurelles.....	24
3. Les transformations globales.....	25
II.3.4 Classification.....	27
II.3.5 Post-traitements.....	27
II.3.5.1 Connaissances pragmatiques.....	27
II.3.5.2 Connaissances linguistiques.....	27
II.4 Mesure de performance.....	28
II.5 Domaines d'applications de la reconnaissance de l'écriture.....	28
II.6 Conclusion.....	28

CHAPITRE III Reconnaissance de l'écriture arabe manuscrite

III.1 Introduction.....	29
III.3 Caractéristiques de l'écriture arabe.....	29
III.2 Présentation de la langue arabe.....	29
III.3.1 Voyelles	31
III.3.2 Signes diacritiques.....	32
III.3.3 Ascendants et descendants.....	32
III.3.4 Ligatures et chevauchement verticaux.....	32
III.4 Principales bases de données existantes.....	33
III.4.1 La base Al-Isra 1999.....	33
III.4.2 La base CENPARMI '1', 2000.....	34
III.4.3 La base IFN/ENIT, 2002.....	34
III.4.4 La base CEDARABIC, 2005	35
III.4.5 La base KHATT, 2012.....	36
III.4.6 La Base AHTID /DW, 2012.....	36

III.5	Surveys existants.....	36
III.5.1	A. Amin, 1998.....	36
III.5.2	M. S. Khorsheed, 2002.....	37
III.5.3	N. Ben Amara et F. Bouslama, 2003.....	37
III.5.4	L. M. Lorigo et V. Govindaraju, 2006.....	37
III.5.6	M. T. PARVEZ et S.A. Mahmoud, 2013.....	37
III.6	Différentes approches et systèmes existants.....	38
III.6.1	Prétraitements.....	39
III.6.1.1	Squelettisation.....	39
III.6.1.2	L'estimation de la ligne de base.....	39
III.6.1.3	Correction de l'inclinaison (slant correction).....	40
III.6.1.4	Normalisation de la taille des mots.....	42
III.6.1.5	Segmentation des mots en caractères.....	42
III.6.2	Extraction de primitives.....	46
III.6.3	Reconnaissance des mots.....	47
III.6.3.1	Approches Globales.....	48
III.6.3.1.1	Par modèles de Markov cachés (HMM).....	48
III.6.3.1.2	Par réseaux de neurones (RNN).....	49
III.6.3.1.3	Par les (SVMs).....	51
III. 7	Compétitions sur la reconnaissance de l'écriture arabe manuscrite.....	52
III.7.1	Compétition ICDAR2005.....	52
III.7.2	Compétition ICDAR 2007	52
III.7.3	Compétition ICDAR 2009.....	54
III.7.4	Compétition ICFHR 2010.....	55
III.7.5	Compétition ICDAR 2011.....	56
III.8	Conclusion	57
CHAPITRE IV Classification		
IV.1	Introduction.....	58
IV.2	Apprentissage.....	58
IV.3	Méthodes de classification	59
IV.4	La méthode des K plus proches voisins flous.....	59
IV.5	Machines à vecteurs supports (SVM).....	61
IV.5.1	Principe	61
IV.5.2	Notions de base : Hyperplan, marge et support vecteur.....	62
IV.5.3	Les SVMs linéaires.....	62
IV.5.4	Les SVMs non linéaires.....	64
IV.5.5	Les fonctions noyaux.....	65
IV.5.6	Les hyper-paramètres.....	66

IV.5.7 SVM Multi-classes.....	67
IV.5.7.1 L'approche Un Contre Tous.....	67
IV.5.7.2 L'approche Un Contre Un.....	68
IV.5.7.3 DAG-SVM.....	68
IV.6 Conclusion.....	69

CHAPITRE V Combinaison de classifieurs

V.1 Introduction.....	70
V.2 Pourquoi combiner plusieurs classifieurs?.....	70
V.3 Types de sorties d'un classifieur.....	71
V.3.1 Type classe.....	71
V.3.2 Type rang.....	72
V.3.3 Type mesure.....	72
V.4 Les méthodes de combinaison des classifieurs.....	72
V.4.1 Approche séquentielle.....	72
V.4.2 Approche parallèle.....	73
V.4.3 Approche hybride.....	74
V.5 Catégorisation des méthodes de combinaison parallèle.....	75
V.6 Combinaison sans apprentissage.....	76
V.6.1 Type classe.....	76
V.6.2 Type rang.....	78
V.6.3 Type Mesure.....	80
V.7 Les Méthodes de combinaison avec apprentissage.....	81
V.7.1 Type classe.....	81
V.7.2 Type rang.....	83
V.7.3 Type mesure.....	84
V.8 Conclusion.....	85

CHAPITRE VI Contribution à la reconnaissance de l'écriture arabe manuscrite

VI.1 Introduction.....	86
VI.2 Prétraitement.....	86
VI.1.1 Correction de l'inclinaison de ligne (Slope correction).....	88
VI.1.2 Correction de l'inclinaison des caractères (Slant correction).....	90
VI.2.3 Normalisation de la taille.....	90
VI.2 .4 Squelettisation.....	91
VI.3 Extraction de primitives.....	92
VI.3.1 Extraction des caractéristiques par la transformée de DCT.....	92

VI.3.2 Extraction des caractéristiques par PCA-HOG.....	93
VI.4 Classification.....	94
VI.4.1 K plus proche voisin flou.....	94
VI.4.2 SVM multi-classes.....	94
VI.4.3 Règles de combinaison parallèle.....	95
VI.5 Résultats expérimentaux.....	96
VI.5.1 Performances des classificateurs individuels.....	97
VI.5.1.1 Le premier classifieur	97
VI.5.1.2 Le deuxième classifieur.....	98
VI.5.1.3 Le Troisième classifieur.....	101
VI.5.1.4 Le quatrième classifieur	103
VI.5.2 Performances de l'ensemble des classifieurs.....	107
VI.5.3 Comparaison de nos résultats avec d'autres systèmes.....	108
VI. 6 Conclusion.....	110

**CHAPITRE VII Nouvel Algorithme de segmentation de l'écriture arabe
manuscrite basé sur une approche de Validation SVM**

VII.1 Introduction.....	111
VII.2 Architecture de l'algorithme proposé.....	111
VII.2.1 Prétraitement.....	112
VII.2.2 Détection de la ligne de base.....	113
VII.2.3 Filtrage des signes diacritiques.....	114
VII.2.4 Détermination des points de segmentation primaires.....	115
VII.2.5 Arabic Morphological Segmenater (AMS).....	116
VII.2.6 Validation par SVM.....	117
VII.3 Résultats expérimentaux.....	118
VII.4 Conclusion.....	121
CONCLUSION GENERALE.....	122

LISTE DES FIGURES

<u>FIGURE</u>	<u>TITRE</u>	<u>PAGE</u>
Figure I.1	Architecture d'un système de reconnaissance des formes.....	5
Figure I.2	Les approches statistiques.....	9
Figure II.1	Architecture d'un système de reconnaissance d'écriture.....	15
Figure II.2	Image binarisée par la méthode d'Otsu.....	17
Figure II.3	Extraction du contour.....	18
Figure II.4	Résultat de squelettisation par l'algorithme de Zhang Suen.....	20
Figure II.5	Exemple de la technique de Zoning.....	22
Figure III.1	Exemple de pseudo mots constituant un mot arabe.....	31
Figure III.2	Exemple de formes de boucles dans des styles différents.....	31
Figure III.3	Mots arabes incluant la Hamza, la Chadda et la Madda.....	32
Figure III.4	Exemple de chèque saoudien de la base CENPARMI '1'	34
Figure III.5	Le concept de slant correction	41
Figure III.6	Résultats de la segmentation	44
Figure III.7	Extractions de caractéristique avec une fenêtre glissante	48
Figure IV.1	Représentation des notions de base du SVM.....	62
Figure IV.2	Changement d'espace.....	64
Figure IV.3	Exemple de classification SVM 3-classes.....	67
Figure IV.4	Architecture d'un système de classification basée sur la méthode Un-Contre-Tous	68
Figure IV.5	Une classification de trois classes par DAG-SVM.....	69
Figure V.1	Combinaison séquentielle de classifieurs.....	73
Figure V.2	Principe de la combinaison parallèle de classifieurs.....	74
Figure V.3	Combinaison hybride de classifieurs.....	75
Figure V.4	Classification des méthodes de combinaison parallèles.....	77
Figure VI.1	Architecture du système développé.....	87
Figure VI.2	Processus de prétraitement.....	88
Figure VI.3	Résultat du slope correction.....	89
Figure VI.4	La lecture Zigzag de la matrice de DCT.....	93
Figure VI.5	Taux de reconnaissance du FKNN1 en Fonction du k.....	98
Figure VI.6	Variation de taux de reconnaissance du FKNN2 en fonction du k.....	101
Figure VI.7	Les taux de reconnaissance maximums des deux classifieurs.....	106
Figure VI.8	Les taux de reconnaissance de l'ensemble des classifieurs.....	107

Figure VII.1	Architecture de l'algorithme proposé.....	112
Figure VII.2	Processus de prétraitement.....	112
Figure VII.3	Détection de la ligne de base.....	113
Figure VII.4	Procédure de filtrage des signes diacritiques.....	114
Figure VII.5	Exemples de la procédure de filtrage des signes diacritiques.....	115
Figure VII.6	Exemple de détermination des points PSP.....	115
Figure VII.7	Procédure d'application de L'AMS.....	116
Figure VII.8	Résultat de l'application de l'AMS.....	117
Figure VII.9	Exemple de détermination des régions de segmentation d'un mot..... arabe manuscrit	117
Figure VII.10	Exemples de l'algorithme de segmentation proposé.....	120

LISTE DES TABLEAUX

<u>TABLEAU</u>	<u>TITRE</u>	<u>PAGE</u>
Tableau I.1	Exemple des applications de la reconnaissance de formes.....	11
Tableau III.1	Les 28 lettres arabes avec leurs différentes formes d'apparitions dans un mot.....	30
Tableau III.2	Lettres arabes ayant des points diacritiques.....	32
Tableau III.3	Deux images et leurs annotations respectives dans la base IFN/ENIT...	35
Tableau III.4	Primitives utilisées dans les systèmes de reconnaissance de mots arabes manuscrits.....	47
Tableau III.5	Résultats de reconnaissance des systèmes participants à la compétition ICDAR 2005.....	52
Tableau III.6	Résultats de reconnaissance des systèmes participants à la compétition ICDAR 2007.....	53
Tableau III.7	Résultats de reconnaissance des systèmes participants à la compétition ICDAR 2009.....	54
Tableau III.8	Résultats de reconnaissance des systèmes participants à la compétition ICFHR 2010.....	56
Tableau III.9	Résultats de reconnaissance des systèmes participants à la compétition ICDAR 2011.....	57
Tableau IV.1	Les fonctions noyau les plus courantes.....	66
Tableau VI.1	Squelettes des images de mots arabes par l'algorithme de Zhang-Suen..	92
Tableau VI.2	Quelques images de mots arabes manuscrits extraites de la base IFN/ENIT.....	96
Tableau VI.3	Taux de reconnaissance en u FKNN1 Fonction du nombre de coefficients de DCT.....	97
Tableau VI.4	variation du taux de reconnaissance du FKNN2 en fonction du nombre de Cellules N.....	99
Tableau VI.5	variation du taux de reconnaissance du FKNN2 en fonction du nombre de bins (B).....	99

Tableau VI.6	variation du taux de reconnaissance du FKNN1 en fonction du % de PCA.....	100
Tableau VI.7	Taux de reconnaissance du classifieur SVM1 en fonction du nombre de coefficients de DCT.....	102
Tableau VI.8	Exemple de détermination des valeurs de C et γ pour le classifieur SVM1.....	102
Tableau VI.9	Taux de reconnaissance du classifieur SVM2 en fonction du nombre de cellules (N).....	103
Tableau VI.10	Taux de reconnaissance du classifieur SVM2 en fonction du nombre de bins (B).....	104
Tableau VI.11	Taux de reconnaissance du classifieur SVM2 en fonction du % de coefficients de PCA-HOG.....	104
Tableau VI.12	Exemple de détermination des valeurs de C et γ pour le classifieur SVM2.....	105
Tableau VI.13	Comparaison du temps d'exécution des classifieurs.....	106
Tableau VI.14	Comparaison des systèmes de reconnaissance de mots arabes manuscrits.....	108
Tableau VI.15	Exemples de l'application du système proposé sur des mots arabes manuscrits.....	109
Tableau VII.1	Résultats de la segmentation.....	118

INTRODUCTION GENERALE

INTRODUCTION GENERALE

La reconnaissance de formes est un domaine très vaste dans lequel les recherches sont encore très actives. En effet, il existe un très grand nombre d'applications qui peuvent nécessiter un module de reconnaissance notamment dans les systèmes de traitement visant à automatiser certaines tâches fastidieuses effectuées par l'être humain. Parmi celles-ci ; on trouve la reconnaissance de l'écriture manuscrite, qui est un problème délicat à traiter, car il regroupe, à lui seul, une grande partie des difficultés généralement rencontrées dans le domaine de la reconnaissance des formes.

Depuis plusieurs décennies, les méthodes de communication ont beaucoup évolué surtout avec l'apparition des ordinateurs. Malgré le grand développement dans ce domaine, et son influence dans notre vie quotidienne, l'écriture reste toujours un moyen essentiel de communication et d'échange d'informations et d'idées. À l'ère des ordinateurs, l'intérêt que l'on a pour l'écriture n'a pas beaucoup changé, au contraire, on cherche à la comprendre et à mieux l'exploiter avec ces mêmes ordinateurs.

L'objectif de tout système de reconnaissance de l'écriture, est de transformer le texte écrit ou imprimé en une représentation compréhensible par une machine et facilement manipulable par un logiciel de traitement de texte. C'est-à-dire, de doter la machine de la capacité de lire l'écriture cursive spontanée. Toutefois, la compréhension de l'écriture par une machine est encore loin d'être pleinement satisfaisante. En effet, si la lecture est l'une des tâches courantes et naturelles pour l'homme qui parvient à lire facilement un texte écrit parfois très dégradé, elle représente pour la machine un défi très complexe. Par ailleurs, la reconnaissance de l'écriture manuscrite a donné lieu au cours des dix dernières années à une activité de recherche très importante, car elle contribue beaucoup à l'accomplissement des tâches administratives d'une manière facile et à résoudre de nombreux problèmes de traitement de l'écrit, ce qui conduit à réduire le temps et l'effort pour accomplir ces tâches fastidieuses. Les avancées réalisées dans ce domaine se concrétisent aujourd'hui par un certain nombre d'applications opérationnelles que ce soit dans le domaine de la reconnaissance hors-ligne ou de la reconnaissance en-ligne. En reconnaissance hors-ligne, des applications pour le tri automatique du courrier, pour la lecture des chèques bancaires ou de traitement de formulaires sont déjà opérationnelles. Dans le cadre de la reconnaissance en-ligne, des applications comme les stylos électroniques, les agendas numériques et les Personal Digital Assistants ont fait également leur apparition. Néanmoins, il existe encore beaucoup d'obstacles à surmonter et de nombreux problèmes à résoudre avant d'arriver à construire une machine capable de simuler parfaitement le processus de la lecture chez l'être humain, surtout dans le cas de la reconnaissance hors ligne d'écriture arabe

manuscrite. En effet l'écriture arabe manuscrite présente des défis et des problèmes très complexes, et la majorité des techniques utilisées pour la reconnaissance de l'écriture latine ou chinoise ne peuvent pas être facilement appliquées sur l'écriture arabe et cela est dû principalement à la complexité morphologique de l'écriture arabe et aux grandes variations des styles d'écriture. Ce qui explique ce retard par rapport à d'autres scripts dans le domaine de la reconnaissance de l'écriture arabe manuscrite.

Les travaux de recherche que nous avons menés dans cette thèse s'intéressent au développement d'un système de reconnaissance de mots arabes manuscrits et la proposition d'un nouvel algorithme de segmentation de l'écriture arabe manuscrite.

En premier lieu, nous proposons une approche globale basée sur la combinaison parallèle des classifieurs pour la reconnaissance hors ligne de mots arabes manuscrits. En effet, l'exploitation des différentes informations fournies par l'ensemble de classifieurs en utilisant les méthodes de combinaison de classifieurs peut conduire à un résultat de classification plus précis. Même si le classifieur de base est moins performant, la connaissance de son comportement peut permettre d'apporter une certaine information complémentaire à propos de la vraie classe de la forme à reconnaître. Ainsi, la combinaison des classifieurs permet généralement d'en cumuler les avantages sans en cumuler les inconvénients.

Dans un second lieu, nous proposons un nouvel algorithme de segmentation de mots arabes manuscrits. La technique utilisée se divise en deux phases principales; la première, est un Arabic Morphological Segmenater (AMS) basé sur des règles morphologiques, est appliquée sur l'image du mot afin d'obtenir une liste de points de segmentations probables (PSP). En conséquence, la deuxième phase est une méthode de validation par un SVM mono classe qui a pour but d'examiner tous les (PSP) pour pouvoir d'enlever celui qui est invalide et garder comme point de segmentation correct, celui qui est valable. Des expériences sont menées sur des images extraites de la base de données IFN/ENIT, pour valider l'approche de segmentation proposée.

Ce présent mémoire est composé de deux grandes parties. La première présente l'état de l'art ainsi que les concepts et notions requises pour la compréhension des systèmes développés. Cette partie est divisée en cinq chapitres. La deuxième partie est constituée de deux chapitres qui présentent l'ensemble de nos contributions dans le domaine de la reconnaissance de l'écriture arabe manuscrite.

Chapitre I : dans ce chapitre nous présentons des généralités sur le domaine de la reconnaissance des formes, ainsi que les différentes étapes qui composent la chaîne de reconnaissance des formes, nous présentons aussi quelques exemples de ces applications.

Chapitre II : ce chapitre présente un état de l'art dans le domaine de la reconnaissance de l'écriture manuscrite, ainsi que les différentes étapes qui le compose. A savoir : acquisition, prétraitement, extractions des primitives, classification et post-traitement.

Chapitre III : il s'intéresse plus particulièrement, à l'écriture Arabe manuscrite et aux travaux effectués dans le domaine de la reconnaissance automatique de l'écriture arabe manuscrite. En effet, nous y présentons tout d'abord un aperçu général de l'écriture arabe manuscrite, en s'intéressant à ces caractéristiques morphologiques. Par la suite nous décrivons les différents travaux antérieurement effectués dans le domaine de la reconnaissance automatique de l'écriture arabe manuscrite, nous exposons aussi les principales bases de données de mots arabes manuscrits existantes.

Chapitre IV : ce chapitre traite une phase essentielle et cruciale dans un système de reconnaissance: la classification, en mettant l'accent sur deux types spécifiques de classifieurs : le k plus proches voisins flou (F-KNN) et les séparateurs à vaste marge (SVM).

Chapitre V : ce chapitre introduit une étude détaillée des systèmes multi-classifieurs et les différentes stratégies de combinaisons, en se concentrant plus particulièrement sur la combinaison parallèle des classifieurs.

Chapitre VI : ce chapitre présente notre première contribution, qui est une approche globale pour la reconnaissance hors ligne de mots arabes manuscrits, basée sur la combinaison parallèle de deux types de classifieurs, k plus proches voisins flou et les séparateurs à vaste marge (SVM).

Chapitre VII : ce chapitre décrit, notre deuxième contribution qui est un nouvel algorithme de segmentation de mots arabes manuscrit, basée sur une approche de validation SVM mono classe.

Nous terminons par une conclusion et quelques perspectives et suggestions sur les travaux futurs dans le domaine de la reconnaissance de l'écriture arabe manuscrite.

CHAPITRE I

**Généralités sur la Reconnaissance
des formes**

Chapitre I Généralités sur la Reconnaissance des formes

I.1 Introduction

La reconnaissance des formes (RDF) est la première étape d'un long processus de compréhension de notre univers dans le cadre général de la communication homme-machine [1]. Cette discipline peut être définie, comme étant l'ensemble des techniques et algorithmes informatiques de représentation et de décision permettant à reproduire la perception humaine. Elle consiste de telle sorte, à doter la machine d'organes de sens (Capteurs physiques tels que: caméra, scanner, microphone) qui captent l'information extérieure sous forme varié.

Parmi les systèmes de reconnaissance de formes existants, l'homme est sans nul doute, le plus parfait. Sa grande capacité à accomplir des tâches de reconnaissance très diverses sur des formes à grande variabilité, est très impressionnante. Actuellement, l'énorme développement dans le domaine scientifique et technique a permis d'essayer d'imiter certaines de ces capacités à l'aide des machines. Sans aller jusqu'au vieux rêve de remplacer complètement l'homme par une machine [2].

La reconnaissance des formes a pour objectif d'une part de décrire ces informations captées du monde extérieur (codage), et d'autre part, de prendre sur la représentation ainsi obtenue, une décision d'indentification par référence à un ensemble d'apprentissage (classification). La reconnaissance de formes a été considérée comme une discipline du domaine de l'intelligence artificielle, dont le but ultime, est de produire des machines qui possèdent la capacité d'apprendre, de penser, de décider, mais aussi d'interagir avec le monde réel, et de reconnaître les objets et les différentes formes.

I.2 Architecture d'un système de Reconnaissance des formes

La reconnaissance des formes a pour objectif d'une part de décrire ces informations: l'apprentissage à partir de données connues et d'autre part, la classification de nouvelles données. En amont de ces deux phases, une étape de prétraitement est utilisée pour réduire la quantité d'informations à traiter et une étape d'extraction de primitives qui consiste à obtenir l'information la plus discriminante à fournir au système de classification. L'ensemble des étapes nécessaires que comporte un système de reconnaissance des formes est présenté sur la Figure I.1.

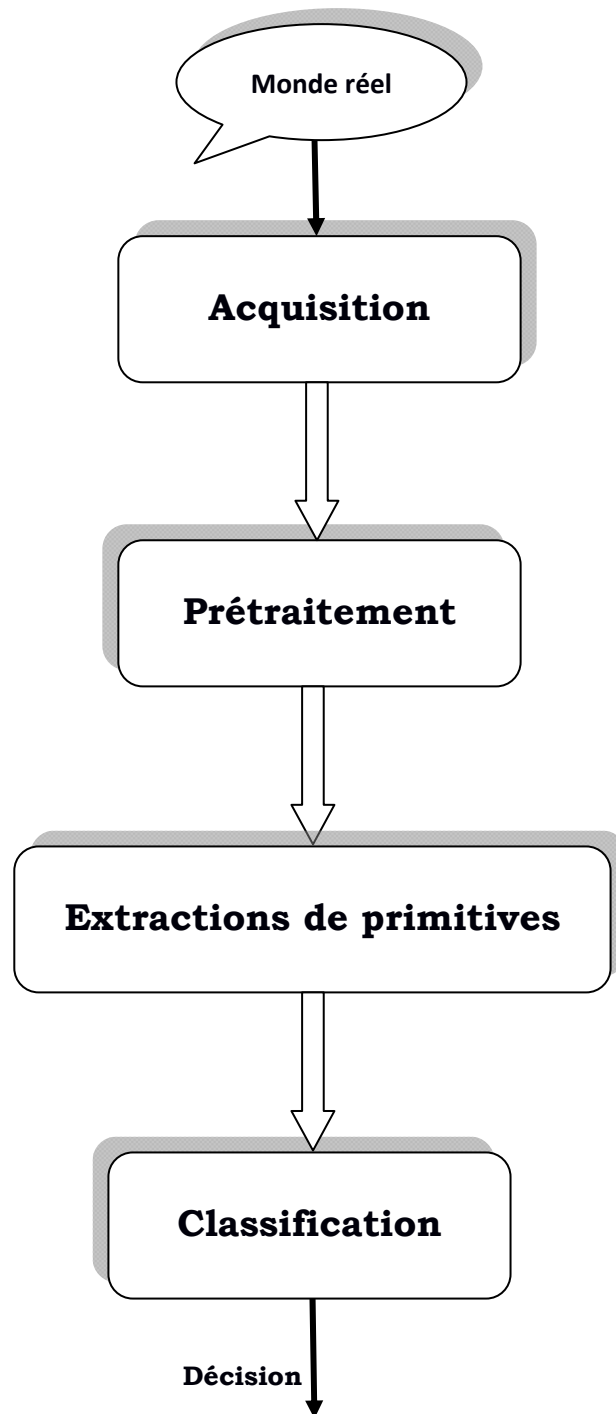


Figure I.1 : Architecture d'un système de reconnaissance des formes.

Rappelons, en premier lieu, le rôle de chaque étape dans l'élaboration du processus complet de reconnaissance des formes :

I.2.1 Acquisition

C'est la première étape dans un processus de reconnaissance de forme, elle permet de transformer l'objet à reconnaître d'une forme réelle ou dite physique en une forme numérique compréhensible par les ordinateurs. Cela peut être réalisé par une caméra ou un scanner dans une tâche de vision ou un microphone pour la parole ou le son. Les données doivent être collectées et acquises avec le plus grand soin et tout bruit ou biais dans les mesures effectuées à ce niveau, est dommageable pour le système final.

Il est également important que la procédure d'acquisition soit contrôlée et fiable, de façon à garantir que toutes les données soient collectées de façon identique et qu'il n'existe pas de dérive. Une hypothèse importante faite par les méthodes de mise au point des systèmes de RDF réside dans le caractère indépendamment et identiquement distribué des données. Les données sont supposées être indépendantes et produites par un même processus sous-jacent.

I.2.2 Prétraitement

Le prétraitement est une suite d'opérations effectuées sur la forme à reconnaître, qui consiste à sélectionner dans l'espace de représentation l'information nécessaire à l'application. Cette sélection passe souvent par l'élimination du bruit introduit lors la phase d'acquisition, par la normalisation des données, aussi bien par la suppression de l'information redondante. Des techniques inspirées du traitement du signal, ont été souvent utilisées pour le prétraitement dans le cadre de la reconnaissance de forme comme par exemple le filtrage, l'extraction de contours, la segmentation, la squelettisation, et la modélisation du signal temporel ...etc.

I.2.3 Extraction des primitives

L'extraction des primitives appelées aussi caractéristiques, est une étape clé dans le processus de la reconnaissance des formes. Elle consiste à représenter les données d'entrées (mots, caractères, phonème, graphème...etc.) sous une forme d'un vecteur de primitives de dimension fixe, pour pouvoir reconnaître les formes où chaque vecteur se comporte comme une signature d'une forme. C'est une étape cruciale et critique dans tout système de reconnaissance de formes. Le but de cette étape est la sélection de l'information pertinente, discriminante et la moins volumineuse possible afin de limiter les ressources machines nécessaires à l'étape de la classification, tout en évitant de perdre des informations importantes et significatives.

La plupart des techniques d'extraction de primitives dans le domaine de la reconnaissance de formes, s'accompagnent généralement d'une perte d'informations irrémédiable. Il est donc nécessaire d'effectuer un compromis entre quantité et qualité de l'information, malheureusement peu d'études théoriques sont faites sur ce sujet, où l'intuition

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

I.2.4 Classification

L'étape de classification joue un rôle primordial dans le processus complet d'un processus de reconnaissance de formes, en désignant l'appartenance d'une forme à une classe. En effet, la classification consiste à élaborer une règle de décision qui va transformer les attributs caractérisant les formes en fonction d'appartenance, c'est à dire un passage de l'espace de codage, vers l'espace de décision [3]. Avant qu'un modèle de décision ne soit intégré dans un système de reconnaissance de formes, il faut avoir procédé auparavant à deux étapes: l'étape d'apprentissage et l'étape de test.

L'apprentissage est une étape clé dans le système de reconnaissance. Il consiste à fournir au système un ensemble de formes connues à priori (on connaît la classe de chacune d'elles). C'est cet ensemble d'apprentissage qui va permettre de régler le système de reconnaissance de façon à ce qu'il soit capable de reconnaître ultérieurement des formes de classes inconnues. On distingue généralement deux types d'apprentissage: apprentissage supervisé et apprentissage non supervisé.

L'apprentissage supervisé, consiste à fournir au module d'apprentissage un échantillon représentatif de l'ensemble des formes à reconnaître. Où l'opérateur de supervision ou professeur, indique l'étiquette correcte de chaque exemple, qui sera utilisée par le module d'apprentissage pour identifier la classe dans laquelle l'opérateur de supervision, souhaite que l'exemple soit arrangé. Donc la phase d'apprentissage, a pour objectif d'analyser les ressemblances entre les formes d'une même famille et les dissemblances entre les formes de familles différentes pour en déduire les meilleures séparations de l'espace de représentations. Alors l'objectif général des méthodes d'apprentissage supervisé, est de construire ou approximer à partir de la base d'apprentissage, une règle ou une fonction de classification qui permet à partir de la description d'une forme, d'affecter la bonne étiquette ou classe à cette forme inconnue par le module d'apprentissage.

Au contraire, dans le cadre de l'apprentissage non supervisé, on ne dispose d'aucune information de supervision, donc, on fournit au système de reconnaissance un grand nombre de formes non étiquetées. Et, c'est l'étape de la classification qui va se charger de regrouper automatiquement les formes jugées similaires (homogènes) dans la même partition de l'espace de

représentation et placer les formes jugées dissimilaires (hétérogènes) dans des partitions distinctes. Ce type d'apprentissage, évite l'assistance d'un opérateur de supervision, mais il n'assure pas toujours, une classification correcte.

L'étape de test permet l'évaluation de la performance du classifieur pour un apprentissage donné. C'est une étape très importante, car elle peut conditionner le choix des méthodes d'extraction de primitives ou le choix de la procédure d'apprentissage. En effet, la détermination a priori de l'ensemble des caractéristiques les plus discriminatives et la méthode d'apprentissage la plus adaptée pour un problème de classification donné, est une tâche très ardue, d'où l'utilité d'une procédure de test ou de validation qui procède par itérations successives. Ces itérations consistent à extraire des caractéristiques les plus discriminatives au problème de reconnaissance à résoudre et à tester la fiabilité du système avec cet ensemble de caractéristiques. Ces tests sont itérés tant qu'on n'a pas atteint les performances désirées. Ces itérations ont pour objectif, de régler quelques paramètres ou de tester une nouvelle famille de primitives, ou de combiner les primitives extraites avec d'autres types de primitives afin de perfectionner le module de classification.

Comme nous avons déjà mentionné ci-dessus, ces deux modules d'apprentissage et de test sont réalisés avant l'intégration du module de décision dans le système de reconnaissance de formes.

I.3 Les approches de reconnaissance de formes

La reconnaissance de formes, peut être définie comme étant un ensemble d'algorithmes permettant de classer des formes en les comparant à des formes-types. Son spectre d'application est très large, tels que la reconnaissance de la parole, la reconnaissance de l'écriture, l'automatisation industrielle, le diagnostic médical, le traitement de l'écrit, etc. Généralement on peut distinguer deux grands types de reconnaissance des formes [4]:

I.3.1 Approches structurelles

Ces approches consistent à utiliser des hypothèses sur les distributions des données à l'intérieur des classes. Donc dans ce cas, les procédures de classification, sont construites à l'aide d'hypothèses probabilistes (par exemple le classifieur Bayésien). Aussi, Elles peuvent être basées sur une représentation des formes à l'aide de grammaires et nécessitent des moyens de calculs importants. Généralement, les approches structurelles ont pour objectif la description de formes complexes à partir de formes élémentaires. Ces dernières, encore appelées caractéristiques ou primitives, sont extraites directement des formes présentes en entrée du système. De ce fait, ce type d'approche n'est pas approprié aux applications en temps réel [5].

I.3.2 Approches statistiques

Contrairement aux approches structurales, les approches de reconnaissance de formes statistiques concernent l'utilisation des techniques statistiques ou numériques pour analyser les mesures des formes à reconnaître afin d'extraire des informations pertinentes et prendre des décisions justifiées, en se basant sur des méthodes paramétriques ou non paramétriques. Les approches statistiques ont beaucoup bénéficié des méthodes d'apprentissage automatique qui s'appuient sur des bases théoriques très connues telles que la théorie de la décision bayésienne, les méthodes de classification non supervisées et l'analyse en composantes principales.

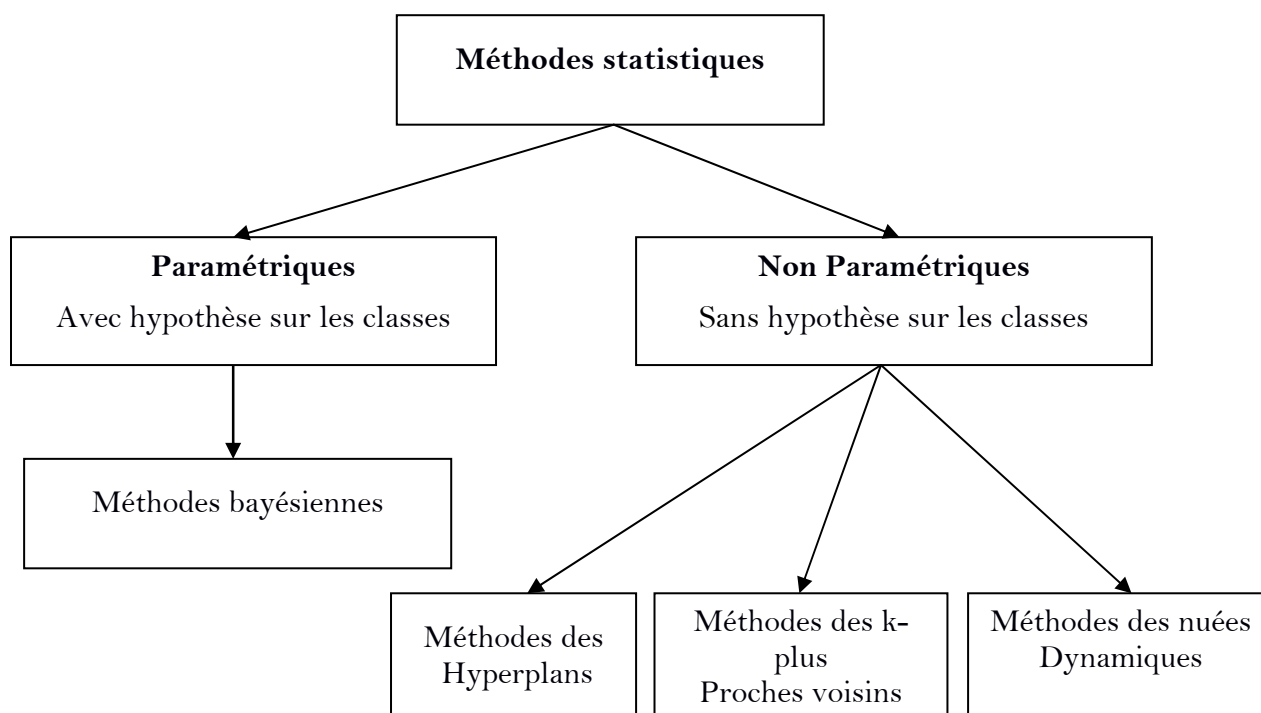


Figure I.2 : Les approches statistiques.

I.4 Domaines d'application de la reconnaissance de formes

La diversité des organes de sens dont dispose la reconnaissance de formes donne une idée sur ses domaines d'application, nous allons citer quelques-uns des plus importants:

I.4.1 Reconnaissance des signaux de communication

La reconnaissance automatique de la parole, est un domaine qui a captivé le public ainsi qu'un grand nombre de chercheurs. Ceci est dû en particulier aux grandes puissances de traitements et de calculs qu'offrent les ordinateurs actuels. Les avancées réalisées dans le domaine de la reconnaissance de la parole ouvrent des perspectives nouvelles grâce aux avantages que présente la commande vocale par rapport à la commande manuelle. L'utilisation

du langage naturel pour la communication entre l'homme et la machine, met la technologie à la portée de tous et entraîne sa diffusion, ce qui conduit à une réduction considérable des contraintes de l'usage des claviers, souris et codes de commandes à maîtriser.

Un autre signal émis par l'homme est l'écriture, dont sa nature imprimée ou manuscrite, et c'est l'un des champs d'applications favoris de la reconnaissance des formes, où se mélangent les méthodes de type statistique et structurelle. Actuellement, de nombreux systèmes de reconnaissance de caractères imprimés, fonctionnent avec d'excellentes performances de même que des systèmes de lecture automatique de chèques ou de tri postal.

I.4.2 Reconnaissance des signaux d'origine naturelle

Un autre domaine extrêmement important, est celui des signaux biomédicaux, où des études récentes ont trouvé des applications intelligentes de la reconnaissance de formes qui peuvent être utilisés pour résoudre divers problèmes de la vie réelle. En effet, L'intégration des systèmes de reconnaissance des signaux biomédicaux est devenue un sujet d'un intérêt croissant pour les chercheurs et les développeurs que se soit dans le domaine académique ou industriel. Parmi ces applications, on peut citer, le diagnostic médical, l'échographie ultrasonore, Surveillance en temps réel dans le cadre de soins intensifs, où les données du patient doivent être analysées en temps réel, Les signaux d'électromyographie (EMG) qui peuvent être utilisés pour des applications telles que le contrôle prothétique des mains, à la reconnaissance de la saisie et à l'interaction homme-ordinateur,...etc.

D'autres signaux d'origine naturelle comme : les signaux sismologiques, les images satellites, les images radar, etc. ont des applications évidentes : détection des ressources, météorologie, prévision des phénomènes naturels, ..., etc.

I.4.3 Reconnaissance des signaux d'origine artificielle

Les signaux sont définis ici comme ceux provenant de machines ou des appareils complexes dont une surveillance automatique peut s'avérer nécessaire. L'insertion automatique du contrôle de la qualité d'un produit à la fin de sa fabrication relève du même domaine. Un exemple très étudié, est celui de la vision par un robot (la vision artificielle), aux multiples applications comme dans le domaine militaire. Un bon exemple est la reconnaissance automatique de la cible par le traitement d'un signal radar. Le tableau suivant donne un aperçu général sur les différents domaines d'applications de la reconnaissance de formes:

Domaine	Application	Forme d'entrée	Classes de sorties
Analyse d'images des documents	Reconnaissance d'écriture	Image du document	Caractères, mots
Classification des documents	Recherche Internet	Texte du document	Catégories sémantiques
Classification des documents	filtrage des emails indésirables	Email	Filtrage ou non filtrage
Traitement automatique des bases de données multimédias	Recherche Internet	Fichiers vidéo	Catégorisation des vidéos
Reconnaissance automatique de la parole	Assistance d'annuaire téléphonique	Signal parole	Paroles
Traitement du langage naturel	Extraction des informations	Phrases	Parties du discours
reconnaissance biométrique	Identification des personnes	Visage, iris, empreintes digitales	Utilisateurs autorisés d'accès contrôle
Médical	Diagnostic assisté par ordinateur	Image microscopique	Cellule Cancéreux/saine
Militaire	Reconnaissance automatique de la cible	Image optique ou infrarouge	Type de la cible
Automatisation industrielle	Inspection des cartes de circuit imprimé	Image d'intensité ou de gamme	Produit défectueux ou non
Téledétection	Prédiction de rendement de récolte	Image multi spectrale	Catégories d'utilisation de terre
Bioinformatique	Analyse de Séquences	Séquences d'ADN	Reconnaissance des types de gènes
Datamining	Recherche de modèles significatifs	Points dans un espace multidimensionnel	Groupes compacts et bien séparés

Tableau I.1 : Exemple des applications de la reconnaissance de formes.

I.5 Conclusion

Nous avons consacré ce présent chapitre à la description générale des différents éléments et aspects concernant le processus de la reconnaissance des formes, en effet ce chapitre décrit le principe, l'architecture générale d'un système de reconnaissance des formes, et les différentes approches utilisées dans ce domaine, ainsi qu'un bref aperçu sur les domaines d'applications. Dans le chapitre suivant nous pouvons poursuivre vers un domaine plus spécialisé, vers la reconnaissance de l'écriture manuscrite.

CHAPITRE II

Reconnaissance de l'écriture manuscrite : Etat de l'art

Chapitre II Reconnaissance de l'écriture manuscrite : Etat de l'art

II.1 Introduction

Malgré le développement accéléré des moyens de communication, l'écriture manuscrite reste toujours le moyen de communication visuelle le plus privilégié par l'être humain. En effet, l'écriture est un moyen naturel de communication qui présente l'avantage d'être familier à la majorité des gens. De ce fait, elle constitue un moyen de dialogue facile avec l'ordinateur. Concevoir un système capable de lire un texte manuscrit, devrait permettre aux ordinateurs de traiter une grande variété de données non encore accessibles et peut aussi pouvoir ouvrir un champ d'applications non encore exploitées.

Les avancées récentes dans le domaine de l'analyse et de la reconnaissance des documents, ainsi que la grande évolution de la puissance de calcul des ordinateurs permettent aujourd'hui le développement de systèmes très performants qui tentent de minimiser les interventions humaines et facilite l'interaction homme-machine. Cependant, ces systèmes sont loin d'atteindre des performances comparables à l'être humain dans sa capacité de lire l'écriture cursive spontanée. En absence de toute contrainte et particularisation du problème, la reconnaissance automatique de l'écriture manuscrite s'avère un problème extrêmement complexe et un défi scientifique qui n'a pas de solution satisfaisante jusqu'à ce jour.

II.2 Différents aspects de la Reconnaissance Automatique de l'Ecriture

II.2.1 Mode d'acquisition (en ligne ou hors ligne)

Une différence fondamentale entre les systèmes de reconnaissance de l'écriture est liée au moyen utilisé pour l'acquisition des données. La reconnaissance de l'écriture se divise en deux grandes catégories, à savoir en ligne ou hors ligne.

II.2.1.1 La reconnaissance en ligne (on line)

L'écriture en ligne ou dite encore dynamique, est obtenue lors de sa réalisation par une saisie en continu du tracé. Les données se présentent alors sous la forme d'une séquence de points ordonnés dans le temps. Dans ce cas, le signal est de type 1D et le système de reconnaissance peut bénéficier de la représentation temporelle, ce qui est très analogue à la reconnaissance automatique de la parole. Il n'est donc pas surprenant, de voir l'application de certaines approches qui ont été développées pour la reconnaissance de la parole dans ce domaine [7]. L'acquisition du tracé est assurée généralement par une tablette graphique, munie d'un stylo électronique. Ce mode de reconnaissance, est réservé à l'écriture manuscrite et utilisé dans plusieurs équipements électroniques : Smartphone, Iphone, Ipad, PDA, ou Tablet PC.

II.2.1.2 la Reconnaissance hors ligne (off line)

L'écriture hors ligne ou dite encore statique, est obtenue par la saisie d'un texte imprimé ou manuscrit déjà existant à l'aide d'un scanner ou bien d'une caméra, qui permet de convertir l'écriture en images binaires ou en niveaux de gris, ayant perdu toute information temporelle ou dynamique sur l'ordre du tracé. L'écriture hors ligne, prend l'aspect d'un signal spatial bidimensionnel numérisé. Par conséquent, ici on dispose seulement des images des caractères ou des mots qui ayant perdus toute information sur l'ordre chronologique de leur génération. Ce qui rend la reconnaissance de ce mode d'écriture plus complexe.

II.2. 2 Reconnaissance de l'imprimé ou du manuscrit

Les approches diffèrent selon qu'il s'agisse de reconnaissance de l'écriture imprimée ou manuscrite. Il est évident que la reconnaissance de l'écriture manuscrite est plus difficile à appliquer que la reconnaissance de l'écriture imprimée. En effet, les caractères imprimés sont mis en forme par des polices standard bien alignés horizontalement et souvent bien séparés verticalement, ce qui simplifie la phase de lecture. De plus, le graphisme des caractères est conforme à une fonte unique qui constitue un modèle pour l'identification. Dans le cas du manuscrit, il existe une grande variété des styles d'écriture humaine, dont les caractères ne présentent aucune sorte de régularité dans la taille ou le graphisme avec une très grande variabilité des espacements et des alignements. Cela complique souvent la phase de reconnaissance.

II.2.3 Approches de reconnaissance

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

II.2.3.1 Approche globale

L'approche globale (ou holistique) se base sur une description unique ou globale de toute l'image du mot considéré comme une entité indivisible. Son principe est de modéliser chaque mot par un modèle spécifique, pour une telle application, il faudra générer autant de modèles que de mots présents dans le vocabulaire. Elle présente l'avantage notamment de garder le caractère dans son contexte avoisinant qui rend la modélisation des variations de l'écriture plus efficace. Cependant, cette méthode est pénalisée par la taille mémoire, le temps de calcul et la

complexité du traitement croît linéairement par la taille du lexique considéré, d'où une limitation du vocabulaire.

Les approches globales sont appuyées par certaines expérimentations psycho visuelles mettant en évidence l'aspect global du processus de lecture. Selon ces expériences l'homme n'a pas nécessairement besoin de reconnaître toutes les lettres pour identifier le mot et l'apparence du mot dans sa globalité est un élément important de la lecture [8]. L'approche globale est généralement utilisée pour la reconnaissance de vocabulaires réduits tels que celui des montants littéraires de chèques bancaires [9]. Elle peut être aussi, utilisée pour d'autres tâches, par exemple, à réduire la taille du lexique ou à valider une reconnaissance analytique [10].

II.2.3.2 Approche analytique

Contrairement aux approches globales, qui construisent des modèles globaux de mots sans faire référence aux modèles de lettre, la modélisation analytique s'appuie sur la structure alphabétique du vocabulaire [11]. Cela confère un énorme avantage aux approches analytiques du fait de leur aptitude à modéliser un vocabulaire quelconque quelle que soit sa taille ou sa nature. Il est possible donc de modéliser un vocabulaire de plusieurs milliers de mots, en utilisant uniquement les 28 lettres de l'alphabet. Selon cette approche, le processus de reconnaissance est basé sur une alternance de entre deux phases : la phase de segmentation et la phase d'identification des segments. La difficulté d'une telle approche a été bien expliquée par le célèbre dileme de Sayre [12] en 1973.

Généralement, le choix de l'approche globale ou analytique est imposé par la taille du vocabulaire. Dans le cas de petits lexiques, il est envisageable de créer autant de modèles que de mots possibles [13]. Par contre, pour un vocabulaire de plusieurs milliers de mots, il n'est pas pratique de créer un modèle pour chaque classe ; il est plus pertinent de modéliser au niveau d'entités plus petites comme les caractères [14] ou pseudo caractères. Dans ce cas, le système doit segmenter l'écriture en entités de base en vue de leur reconnaissance.

II.3 Architecture d'un système de reconnaissance de l'écriture hors ligne

Un système de reconnaissance automatique de l'écriture doit transformer un texte écrit sur un support de qualité arbitrairement variable en une représentation compréhensible par une machine. Pour se faire, il nécessite un certain nombre d'étapes à mettre en œuvre. L'ensemble de ces phases forment généralement la structure de système de reconnaissance d'écriture, qui peut se résumer par le schéma de la figure ci-dessous.

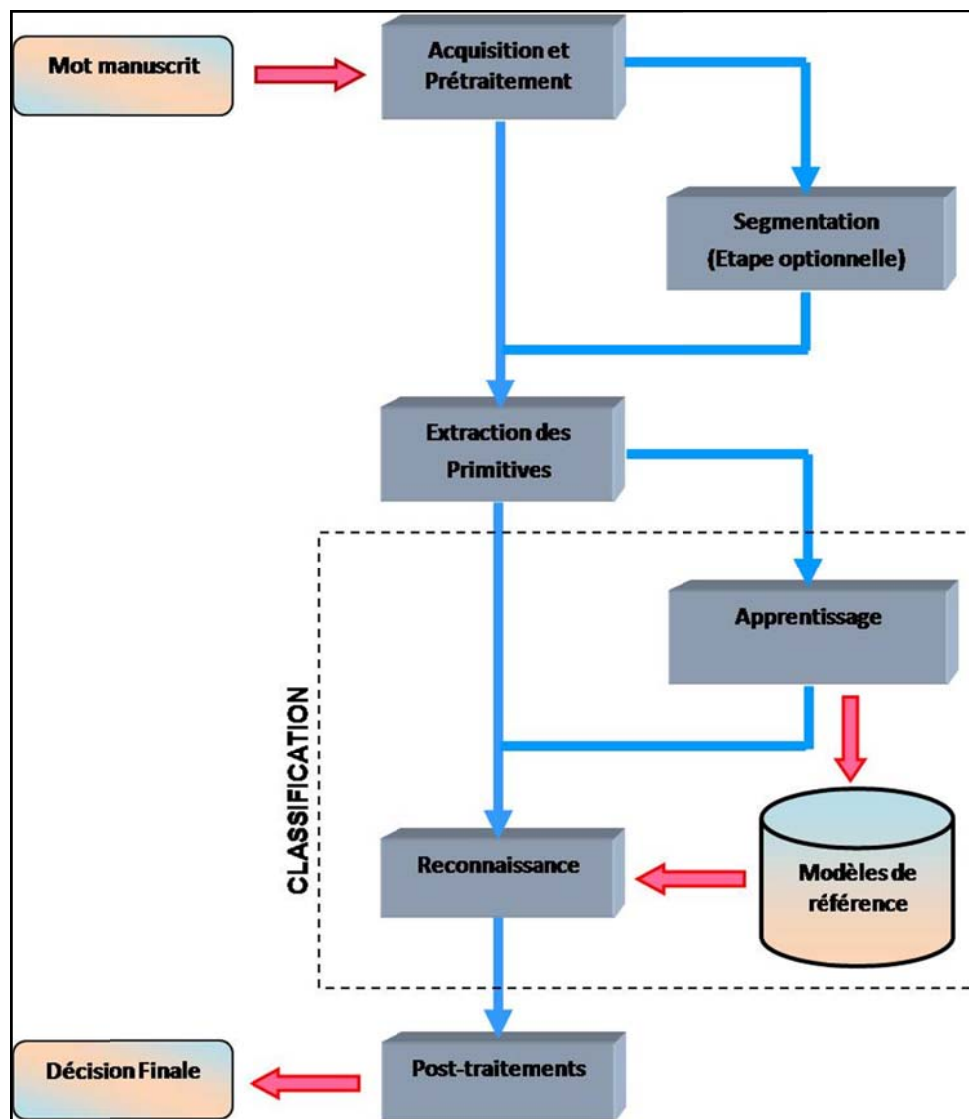


Figure II.1 : Architecture d'un système de reconnaissance d'écriture.

II.3.1 Acquisition

C'est la première étape dans un système de reconnaissance de l'écriture. Cette phase consiste à capter l'image d'un texte au moyen des capteurs physiques et de la convertir dans une forme numériques, adaptée au système de traitement, avec un minimum de dégradation possible. Dans le contexte de l'écriture hors ligne qui fait l'objet de notre étude, les systèmes d'acquisition les plus utilisés sont essentiellement des scanners ou des caméras. Cette étape est assez simple mais très importante car elle peut facilement influencer les étapes suivantes. Elle dépend de deux paramètres importants ; la résolution et le niveau d'éclairage.

Résolution: la résolution normale est 300 dpi, pourtant quand la taille de l'écriture est petite, il faut augmenter la résolution.

Niveau d'éclairage: si on ajuste le scanner pour que l'image soit plus claire, le bruit est réduit mais des très minces disparaissent aussi.

Dans le cas des systèmes de reconnaissance « en ligne », les dispositifs d'acquisition les plus répandus sont des tablettes à numériser ou des "papiers électroniques". La tablette de numérisation, est constituée généralement d'un support rigide portant une surface sensible et un stylet pour écrire.

II.3.2 Prétraitement

L'étape de prétraitement n'est pas spécifique à la reconnaissance de l'écriture manuscrite mais fait partie de tout système de reconnaissance de forme. Elle consiste à améliorer la qualité de l'information, pour préparer les données aux prochaines phases de traitements. Dans le cadre de l'écriture manuscrite, le prétraitement a pour objectif de réduire au maximum les bruits dans l'image et d'éliminer autant que possible les variabilités dues aux styles d'écritures, et à la réduction de la quantité d'information à traiter pour ne garder que les informations les plus pertinentes. Par la suite, nous allons développer les principales opérations du prétraitement.

II.3.2.1 Filtrage

Le filtrage consiste à modifier la distribution fréquentielle des composantes d'un signal selon des spécifications données, généralement dans le but d'améliorer la qualité visuelle de l'image et d'éliminer les effets perturbants des bruits. Il existe une variété de filtre numériques, on peut citer, les filtres passe haut, passe bas, passe bande, filtre médian...etc.

II.3.2.2 Binarisation

La binarisation est souvent la première phase dans les systèmes de traitement et d'analyse d'images. Elle a pour but de diminuer la quantité d'information présente dans l'image et de ne garder que les informations utiles. Elle permet de mieux distinguer les caractères du fond, et consiste à attribuer à chaque pixel de l'image une valeur de 0 ou 1: le "0" représente le noir (le texte) et le "1" représente le blanc (le fond). Pour cela, elle applique en premier lieu l'opération de seuillage.

Le principe de seuillage, consiste à déterminer la valeur du seuil à partir duquel tous les pixels ayant un niveau de gris inférieur à cette valeur sont représentés par un "0" (le noir) et tout les pixels de niveau de gris supérieur auront la valeur "1" (le blanc). On distingue deux types de Seuillage : le seuillage globale et le seuillage adaptatif. Le premier type consiste à prendre un seuil ajustable mais identique pour toute l'image, cette méthode convient pour les documents de contraste fort et de bonne qualité. Dans le cas contraire, le seuillage adaptatif est plus approprié, où le calcul du seuil se fait localement en fonction du voisinage du pixel traité.

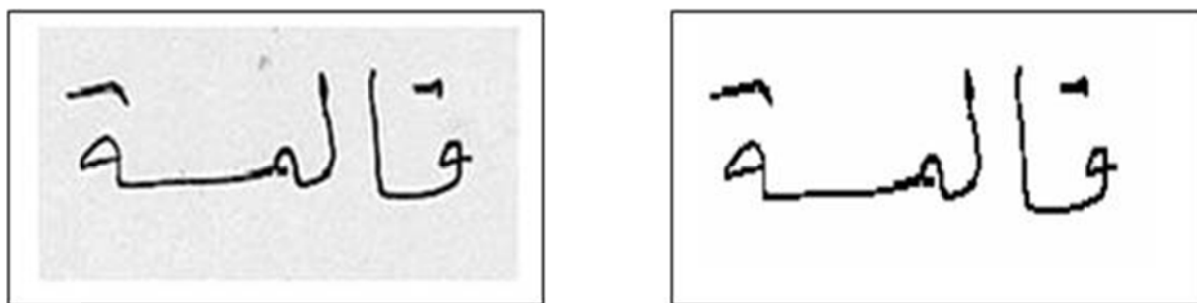


Figure II.2 : Image binarisée par la méthode d'Otsu [15].

II.3.2.3 Lissage

L'image du mot peut être entachée de bruit introduit par la qualité du document ou le mode d'acquisition utilisé, ce qui conduit soit à des absences de points (trous), soit à des points en plus (des empâtements). Le lissage consiste à réduire autant que possible les discontinuités introduites dans l'image et ainsi à rétablir la régularité et la continuité du contour du mot. Les techniques de lissage permettent de résoudre ces problèmes par des opérations locales appelées Nettoyage et bouchage.

Le nettoyage permet de supprimer les petites tâches et les excroissances de la forme. Il est réalisé de différentes manières suivant le type de bruit à enlever [1] :

- Par élimination des points isolés de l'image du mot ou situés à l'extrémité des contours, en appliquant sur l'image le masque ci-dessous dans les huit directions:

0	0	0
x	1	0
0	0	0

Où :

1: représente un point de la forme.

0: représente un point du fond.

x: représente un point quelconque.

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

x	0	0
1	1	0
0	0	0

Pour le bouchage, il s'agit d'égaliser les contours et de boucher les trous internes à la forme des caractères en lui ajoutant des points noirs. Par exemple le bouchage de trous isolés, consiste à mettre à 1 un point de fond si son voisinage correspond au masque suivant :

1	1	1
1	0	1
1	1	1

II.3.2.4 Extraction des contours

En traitement et analyse des images, on appelle un point du contour, un pixel qui se situe sur les frontières séparant deux régions voisines. L'extraction du contour est une technique de changement de représentation qui utilise le contour en tant qu'empreinte des formes contenant une quantité réduite de données. Beaucoup de techniques ont été proposées pour extraire les contours dans une image. La plupart de ces techniques, ont utilisé des opérateurs locaux (directionnels ou pas) de dérivation du premier ordre et du second ordre, parmi lesquelles on peut citer : (Roberts, Prewitt, Sobel, et Kirsh, ect.). Toutefois, ces opérateurs ne donnent souvent pas, eux seuls, de bons résultats sur une image réelle. Pour remédier à ça, une opération de seuillage et une technique de suivi de contours sont en effet nécessaires pour éliminer le bruit et amincir les contours à une épaisseur d'un pixel. Depuis quelques années de nouvelles approches qui procèdent par optimisation de critères prenant en compte un modèle prédéfini du contour à détecter, ont donné de bons résultats par exemple la méthode de Canny [16].



a- Image originale



b- Extraction du contour par la méthode de Canny

Figure II.3 : Extraction du contour.

II.3.2.5 Extraction des composantes connexes

L'extraction des composantes connexes, procédure appelée aussi capture des connexités ou étiquetage des pixels, est une autre technique de changement de représentation largement utilisée en reconnaissance des Formes (RDF) pour segmenter les images binaires et sélectionner les zones d'informations utiles. Cette procédure consiste à regrouper les pixels voisins dans un ensemble appelé composante connexe. Chaque ensemble est disjoint des autres et peut ensuite être facilement isolé ou chaque composante connexe est identifiée d'une manière unique par une étiquette. Généralement deux types de connexité sont considérées: la 8-connexité et la 4-connexité (4 ou 8 représente le nombre de voisins d'un pixel). Pour l'extraction des composantes connexes, il existe deux principaux algorithmes:

- Le premier algorithme est basé sur une procédure de suivi de contour, qui consiste à parcourir le contour d'un objet et en revenant au même point de départ, ce qui conduit à délimiter une composante connexe en affectant une valeur d'étiquette distincte des autres composantes connexes. Cependant, les contours intérieurs correspondant aux éventuels trous sont exclus.

– Le second algorithme appelé aussi double parcours peut se résumer comme suit:

– **Entrées/Sorties:**

– paramètres d'entrée: image binaire (B).

– Résultat de sortie: matrice d'étiquettes (L).

– **Algorithme**

1. Premier parcours de l'image par balayage ligne par ligne en commençant en haut à gauche:

– Pour chaque pixel 1 de l'image binaire (B), on affecte:

– la plus petite étiquette parmi celles de ses voisins haut et gauche.

– ou une nouvelle étiquette si aucun de ces 2 voisins n'est encore étiqueté.

2. deuxième parcours de l'image dans le sens inverse du premier parcours c'est-à-dire en commençant en bas à droite :

– en affectant à chaque pixel précédemment étiqueté, la plus petite étiquette parmi la sienne et celles de ses voisins bas et droite possédant une étiquette non nulle.

II.3.2.6 Squelettisation

La squelettisation, est une technique de changement de représentation qui sert à passer de l'image ordinaire à sa représentation en 'fil de fer'. Elle s'effectue sur une image binaire afin de réduire le tracé d'un mot à un pixel, tout en conservant les propriétés topologiques de celui-ci (nombre de parties, les boucles et les concavités, position des points diacritiques,) et ses propriétés métriques (longueur, les distances entre ces parties).

La procédure de squelettisation sert à réduire la quantité de l'information traitée, et à faciliter également l'extraction de quelques primitives importantes, telles que les points de branchement, de croisement, extrêmes, et de leurs positions relatives.

Le squelette doit seulement remplir trois conditions :

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

Il existe de nombreux travaux de recherche sur la squelettisation qui sont regroupés en deux types d'algorithmes : les algorithmes séquentiels comme l'algorithme de HILDITCH [17], et les algorithmes parallèles comme ZHANG ET SUEN [18].



Figure II.4 : Résultat de squelettisation par l'algorithme de Zhang Suen.

II.3.2.7 Correction de l'inclinaison des caractères

L'inclinaison des caractères, est définie comme étant l'angle entre l'axe correspondant à la direction moyenne des caractères et l'axe vertical ; où les lettres peuvent être inclinées vers la droite ou vers la gauche. L'objectif de l'opération de la correction de l'inclinaison des caractères, est de corriger cette inclinaison de l'écriture pour la rendre plus standard et la plus indépendante possible des styles d'écritures du scripteur. Plusieurs approches ont été proposées pour cette étape de prétraitement; certaines méthodes sont fondées sur la détection et la correction des traits quasi-verticaux, d'autres utilisent les histogrammes de projection verticale.

II.3.2.8 Correction de l'inclinaison des lignes

L'inclinaison des lignes de textes peut être définie comme étant une déformation introduite, souvent pendant la phase de la numérisation du support, ou comme les défauts d'orientation causés par les scripteurs. Il est nécessaire alors de corriger la pente d'un mot et de redresser l'inclinaison des lettres afin d'éliminer autant que possible les variabilités liées aux styles d'écritures et également à faciliter la phase de segmentation. Ce prétraitement inclut généralement deux étapes ; La première permet l'estimation de l'angle d'inclinaison globale de la ligne (α) par un algorithme de détection de l'angle d'inclinaison. La deuxième étape consiste à

corriger l'inclinaison par l'application d'une rotation de l'image d'un angle (α) dans le sens opposé pour faire la correction.

Plusieurs méthodes ont été proposées pour la correction de l'inclinaison des lignes. Les deux plus populaires, sont la transformée de Hough appliquée sur les centre de gravité des composantes connexes [19] et les histogrammes de projection [20].

II.3.2.9 Normalisation de la taille

La normalisation de la taille, consiste à ramener les images de mots ou de caractères à des tailles standards à fin de réduire autant que possible les variabilités causées par les styles d'écritures, pour la rendre plus indépendante du scripteur. L'importance de cette étape de prétraitement, est de rendre l'étape d'extraction de primitives moins influencée par la variation des styles d'écritures. De nombreux travaux ont été proposés dans la littérature pour la normalisation de la taille de l'écriture, la méthode la plus connue est l'algorithme de Srihari [21]. Cet algorithme, procède en deux étapes. La première consiste à normaliser le caractère en hauteur et la seconde, en largeur. L'ordre de normalisation ainsi choisi évite la déformation des caractères fins.

II.3.3 Extractions des primitives

En Reconnaissance de l'écriture manuscrite, les données à traiter sont des images. Généralement, cette représentation de l'information, n'est pas adaptée pour les systèmes de reconnaissance. Donc une étape d'extraction de caractéristiques est réalisée afin d'extraire l'information la plus discriminante pour la tâche de reconnaissance et également pour réduire le volume d'informations qui sera fourni au système de reconnaissance, tout en évitant le risque de perte des informations importantes et signifiantes. C'est-à-dire pour un problème de classification donné, la principale qualité recherchée pour un ensemble de primitives, est sa capacité de rassembler les formes appartenant à une même classe dans une même partition de l'espace de représentation, tout en éloignant autant que possible les autres. Cette qualité est communément appelée pouvoir discriminant de l'ensemble de caractéristiques.

Parmi les problèmes fondamentaux de la reconnaissance d'écriture manuscrite, est de déterminer quelles caractéristiques employer pour obtenir des résultats satisfaisants de classification. Malheureusement, il n'y a pas de méthode exacte pour guider le choix de ces primitives, mais on peut formuler les points suivants à propos de ces primitives:

- **Discriminabilité:** une bonne primitive doit avoir des valeurs significativement différentes pour des exemples appartenant à des classes différentes.

- **Fiabilité** : une bonne primitive doit avoir des valeurs très similaires (très proches) pour les exemples d'une même classe.
- **Indépendance**: dans un vecteur de caractéristiques, une primitive quelconque ne doit pas dépendre d'une autre primitive.
- **Nombre**: cette propriété concerne, le coût, la complexité et les exigences en temps de calcul d'un système de reconnaissance qui croient automatiquement avec la croissance du nombre de primitives utilisées. Il est donc nécessaire de choisir les caractéristiques les plus utiles, pour rendre le système d'une part moins gourmand en temps de calcul, et d'autre part plus fiable.

Les primitives peuvent être classées en trois catégories qui sont statistiques, structurelles, et transformations globales.

1. Les caractéristiques statistiques

Les primitives statistiques, consistent à représenter l'information distribuée sur la totalité de l'image. Elles sont générées généralement par l'estimation de la densité de pixels appartenant à l'image entière ou dans certaines parties uniquement, en utilisant des mesures statistiques (entropie, moyenne, variance, etc.) associées à des caractéristiques locales.

1.1 Le Zonage (Zoning)

La technique de zonage est largement utilisée dans le domaine de la reconnaissance de l'écriture à cause de la simplicité de sa mise en œuvre [22], elle consiste à diviser l'image de la forme à reconnaître (mot ou caractères), son contour ou son squelette en plusieurs zones de façons différentes et à effectuer une extraction de primitives sur chaque zone et formant par concaténation le vecteur de primitives. Le but de cette méthode d'extraction, est d'obtenir des caractéristiques locales au lieu des caractéristiques globales afin d'augmenter le pouvoir discriminant. Il faut noter seulement que le choix de la stratégie de division de la zone d'intérêt, influe fortement sur l'efficacité de cette technique.

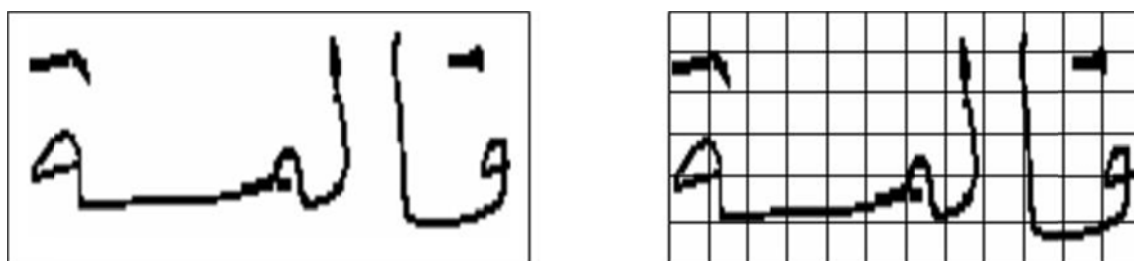


Figure II.5 : Exemple de la technique de Zoning.

1.2 L'approche VH2D

L'approche VH2D [23] (Vertical-Horizontal-2Diagonal), proposé par Cheng et Xia consiste à projeter chaque caractère sur l'abscisse, l'ordonnée, ainsi que sur les diagonales 45° et 135°. Les projections s'effectuent en calculant la somme des valeurs des pixels i_{mn} selon une direction donnée.

- **Projection verticale**

La projection verticale d'une image $I=(i_{mn})$ (de dimension $N \times N$) représentant un caractère C est dénotée par:

$$P^v = [P_1^v, P_2^v, \dots, P_N^v], \text{ où } P_n^v = \sum_{m=1}^N i_{mn} \quad (\text{II.1})$$

- **Projection horizontale**

La projection horizontale d'une image $I=(i_{mn})$ (de dimension $N \times N$) représentant un caractère C est dénotée par:

$$P^h = [P_1^h, P_2^h, \dots, P_N^h], \text{ où } P_m^h = \sum_{n=1}^N i_{mn} \quad (\text{II.2})$$

- **Projection sur la diagonale 45°**

La projection sur la diagonale 45° d'une image $I=(i_{mn})$ (de dimension $N \times N$) représentant un caractère C est dénotée par:

$$P_m^{d1} = \begin{cases} \sum_{l=N-m+1}^N \sum_{k=1}^m i_{lk} & 1 \leq m \leq N \quad \text{et } l=k+N-1 \\ \sum_{l=1}^{2N-m} \sum_{k=m-N+1}^N i_{lk} & 1+N \leq m \leq 2N-1 \quad \text{et } l=k+N-1 \end{cases} \quad P^{d1} = [P_1^{d1}, P_2^{d1}, \dots, P_{2N-1}^{d1}], \text{ où} \quad (\text{II.3})$$

- **Projection sur la diagonale 135°**

La projection sur la diagonale 135° d'une image $I=(i_{mn})$ (de dimension $N \times N$) représentant un caractère C est dénotée par:

$$P_m^{d2} = \begin{cases} \sum_{l=1}^m \sum_{k=1}^m i_{lk} & 1 \leq m \leq N \quad \text{et } l=m-l+1 \\ \sum_{l=m-N+1}^N \sum_{k=m-N+1}^N i_{lk} & 1+N \leq m \leq 2N-1 \quad \text{et } l=m-l+1 \end{cases} \quad P^{d2} = [P_1^{d2}, P_2^{d2}, \dots, P_{2N-1}^{d2}], \text{ où} \quad (\text{II.4})$$

2. Les caractéristiques structurelles

Sont celles qui décrivent le mot dans sa globalité, elles sont généralement extraites non pas de l'image brute, mais à partir d'une représentation de la forme par un squelette ou un contour. Elles représentent les propriétés topologiques et géométriques de la forme. La faible sensibilité au bruit et aux variations des styles d'écriture, rend cette famille de primitives largement utilisée pour la reconnaissance de l'écriture manuscrite.

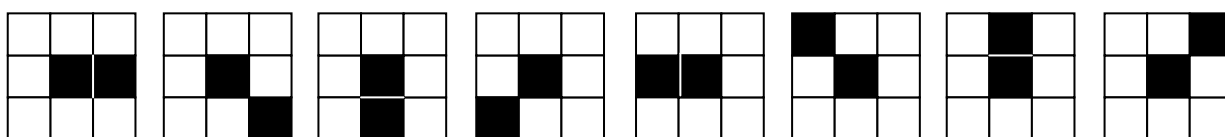
- Extraction des points caractéristiques

La représentation en squelette du mot, permet d'extraire facilement certains types de caractéristiques qui sont difficiles à obtenir à partir de sa représentation bitmap. Parmi ces caractéristiques, on trouve les points caractéristiques, qui sont les pixels noirs dans le squelette du mot ayant un nombre de voisins noirs différents de 0 ou 2. Il y en a deux types : les points extrêmes et de jonctions.

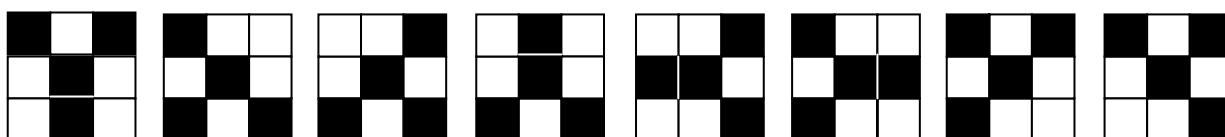
- Un point extrême correspond au début/fin d'un segment de ligne.
- Un point de jonction connecte trois branches ou plus. On y trouve deux types de points de branchement et de croisement.
- Un point de branchement correspond à un branchement de trois traits du squelette de l'image.
- Un point de croisement correspond à un point où quatre traits se croisent.

L'extraction des points caractéristiques se fait par le placement d'un masque de taille 3x3 sur les pixels noirs afin de calculer le nombre N de voisins noirs du pixel central pour chaque pixel noir du squelette.

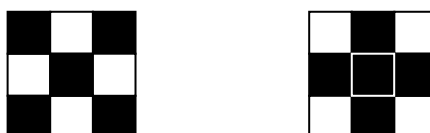
- Si $N=1$: le pixel est un point extrême.



- Si $N=3$: le point est un point de branchement.



- Si $N=4$: le point est un point de croisement.



3. Les transformations globales

Ce type de caractéristiques, se base sur une transformation globale de l'image, qui sert à convertir la représentation en pixels de l'image en une représentation plus abstraite, afin de réduire la dimension, sans perdre les informations significatives sur la forme à reconnaître. Par exemple : la transformée en DCT, les ondelettes, la transformée de Fourier, Gabor et les moments de Zernike.

3.1 La transformée en cosinus discrète (2DCT)

La transformée en cosinus discrète DCT (Discrete Cosine Transform) est une fonction mathématique, qui consiste à changer le domaine de représentation d'un signal. La DCT est largement utilisée dans le domaine de la reconnaissance de formes et spécialement en compression des images. En effet, l'une des caractéristiques les plus intéressantes de la DCT est son excellente capacité de regrouper l'énergie de l'image dans un petit nombre de coefficients (coefficients de basses fréquences). L'application de la DCT va transformer l'image du domaine spatial en une représentation identique dans le domaine fréquentiel. Ce changement de domaine est très intéressant parce qu'une image ordinaire admet une grande continuité entre les valeurs des pixels. Les hautes fréquences étant réservées à des changements brusques d'intensité du pixel, qui sont généralement une minorité dans une image. Ainsi, les basses fréquences peuvent représenter l'intégralité de l'information de l'image sur un très faible nombre de coefficients. L'application de la DCT sur une image de taille $M \times N$ donne en résultat une matrice de même taille. Les basses fréquences se trouvant en haut à gauche de la matrice de DCT, et les hautes fréquences en bas à droite. La transformation matricielle DCT étant orthogonale, elle a une méthode de transformation inverse pour pouvoir revenir dans le domaine spatial.

Kernel et Stiefelhagen [24], ont montré que l'information nécessaire à la réalisation d'un taux de la classification performant est contenue dans les premiers coefficients de la DCT (les basses fréquences) par balayage. Ces coefficients sont calculés par l'équation suivante :

$$F(u, v) = C(u)C(v) \left[\sum_{x=0}^{(M-1)} \sum_{y=0}^{(N-1)} f(x, y) \cos \frac{(2x+1)u\pi}{2M} \cos \frac{(2y+1)v\pi}{2N} \right] \quad (\text{II.5})$$

Où : $f(x,y)$ est l'image d'entrée. $M \times N$ représente la taille de l'image, et :

$$C(u) = \frac{1}{\sqrt{M}} \quad \text{Si } u = 0 \quad \text{Ou} \quad \sqrt{\frac{2}{M}} \quad \text{Si } 1 \leq u \leq M-1; \quad (\text{II.6})$$

3.2 Histogrammes de gradients orientés (HOG)

Les histogrammes de gradients orientés (Histograms of Oriented Gradients ou HOG), sont une méthode d'extraction de primitives, basée sur le calcul de gradient qui peut se faire en tout point d'une image. Ils ont été introduits par [25] dans le cadre de la détection de piétons. Depuis leur apparition, ils ont été largement utilisés pour un certain nombre de problèmes de vision par ordinateur avec un grand succès. L'idée de base des descripteurs HOG est de discrétiser l'orientation du gradient dans l'image à l'aide d'histogrammes. Ainsi les contours, donc la forme des objets, sont codés ce qui permet de les reconnaître. L'image de la forme à caractériser est découpée en plusieurs cellules de petites tailles et chaque cellule est caractérisée par un histogramme d'orientations des gradients calculés sur les pixels à l'intérieur de la cellule. Le descripteur de HOG complet est alors formé par la concaténation de ces histogrammes. Les étapes suivantes décrivent en détails la procédure de calcul des descripteurs de HOG.

Etape1 : Calcul du gradient, il s'agit de calculer le gradient pour tous les points de l'image ; usuellement deux masques de dérivation sont appliqués sur l'image, un horizontal $[-1, 0, 1]$ et un vertical $[-1, 0, 1]^T$.

Etape 2: La Norme et l'orientation du gradient sont calculés en utilisant les équations suivantes:

$$NG = \sqrt{G_h(x, y)^2 + G_v(x, y)^2} \quad (II.7)$$

$$OG = \arctan \frac{G_h(x, y)}{G_v(x, y)} \quad (II.8)$$

Où: G_h et G_v représente le gradient horizontal et vertical; NG et OG représentent la Norme et l'orientation respectivement au point (x, y) dans l'image.

Etape 3: L'image est divisée en $N*N$ cellules et l'histogramme des orientations, est calculé pour chaque cellule. Si l'histogramme est divisé en K barres (bins) en fonction de l'orientation, la valeur du $i^{\text{ème}}$ bin V_i pour la cellule C est calculée de la manière suivante:

$$V_i = \sum_{(x, y) \in C} SG(x, y)/OG(x, y) \in \text{bin } i \quad (II.9)$$

Etape 4 : L'histogramme de chaque cellule est normalisé.

Etape 5: Les histogrammes de toutes les cellules sont concaténés pour former le descripteur. La dimension du descripteur est en fonction de la taille de la cellule N et le nombre d'intervalles dans l'histogramme K (bins).

II.3.4 Classification

En reconnaissance de l'écriture manuscrite, la phase de classification est considérée comme étant l'une des étapes les plus importantes, qui conditionne en grande partie les performances des systèmes. La classification consiste à élaborer une règle de décision qui identifie les classes auxquelles appartiennent les formes à partir des caractéristiques préalablement choisies et calculés ; c'est-à-dire le passage de l'espace de représentation ou codage vers l'espace de décision. Vu l'importance de l'étape de la classification, nous avons envisagé de consacrer tout un chapitre qui présente en détail les différentes méthodes de classification employées dans notre système de reconnaissance des mots arabes manuscrits (chapitre IV de cette thèse).

II.3.5 Post-traitements

Ce module est facultatif, peut être ajouté pour améliorer le taux de reconnaissance en introduisant des informations contextuelles permettant de lever l'ambiguïté dans la reconnaissance de certains caractères ou mots. Ces informations peuvent se présenter sous forme de:

II.3.5.1 Connaissances pragmatiques

Dans certaines applications comme le tri postal, on connaît bien le nombre de chiffres du code postal, qui peut être confronté avec la reconnaissance du bureau distributeur, pour permettre de lever certaines ambiguïtés.

II.3.5.2 Connaissances linguistiques

Le rôle de la phase de post traitement dans un système de reconnaissance de l'écriture manuscrite, consiste à vérifier et à corriger les erreurs de l'étape de reconnaissance, en tenant compte des connaissances linguistiques de différents niveaux (lexical, syntaxique ou sémantique) propres au document.

L'utilisation d'un lexique, permet de vérifier la présence d'un mot dans un dictionnaire de références. Si la reconnaissance s'est effectuée au niveau caractère, le module post traitement permet par exemple de vérifier à l'aide d'un dictionnaire l'exactitude des mots obtenus en combinant les différentes hypothèses fournies par l'étape de reconnaissance. On attribue ensuite à chaque mot présent dans le lexique, une valeur de confiance qui dépend de celles attribuées par l'étape de reconnaissance aux différents caractères composant ce mot.

L'emploi d'un ensemble de règles grammaticales prédéfinies permet de prendre en compte le contexte syntaxique d'un document afin de confirmer ou infirmer la séquence de mots proposés. Cette approche est largement utilisée dans les systèmes de lecture des montants littéraux et numériques de chèques bancaires.

Il est également possible de mettre en corrélation des résultats obtenus sur des champs différents de l'image, mais codant la même information (cas des chèques bancaires, des bons de commande ou des enveloppes postales).

II.4 Mesure de performance

Afin de juger l'efficacité de ces systèmes de reconnaissance de l'écriture et de mesurer leurs performances ; plusieurs critères ont été proposés dans la littérature. Ces évaluations sont généralement caractérisées, par les différents taux ci-dessous:

- **le taux de reconnaissance:** représente le pourcentage de caractères ou mot reconnus parmi ceux présentés au système.
- **Le taux de confusion:** représente le pourcentage de caractères ou mots acceptés par le système, mais classés de façon incorrecte (à cause d'un problème d'ambiguïté).
- **Le taux de rejet:** représente le pourcentage de caractère ou mots rejetés (c'est à dire les formes pour lesquelles le système refuse de se prononcer) parmi ceux représentés à l'entrée du système.

II.5 Domaines d'applications de la reconnaissance de l'écriture

Les progrès réalisés dans le domaine de la reconnaissance de l'écriture manuscrite tant d'un point de vue théorique que méthodologique se concrétisent aujourd'hui, par un certain nombre d'applications opérationnelles dans plusieurs domaines d'activité parmi lesquelles on peut citer :

- la lecture des adresses postales et le tri automatique du courrier [26].
- la lecture et l'authentification de chèques bancaires [27].
- l'indexation et l'archivage automatique des documents.
- Le traitement automatique des formulaires et des documents administratifs.
- la reconnaissance des numéros minéralogiques pour le contrôle routier.
- la reconnaissance des documents techniques (schémas électroniques, dessins techniques, équations mathématiques, plans architecturaux, plans cartographiques, etc.).

II.6 Conclusion

Dans ce chapitre, nous avons présenté le domaine de la reconnaissance de l'écriture manuscrite d'une manière générale. En mettant le point sur différents aspects d'un système de reconnaissance de l'écriture manuscrite ainsi que les différentes phases qui le constituent à savoir, l'acquisition, le prétraitement, l'extraction des primitives ...Etc. Nous avons abordé chaque étape du système avec l'analyse et la catégorisation des différentes méthodes et techniques qui y sont développées et utilisées dans le cadre de la reconnaissance de l'écriture manuscrite.

CHAPITRE III

Reconnaissance de l'écriture Arabe manuscrite

Chapitre III Reconnaissance de l'écriture Arabe manuscrite.

III.1 Introduction

Contrairement à la majorité d'autres scripts, la reconnaissance de l'écriture arabe manuscrite demeure encore aujourd'hui un défi scientifique très complexe, qui n'a pas encore de solution satisfaisante. Ceci est dû principalement à la complexité morphologique de l'écriture arabe et aux grandes variations des styles d'écriture qui posent de nombreux problèmes aux systèmes de reconnaissance automatique. Il est très important avant de présenter l'approche proposée pour la reconnaissance hors ligne des mots arabes manuscrits, d'étudier les différentes techniques utilisées pour la reconnaissance de l'écriture arabe manuscrite, de présenter ces caractéristiques et d'analyser les différents problèmes rencontrés par les chercheurs afin de concevoir des systèmes de hautes performances.

III.2 Présentation de la langue arabe

La langue arabe est la langue du coran, elle appartient à la famille des langues sémitiques parlées au Proche-Orient (de la Méditerranée à l'Iran), et l'une des langues les plus populaires au monde, parlées par plus de 422 millions de personnes, et plus de 1 milliard de personnes, utilisent la langue arabe dans plusieurs activités religieuses (l'islam). C'est la langue officielle de 26 pays et de plusieurs organismes internationaux et l'une des six langues officielles de l'Organisation des Nations unies. L'ensemble de l'alphabet utilisé pour écrire cette langue, est l'alphabet arabe, (voir le tableau III.1). Il y a également un certain nombre de langues qui utilisent l'alphabet arabe, tels que le persan, l'urdu, le Kurde et le Jawi.

III.3 Caractéristiques de l'écriture arabe

L'écriture arabe est par nature cursive, aussi bien dans sa forme imprimé que manuscrite. Ce qui signifie que certaines lettres doivent être connectées à d'autres ce qui rend la tâche de la reconnaissance très complexe. L'alphabet arabe, contient 28 lettres et le texte arabe s'écrit cursivement de droite à gauche. Les notions de lettre majuscule et lettre minuscule n'existent pas. Contrairement au latin, chaque lettre arabe a deux ou quatre formes selon sa position dans le mot, 22 lettres peuvent prendre 4 formes selon leur apparition dans le texte; au début, au milieu, à la fin, ou isolé et 6 lettres non cursive (ne peuvent pas être liées à la lettre suivante) avec seulement deux formes correspondant à l'isolé et la position de fin. Le Tableau III.1, illustre les différentes lettres arabes ainsi que leurs formes au début, au milieu ou en fin de mot. La présence de ces 6 lettres dans le texte introduit donc une coupure dans le mot d'où la notion de pseudo mots, le mot donc peut ainsi être composé d'un ou de plusieurs de pseudo mots (Figure III.1).

Caractère	Début	Milieu	Fin	isolé
Alif			ا	ا
Ba	ب	ب	ب	ب
Ta	ت	ت	ت ou ة	ت ou ة
Tha	ث	ث	ث	ث
Nun	ن	ن	ن	ن
Ya	ي	ي	ي	ي
Jim	ج	ج	ج	ج
Ha	ح	ح	ح	ح
Kha	خ	خ	خ	خ
Dal			د	د
Thé			ذ	ذ
Ra			ر	ر
Za			ز	ز
Waw			و	و
Sin	س	س	س	س
Chin	ش	ش	ش	ش
Sad	ص	ص	ص	ص
Dhad	ض	ض	ض	ض
Tad	ط	ط	ط	ط
Dha	ظ	ظ	ظ	ظ
Ayn	ع	ع	ع	ع
Ghayn	غ	غ	غ	غ
Fa	ف	ف	ف	ف
Qaf	ق	ق	ق	ق
Kaf	ك	ك	ك	ك
Lam	ل	ل	ل	ل
Mim	م	م	م	م
He	ه	ه	ه	ه

Tableau III.1 : Les 28 lettres arabes avec leurs différentes formes d'apparitions dans un mot.

- La longueur de l'espace inter-mot (l'espacement entre deux mots distincts) est généralement supérieur à l'espace intra-mots; entre caractères non attachés. Cependant, dans le cas du manuscrit, l'espacement entre les différents pseudo-mots d'un même mot n'est pas forcément supérieur à l'espacement entre deux mots différents, ce qui complique beaucoup la procédure de segmentation [28].
- Certains caractères arabes incluent une boucle qui peut avoir différentes formes.

III.3.2 Signes diacritiques

Un signe diacritique est une composante auxiliaire d'une lettre, qui vient la compléter ou en modifier le sens. On distingue, les signes de voyelles, les points diacritiques, et autres signes secondaires. L'arabe est une écriture riche en points diacritiques. En plus de ça l'écriture de ces points est souvent respectée par les scripteurs car ces derniers permettent de distinguer entre les lettres ayant le même corps principal. Plusieurs lettres d'alphabet arabe partagent la même forme et sont différenciées seulement en termes du nombre et du placement de points sur les lettres. Ces points sont placés soit au dessus, soit en dessous de la forme à laquelle ils sont associés, mais jamais les deux à la fois. Dans l'alphabet arabe, 15 lettres parmi les 28 possèdent un ou plusieurs points. Le Tableau III.2 présente les lettres arabes ayant des points diacritiques ainsi que leurs nombres et positions.

Nombre de points	Position	
	Au-dessus	Au-dessous
Un point diacritique	خ ز ض ظ غ ف ن ذ	ب ج
Deux points diacritiques	ت ق	ي
Trois points diacritiques	ث ش	/

Tableau III.2 : Lettres arabes ayant des points diacritiques.

Les autres signes diacritiques sont la hamza (Zigzag), la Chadaah et la Madaah (voir Figure III.3). La Chadaah est une accentuation de la lettre (c'est l'équivalent d'une consonne doublée). Hamza et Madaah suivent des contraintes morphosyntaxiques plus complexes.

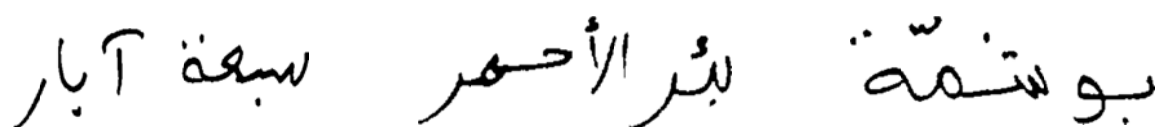


Figure III.3 : Mots arabes incluant la Hamza, la Chadda et la Madda.

III.3.3 Ascendants et descendants

Comme dans l'écriture latine, l'écriture arabe contient des ascendants et des descendants. En arabe, les descendants ou les jambages peuvent se prolonger horizontalement sous la bande de base, ce qui conduit à une superposition verticale entre la lettre qui comprend le descendant et la lettre suivante. L'ascendant appelé aussi hampe, est détectée si l'ordonnée supérieure d'un caractère est située dans la zone supérieure à la zone médiane (bande de base) du mot c'est-à-dire supérieure à la ligne de base supérieure.

III.3.4 ligatures et chevauchement verticaux

Une ligature est la fusion de deux caractères ou plus pour n'en former qu'un seul nouveau, considéré ou non comme un caractère (à part entière). Pour l'écriture arabe, on distingue trois types de ligatures : les ligatures contextuelles, linguistiques et esthétiques. La ligature contextuelle se manifeste sous la forme d'une chaîne de caractères prenant des formes spéciales suivant leur position dans le mot en suivant des règles grammaticales strictes et liées uniquement à l'écriture. Les ligatures linguistiques obéissent à certaines règles grammaticales et sont obligatoires dans l'écriture d'une langue donnée. L'exemple le plus connu de ce type de ligatures, est la ligature arabe Lam-Alif. À la différence des deux premiers types, les ligatures esthétiques sont des graphies ayant une nature optionnelle, elles existent pour des raisons esthétiques, de lisibilité ou de tradition, et peuvent être remplacées par leurs composantes sans changer la validité grammaticale, ou le sens du texte. Des chevauchements verticaux sans contact peuvent se produire par l'intersection des descendants qui se prolongent horizontalement sous la ligne de base et les pseudos mots suivants.

III.4 Principales bases de données existantes

La disponibilité des bases de données standards dans le domaine de la reconnaissance de l'écriture manuscrite, joue un rôle essentiel dans l'évaluation et la comparaison des résultats obtenus par les différents groupes de recherche. Le domaine de la reconnaissance automatique de l'écriture arabe manuscrite, a été caractérisé pendant de nombreuses années par l'absence d'une base de données de référence standard, qui permette des comparaisons objectives entre les différents travaux de recherche. Mais c'est la base IFN/ENIT, gratuite pour la recherche académique, qui s'est imposée comme base de données de référence standard pour la comparaison et l'évaluation des performances des systèmes de reconnaissance de l'écriture arabe manuscrite.

III.4.1 La base Al-Isra 1999

En 1999, la base de données Al-ISRA a été collectée par un groupe de chercheurs [29] de l'Université de la Colombie-Britannique au Canada. Elle contient 37.000 mots arabes, 10.000 chiffres (en deux types, "Mashriqi" et "Maghribi"), 2500 signatures et 500 phrases arabes simples (ligne de la poésie) avec les voyelles écrites par cinq cents étudiants choisis au hasard à l'université d'Al-Isra à Amman, Jordanie. Cette base de données est considérée comme étant la première base publiée dans le domaine de la reconnaissance de l'écriture arabe manuscrite. Elle est disponible pour les chercheurs Canadiens, mais son accès est soumis à des restrictions pour les chercheurs du reste du monde.

III.4.2 La base CENPARMI '1', 2000

En l'an 2000, au sein du Centre for Pattern Recognition and Machine Intelligence (CENPARMI), une équipe de recherche [30], a achevé la réalisation d'une base de données pour la reconnaissance des chèques arabes manuscrits, scannés en niveaux de gris avec une résolution de 300 dpi (voir Figure III.4). Une sélection de 3000 images de chèques a permis de mettre au point quatre bases de données :

- Une base de 2499 montants littéraux.
- Une base de 2499 montants numériques écrits avec les chiffres indiens.
- Une base de 29498 pseudo-mots
- Une base de 15175 chiffres indiens isolés.



Figure III. 4 : Exemple de chèque saoudien de la base CENPARMI '1' (figure extraite de [30]).

III.4.3 La base IFN/ENIT, 2002

La base de données de référence IFN/ENIT[31] de noms de villes tunisiennes, est une base de mots arabes manuscrits dont le développement est le fruit d'un projet de coopération entre le laboratoire de systèmes et de traitement de signal de l'ENIT (Ecole Nationale d'Ingénieurs de Tunis) et le laboratoire de recherche sur la reconnaissance de l'écriture arabe de l'IFN (Institute Fur Nachrichten technik, Institute of Communications Technology) de l'université de Braunschweig, Allemagne.

Cette base de données est composée de 5 sous-ensembles (a-e), contenant en totalité (en version v2.0p1e) 32492 images de noms de villes/villages tunisiens (plus de détails sont présentés dans le tableau). Plus de 1000 scripteurs de différents âges et professions ont participé à collecter ces noms (noms manuscrits de 937 villes/villages tunisiens avec leur codes postaux). Ces scripteurs ont été priés de remplir des formulaires avec les noms de villes/villages tunisiens et les codes postaux correspondants. Ces formulaires ont été scannés à 300 dpi, puis les images de noms de villes ont été extraites (sous formats TIFF et BMP). Ces images sont accompagnées

des fichiers d'annotations qui contiennent des informations sur l'image, le nom de ville/village en code ASCII et la séquence détaillée des caractères. Ils indiquent également la position de la ligne de base (pour le set a), le nombre de mots, caractères et pseudo-mots et enfin des informations sur la qualité de la base de ligne et de l'écriture (un exemple de cette annotation est présenté dans le tableau III.3). Cette base, par sa taille (plus de 26000 mots) et sa disponibilité (gratuite pour la recherche académique) s'est imposée comme la base de référence des travaux en reconnaissance de l'écriture arabe manuscrite. Elle sert de support à l'organisation des compétitions scientifiques internationales dans le domaine de la reconnaissance de l'écriture arabe manuscrite : ICDAR (Arabic Handwritten Competition) 2005, 2007, 2009 et 2011.

Image		
Ground truth:		
Postcode	4021	3024
Global Word	القلاع الصغرى	شعال
Character shape Sequence	aaA laB kaM laM ayM teE aaA laB saMlIL ghM raE eeA	shB ayM aaE laA
Baseline y1,y2	92, 90	92, 90
Baseline quality	B1 (B 1+OK ; B2=bad)	B1
Quantity of words	2	1
Quantity of PAWs	5	2
Quantity of Characters	12	4
Top line information Y1,Y2 (in data set a only!!!)	60, 58	44, 58

Tableau III.3 : Deux images et leurs annotations respectives dans la base IFN/ENIT.B pour Begin M pour Middle, A pour Alone et F pour Final.

III.4. 4 La base CEDARABIC, 2005

Dix scripteurs différents ont contribué à collecté les images de La base de données CEDARABIC [32]. Chaque scripteur d'entre eux était invité à écrire 10 pages complètes différentes de texte manuscrit. Chaque page comprend entre 150 et 200 mots, soit un total d'environ 20000 mots. Les documents sont scannés en 300 dpi. Ces fichiers sont accompagnés d'une annotation manuelle complète qui contient : une segmentation en mots, une séquence des lettres arabes, ainsi que la prononciation et la traduction en anglais. Cette base de données est donc particulièrement adaptée pour des travaux sur la segmentation en mots et pour la recherche de mots clés dans une page d'écriture arabe manuscrite.

III.4.5 La base KHATT, 2012

La base de données KHATT (**KFUPM Handwritten Arabic Text**) est une base de données de texte arabe non contraint, écrite par 1000 scripteurs différents [33]. Cette base est dédiée à la recherche scientifique développée par une équipe de recherche de KFUPM, Dhahran, Arabie Saoudite. Elle contient 2000 des paragraphes uniques aléatoirement choisis dans différents domaines comme l'art, l'enseignement, la santé, la nature et la technologie, ainsi que 2000 images de paragraphes similaires, couvrant toutes les formes de caractères arabes et des paragraphes facultativement écrits sur des sujets ouverts. Les images sont scannées à des résolutions différentes (200, 300, et 600 dpi) ; ces images sont accompagnées des fichiers d'annotations qui contiennent des informations sur le texte écrit. Cette base est divisée en 3 sous-ensembles, 70% pour l'apprentissage, 15% pour la validation et 15% pour le test.

III.4.6 La Base AHTID /MW, 2012

La base AHTID/MW [34] (**Arabic Handwritten Text Images Database written by Multiple Writers**) a été collecté au sein du Laboratoire MIRACL, ISIMS, Université de Sfax - Tunisie en collaboration avec l'Institut de la Technologie de Communications (IFN), Braunschweig, Allemagne. Cette base peut être utilisée pour la recherche dans le domaine de la reconnaissance d'écriture arabe manuscrite, la segmentation des mots et l'identification de scripteur. La base AHTID/MW contient 3710 lignes de texte et 22,896 mots écrits par 53 scripteurs différents dont l'arabe est leur langue maternelle. Ces images sont scannées en niveau de gris avec une résolution de 300dpi en format BMP. Cette base est divisée en quatre ensembles distincts. Chaque image de mot/ligne dans la base de données AHTID/MW est entièrement décrite en utilisant un fichier XML contenant des informations sur la séquence des PAW (party of Arabic Word), ainsi que la séquence des caractères pour chaque PAW.

III.5 Surveys existants

Plusieurs surveys ont été proposés dans domaine de la reconnaissance de l'écriture arabe manuscrite, pour présenter l'état de l'art et les différentes approches développées.

III.5.1 A. Amin, 1998

Dans [35] Adnane amine, présente un état de l'art dans le domaine de la reconnaissance de l'écriture arabe (manuscrite et imprimée), ainsi que les différentes approches de la reconnaissance de l'écriture arabe, les approches en ligne / hors ligne et les approches globales/Analytiques. Il présente également les caractéristiques de l'écriture arabe (alphabet, signes diacritiques, styles d'écriture, ...). IL expose aussi le problème de la segmentation

(implicite ou explicite) des mots en caractères, en discutant les différentes méthodes proposées. Ce travail catégorise les systèmes de reconnaissances, selon le type des primitives utilisées (locales ou globales) et selon le type de classifieur employé (RNA ou HMM).

III.5.2 M. S. Khorsheed, 2002

Dans [36] M. S. Khorsheed, présente une synthèse des techniques de prétraitement, de segmentation, d'extraction de primitives et de classification dans le domaine de la reconnaissance de l'écriture arabe. Il expose également les résultats de l'application de ces techniques sur des mots arabes imprimés et des textes arabes manuscrits. Il présente aussi une brève description de quatre OCR de textes arabes imprimés: TextPert, ICRA, OmniPage, et Al-Qari' al-Ali.

III.5.3 N. Ben Amara et F. Bouzlama, 2003

Les auteurs de [37], présentent un état de l'art des techniques de la classification dans le domaine de la reconnaissance de l'écriture arabe manuscrite en montrant l'avantage de la combinaison de plusieurs sources d'informations. Ils présentent alors trois niveaux de combinaisons:

- La combinaison en niveau de primitives motivée, d'une part par l'insuffisance d'une seule méthode à traiter la variation intra-classes, et d'autre part, par la sensibilité de quelques méthodes au bruit et aux distorsions.
- la combinaison en niveau de classification, est justifiée par l'absence d'un classifieur parfait en termes de généralisation et en termes de temps d'exécution.
- La combinaison de différentes familles de caractéristiques et de classifieurs.

Les auteurs présentent une variété de travaux, qui exploitent ces différents niveaux de combinaison. Ils concluent que la combinaison peut remédier aux problèmes de la variation des styles d'écriture et de la présence de bruit et de distorsion.

III.5.4 L. M. Lorigo et V. Govindaraju, 2006

Le travail de L. M. Lorigo et V. Govindaraju [38], est considéré comme étant le Survey le plus complet à l'heure actuelle dans le domaine de la reconnaissance de l'écriture arabe manuscrite. Il décrit la plupart des problèmes spécifiques liés à la reconnaissance de l'écriture arabe. Il inclut également un tour d'horizon de certaines bases de données: AHDB, CENPARMI et IFN/ENIT.

Il discute la chaîne de reconnaissance de l'écriture arabe manuscrite selon les étapes suivantes:

- Représentation (squelette, contour, ou pixels),
- Segmentation,
- Extractions de primitives (statistiques et structurelles),
- Classifieurs (les systèmes à base de règles, Réseaux de Neurones, Modèles de Markov Cachés, et approches hybrides),

Les auteurs présentent les six systèmes qui ont participé à la compétition internationale ICDAR 2005 [39].

En conclusion, les auteurs expliquent que des systèmes de reconnaissance de l'écriture arabe achèvent des résultats satisfaisants sur des applications pour lesquelles, soit le vocabulaire est limité, soit la forme des mots est relativement contrainte. L'avenir de la discipline se situe dans la capacité des systèmes à traiter de l'écriture libre sans contraintes.

III.5.6 M. T. PARVEZ et S.A. Mahmoud, 2013

Dans [40], M. T. PARVEZ et S.A. Mahmoud ont présenté un excellent Survey qui explore les développements récents dans le domaine de la reconnaissance de l'écriture arabe. Le Survey commence par une présentation des caractéristiques spécifiques de l'écriture arabe, suivi d'un modèle générale d'un système de reconnaissance hors ligne de l'écriture arabe manuscrite. Ils discutent également les bases de données les plus utilisées pour la reconnaissance de l'écriture arabe manuscrite. Ensuite ils présentent des travaux de recherches concernant le prétraitement, tels que la représentation des textes (squelettes et contours), la détection de la ligne de base, la correction de l'inclinaison, et l'étape de segmentation des mots en caractères. Les auteurs regroupent les travaux de recherche sur l'étape d'extraction de primitives en deux catégories: statistiques et structurelles. Ils discutent aussi un nombre important de travaux de recherches qui utilisent des différentes approches de classification, telle que les HMMs, les réseaux de neurones, SVM, KNN et la combinaison de classifieurs. Ils présentent aussi quelques techniques de post traitements utilisées dans le cadre de la reconnaissance de l'écriture arabe.

III.6 Différentes approches et systèmes existants

Nous présenterons les travaux effectués dans le domaine de la reconnaissance de l'écriture arabe manuscrite selon un regroupage par fonctionnalité. Nous aborderons ainsi les prétraitements, la segmentation, l'extraction de primitives, la classification et les post-traitements.

III.6.1 Prétraitements

La phase de prétraitement, est une étape cruciale dans tout système de reconnaissance d'écriture, son importance croit surtout pour le traitement du script arabe, qui a ces propres caractéristiques nécessitant le développement des techniques de prétraitement plus spécifiques à ce script ou au moins des adaptations appropriées des techniques existantes. Dans cette section, nous présentons les techniques de prétraitement développées spécialement pour l'écriture arabe, et nous citons aussi les différentes adaptations proposées pour quelques techniques standards.

III.6.1.1 Squelettisation

La squelettisation est une technique de changement de représentation très utilisée dans le domaine de la reconnaissance de l'écriture, notamment à des fins de l'extraction de caractéristiques morphologiques et à la réduction de la quantité d'informations à traiter.

Dans [41], les auteurs présentent un algorithme à base de l'algorithme de clustering fuzzy isodata pour la squelettisation des caractères arabes, appelé CBSA (Clustering Based Skeletonization Algorithm). Cet algorithme de squelettisation est l'un des premiers travaux qui traitent la particularité de l'écriture arabe.

Un autre algorithme de squelettisation à base de clusters est présenté en [42], ce travail emploie le réseau ART2 (Adaptive Resonance Theory 2), qui est un réseau de neurones d'auto-organisation pour trouver les clusters des caractères arabes. Le squelette du caractère est généré par traçage des centres de clusters et reliant les clusters adjacents par des lignes droites.

Dans [43], les auteurs introduisent un algorithme de squelettisation de l'écriture arabe manuscrite utilisant le codage couleur pour marquer, supprimer, et récupérer des pixels dans une image de texte arabe, ce qui permet l'obtention d'un bon squelette de cette image d'une façon très simple et efficace comparée aux algorithmes basés sur une morphologie complexe et des calculs mathématiques très gourmands en temps de calcul. D'après les auteurs, cet algorithme préserve très bien la forme de l'image originale et le squelette obtenu peut être efficacement incorporé dans un système de reconnaissance de l'écriture arabe manuscrite. Certains chercheurs ont extrait le squelette à l'aide d'une version modifiée de l'algorithme de Hilditch [17], utilisée pour la squelettisation de l'écriture latine.

III.6.1.2 l'estimation de la ligne de base

L'estimation de ligne de base, est un processus très important dans l'analyse des documents et des systèmes de reconnaissance de l'écriture manuscrite. Cette technique est largement utilisée dans la majorité des étapes de prétraitement, telle que la normalisation de texte, la correction de l'inclinaison, la segmentation en caractères, et aussi bien que dans l'extraction de

caractéristiques. Pour l'écriture arabe manuscrite, plusieurs méthodes ont été proposées pour cette procédure de prétraitement. La méthode la plus répandue pour la détection de la ligne de base, est la technique basée sur la projection horizontale [44], cette méthode est facile à implémenter et donne de bons résultats pour le manuscrit de bonne qualité ; où les mots sont assez longs et l'écriture est droite.

Une autre méthode très populaire pour l'estimation de la ligne de base dans les textes arabes, consiste à employer la transformée de Hough [45]. Cette transformation permet la conversion des points caractéristiques de l'image vers l'espace de Hough. Cet espace est représenté par un accumulateur qui contient les cellules d'accumulation. Ces dernières sont remplies par les votes de pixels appartenant à une droite donnée de paramètres (ρ, T) . Chaque maximum obtenu donne une indication sur les paramètres spécifiques de la droite détectée. La présence de maximum dans l'accumulateur permet le repérage de la ligne de base.

Des approches basées sur le squelette du mot ont prouvé leur efficacité pour estimer la ligne de base. La méthode développée en [46] comporte les étapes suivantes:

- L'extraction du squelette du mot par une approximation polygonale.
- Des primitives sont extraites sur chaque squelette d'une composante connexe, comme des mesures statistiques sur la boîte englobante (hauteur, largeur, aire, ratio largeur/hauteur) ou sur le squelette lui-même (longueur, nombre de nœuds, nombre d'arcs, degré des nœuds, longueur de chaque arc, angles, ...).
- Des primitives statistiques sont également calculées sur l'ensemble des squelettes pour une image donnée (moyenne et variance de la taille des boîtes englobantes, moyenne et variance de la longueur des squelettes, ...).
- Ces primitives permettent d'extraire d'autres caractéristiques de plus haut niveau, tels que des points diacritiques, des signes, des points caractéristiques, des structures simples ou des courbes qui correspondent aux descendants.
- La réduction de l'ensemble des primitives de l'image du mot par la sélection uniquement des composantes connexes pertinente pour l'estimation de la ligne de base. Cette sélection est faite en se basant sur des connaissances morphologiques de l'écriture arabe, par exemple les points diacritiques ou les Chadaahs qui ne sont pas des primitives pertinentes.
- Cette sélection permet le calcul d'une première estimation de la ligne de base. Cette dernière permet de définir une bande dont la hauteur correspond à 1/3 de la hauteur du mot, dans laquelle on va chercher des points supports plus précis.
- La ligne de base finale, est enfin obtenue par régression linéaire de tous ces points supports.

Les auteurs évaluent leur algorithme par rapport à l'annotation de la ligne de base des mots de la base IFN/ENIT. Ils estiment qu'un décalage de 7 pixels par rapport à l'annotation donne un résultat tout à fait satisfaisant, jusqu'à 15 pixels de décalage, la ligne de base reste acceptable. Snoussi-Maddouri [47], a présenté six différentes méthodes, qui sont respectivement : la projection des histogrammes horizontaux, une combinaison à base de contour et méthode de projection appelée : méthode Min-Max, une autre combinaison de la méthode Min-Max avec la position de quelques primitives structurelles, la minimisation d'entropie, méthode basée sur le squelette du mot, et enfin une technique basée sur la transformée de Hough.

III.6.1.3 la correction de l'inclinaison (slant correction)

Le but de cette étape de prétraitement, est d'éliminer n'importe quelle inclinaison dans chaque mot du texte, pour le rendre le plus indépendant possible des spécificités d'écriture du scripteur. La Figure(III.5) illustre le concept de correction de l'inclinaison des caractères arabes. L'idée de base dans la plupart des algorithmes de correction de l'inclinaison est de localiser des tracés quasi-verticaux dans le mot est d'estimer la moyenne de l'inclinaison du mot à partir de ces tracés ; Puis, l'inclinaison dans le mot est corrigée par l'application d'une transformation géométrique de type cisaillement de l'image du mot.

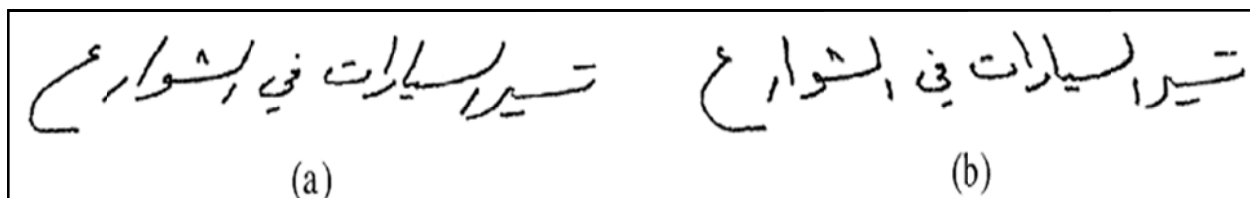


Figure III.5 : Le Concept de slant correction : (a) Image originale (b) Image après slant correction.

Dans [48], les auteurs ont utilisé la chaîne de Freeman pour détecter les tracés quasi-verticaux. Pour calculer l'angle d'inclinaison moyenne (α) du mot, tous les contours sont parcourus pour calculer l'angle d'inclinaison pour chaque tracé quasi-vertical. Ensuite, la moyenne pondérée (des angles d'inclinaisons multipliées par la hauteur euclidienne de la ligne verticale) des angles d'inclinaison individuels, est prise et la moyenne globale d'angle d'inclinaison du mot est déterminée.

Les histogrammes de projections verticales s'utilisent dans de nombreux travaux [49] de la même façon que pour l'estimation de la ligne de base du paragraphe précédent, mais cette fois dans des directions proches de la verticale. Certains chercheurs ont utilisés des méthodes basées sur la transformée de Hough pour corriger l'inclinaison des caractères manuscrits [50].

III.6.1.4 Normalisation de la taille des mots

Cette étape permet de ramener les images de mots à des tailles standards. Cette opération peut être indispensable pour certains types de systèmes, comme les réseaux de neurones et pour certaines méthodes d'extraction des primitives. La majorité des méthodes de normalisation sont développées principalement pour l'écriture latine, ensuite elle sont adaptées et appliquées pour la normalisation des mots arabes manuscrits. Parmi lesquelles on trouve la méthode basée sur le calcul de l'histogramme des transitions horizontales, qui consiste à détecter la position de la ligne de base, ensuite on doit ramener le centre du mot à cette position de la ligne de base détectée et enfin on redimensionne l'image résultante. [1]. Certains travaux ont utilisé l'algorithme de Srihari pour l'opération de normalisation des mots arabes [51].

III.6.1.5 Segmentation des mots en caractères

Quelques techniques de classification exigent la segmentation de mots en caractères, pseudo mots, graphèmes, ou d'autres unités plus petites. Ces unités qui constituent le mot sont utilisées pour l'extraction de caractéristiques ou la modélisation. Les chercheurs ont proposé beaucoup d'algorithmes pour la segmentation de mots. Les méthodes basées sur la projection verticale ou celles basées sur l'histogramme de l'image sont considérées comme l'une des méthodes classiques les plus utilisées pour segmenter le script arabe, suivi d'autres méthodes basées sur les contours [52], squelettes [53], réseaux de neurones [54], à base de graphes [55], et des techniques morphologiques [56]. Dans le reste de ce paragraphe on regroupe les techniques de segmentation en différentes catégories:

a. Segmentation basée sur les histogrammes de projection

L'idée de base de la segmentation à base de projection est, de prendre une projection horizontale de l'image de mot. La présence des minimums locaux dans l'histogramme de projection indique des emplacements possibles de points de segmentation. Roméo-Pakker[57] a utilisé des projections horizontales et verticales, une représentation en chaîne de code de Freeman, et des règles pour segmenter le texte manuscrit en caractères. Cependant, en raison de l'irrégularité dans l'écriture cursive, les algorithmes de segmentation à base de projection sont moins robustes et moins précis. Dans [58], Syiam propose un algorithme de segmentation en caractères des mots arabes manuscrits. La méthode combine l'histogramme de projection verticale avec une technique de clustering (K-Means). En plus beaucoup de techniques utilisent la projection verticale comme un prétraitement pour segmenter les documents en lignes d'écritures et lignes en mots.

b. Segmentation basée sur le traçage de contours

Plusieurs techniques ont été développées en utilisant le traçage de contours pour résoudre le problème de la segmentation des mots en caractères. Dans [59], les auteurs proposent une méthode de segmentation basée sur le contour supérieure de chaque mot. Des minima locaux sont détectés afin de trouver les points candidats de segmentation.

Ces points sont analysés avec un automate qui considère la forme du mot pour la détermination des points définitifs de segmentation des mots en caractères. Dans [60], un algorithme pour segmenter les mots arabes manuscrits en graphèmes (caractères ou parties des caractères) est proposé. Cet algorithme, est basé sur l'analyse des minimums locaux des contours supérieurs des mots pour la détermination des zones d'intérêt de segmentation. Les points candidats de segmentation sont recherchés dans ces zones en fonction des trois règles suivantes :

- Les points candidats situés au-dessus d'une boucle sont éliminés,
- L'épaisseur du tracé à l'endroit du point candidat doit être inférieure à un seuil.
- Si plusieurs points candidats d'une même zone sont voisins, le point choisi est le plus proche de la ligne de base.

En [61], ils proposent un algorithme de segmentation nommé ACSA (Arabic Character segmentation Algorithm) composé de cinq étapes. Cet algorithme est basé sur l'exploitation des caractéristiques topologiques de l'écriture arabe, pour la mise en œuvre d'un ensemble de règles morphologiques. Ces règles sont utilisées pour valider ou non la sélection des points de segmentation probables qui se situent au niveau des minimums locaux des contours extérieurs bas.

c. Segmentation basée sur le squelette

La squelettisation est une technique essentielle qui pourrait aider à résoudre le problème de la segmentation en caractères. Cependant la combiner avec d'autres techniques serait essentiel pour garantir une meilleure segmentation. Dans [53], les auteurs proposent un algorithme de segmentation de l'écriture arabe manuscrite, basé sur le contour et le squelette du mot.

Cet algorithme est résumé comme suit:

Entrées: les Segments de l'image squelette, la chaîne de codes des segments

- pour chaque segment dans le squelette de l'image:

a- pour chaque point d'intersection du segment:

- i- Trouvez le code de chaîne du contour correspondant au squelette du segment courant.
- ii- Construire la distance map (entre le point d'intersection et tout le code de la chaîne des points de contour), en commençant à partir du point de chaîne de code le plus éloigné.

- iii- Trouvez les trois premiers pics inférieurs.
- iv- Vérifier si le troisième pic appliqué pour la troisième frontière de segment.
- v- Tracez des lignes du point d'intersection aux points des pics appliqués.

b- Colorier les nouveaux segments.

En [62], ont proposé un algorithme de segmentation à base des règles morphologiques qui segmentent les mots arabes cursifs en graphèmes par l'utilisation de points de caractéristiques spéciaux extraits du squelette du mot. Ces points caractéristiques comprennent des points d'extrémité, des points de branchement, et des points de croisement qui sont identifiés en examinant les huit voisinages de chaque pixel du squelette.

Une autre approche basée sur la squelettisation est proposée en [63], inspirée d'un algorithme heuristique proposé en [64] pour segmenter les mots arabes manuscrits, mais au lieu d'utiliser seulement l'ensemble des minima locaux des contours comme des candidats de points de segmentation (SP), ils ont inclus dans SP toutes les colonnes des pixels en tant que candidats. Ensuite, une opération d'élection basée sur des règles heuristique concernant les points caractéristiques (points d'extrémité, points de branchement, et points de boucles) est effectuée, Afin d'exclure des candidats improbables des points de segmentation (SP). La Figure III.6, montre des échantillons prélevés sur IESK-ARDB [65] ainsi que la base de données IFN / ENIT, segmentés selon cette approche.

Binary	Thinned	Segmented	Binary	Thinned	Segmented

Figure III.6 : Résultats de la segmentation : (A) des images de la base IESK-arDB. (B) des images de la base IFN/ENIT.

d. Segmentation validée par la reconnaissance

L'algorithme de Ding et Liu [66], consiste à obtenir une sur-segmentation du corps principal du mot arabe, puis sélectionner le meilleur chemin de segmentation. Les points candidats de segmentation (des graphèmes) sont des points situés sur la ligne des bases, des minimums locaux sur le contour supérieur et des points des contours supérieurs et inférieurs qui sont proches les uns des autres. Les points de segmentation qui sont trop proches sont éliminés. Ces graphèmes du corps principal plus les signes diacritiques au-dessus et au-dessous du corps principal sont mis dans des listes. Toutes les combinaisons de ces graphèmes et signes diacritiques, sont examinées et la meilleure combinaison est choisie comme étant la segmentation la plus probable du mot. Toutes les segmentations possibles du mot sont évaluées en optimisant une fonction objective.

Al Hamad et Abu Zitar [67] proposent un algorithme de segmentation et une stratégie de validation pour les mots arabes manuscrits. Leur algorithme fonctionne en trois étapes. D'abord une sur-segmentation obtenue d'un histogramme vertical modifié de l'image du mot amincie. Ces points de segmentations initiales sont ensuite validés par un schéma de validation des points de segmentation basée sur un réseau de neurones.

Pour chaque point de segmentation trois zones sont localisées, une zone de segmentation (SA) de petite dimension située autour du point de segmentation; la zone située entre le courant et le point de segmentation précédant (right character: RC), et la zone centrée sur le point de segmentation actuelle (appelée le caractère central (CC) avec une demi-largeur du RC).

La technique de base dans le travail de [68], est basée sur la validation par des réseaux de neurones. Il se compose de deux étapes, la première est un segmenteur heuristique (AHS) utilisé pour trouver les points de segmentation potentiels (PSP). La deuxième étape consiste à utiliser deux réseaux de neurones de rétro-propagation du gradient. Cette étape a pour but d'examiner tous les PSPs pour se décider lequel est invalide pour l'enlever et celui qui est valable pour le garder comme point de segmentation correct. Le premier réseau de neurones est entraîné par 120 caractéristiques extraites par la méthode MDF, qui décrivent les formes de 60 caractères arabes écrits par 20 scripteurs différents. Le deuxième réseau est entraîné par les caractéristiques des zones de segmentation (SA) correcte et incorrecte. La sortie du premier réseau contient 61 valeurs (60 valeurs pour les 60 formes correctes des caractères et une valeur pour le rejet), et une seule sortie pour le deuxième réseau (plus de 0.5 pour les SA correctes et moins de 0.5 pour les incorrects).

III.6.2 Extraction de primitives

Les méthodes d'extractions de primitives dans le domaine de la reconnaissance de l'écriture arabe manuscrite se différent selon la nature des caractéristiques employées. Généralement, ces dernières peuvent être classées en trois grandes familles: Caractéristiques statistiques, structurelles, et transformation globales. Nous présentons dans le tableau suivant les principales caractéristiques extraites dans les systèmes de la reconnaissance hors ligne de mots arabes manuscrits.

Système	Année	Caractéristiques	Type
Miled [69]	1998	Une séquence chronologique de droite à gauche d'indices visuels codés.	Structurelles
Khorsheed[70]	1999	les points caractéristiques, les boucles, et 08 primitives de segment.	Structurelles
Benouareth [71]	2000	caractéristiques perceptuelles (le nombre de sous-mots, ascendants, descendants, boucles, points diacritiques et zigzags ou Hamza).	Structurelles
Azizi [72]	2002	Zoning + ascendants, descendants, boucles et points diacritiques+ Moments de Hu	Statistiques + structurelles
Khorsheed[73]	2003	Points diacritiques, points extrêmes, points de jonctions, et boucles	Structurelles
Burrow [50]	2004	Les moments de Zernike	Transformations globales
Al-Maadeed [74]	2004	Les boucles et leurs positions d'intersection (haut, bas, gauche droite), les points diacritiques et leur position, les points extrêmes, les points de jonction, les moments invariants, et la densité de l'histogramme horizontal	Statistiques + structurelles+transformations globales
Farah [75]	2005	-Les ascendants, les descendants, et les boucles. -La technique de Zoning	Statistiques + structurelles
Gasmi [76]	2005	-Le nombre des pseudo-mots, le nombre d'ascendants, le nombre de descendants, les boucles et les points diacritiques. - La densité des pixels	Statistiques + structurelles
Farah [77]	2006	-La chaîne de code de Freeman. nombre d'ascendants, nombre de descendants, boucles et points diacritiques.	Statistiques + structurelles
Al-Abed et Märgner [78]	2007	Primitives de directions calculées à partir du squelette du mot.	Structurelles

Nouar [79]	2008	-Nombre de descendants, nombre d'ascendants, nombre de boucles, ...etc. -La technique de Zoning	Statistiques + structurelles
Benouareth [80]	2008	- La technique de Zoning - Points caractéristiques - Concavités	Statistiques + structurelles
Broumandnia[81]	2008	La transformée en ondelettes	Transformations globales
Zavorin[82]	2008	Boucles, points diacritiques, points extrêmes et de jonctions	Structurelles
AlKhateeb[83]	2009	Les coefficients de DCT et les moments de Hu	Transformations globales
Rami alhadj[84]	2009	Distribution de pixels et concavités	Statistiques + structurelles
Kessentini[85]	2010	Directions, contours, et densités de pixels	Statistiques+ structurelles
Chen [86]	2010	Gabor	Transformations globales
Nemmour[87]	2011	La transformée de Ridglet	Transformations globales
Alalshekmubarak[88]	2012	La technique de zoning	Structurelles
Nemouchi[89]	2012	- Primitives structurelles - Moments de Zernike - technique de Zoning - Chaîne de Freeman	Statistiques + structurelles+transformations globales
Chergui [90]	2012	- Moments de Tchebichef - Moments de Zernike - Moments de Hu	Transformations globales
Abdel Azeem [91]	2013	- Concavités primitives - Gradient primitives	Structurelles+ Transformations globales
EI qacimy [92]	2015	Coefficients de DCT	Transformations globales

Tableau III.4 : Primitives utilisées dans les systèmes de reconnaissance de mots arabes manuscrits.

III.6.3 Reconnaissance des mots

Classiquement deux types d'approches de reconnaissances des mots sont utilisés, les approches analytiques et les approches globales. Ces approches se déclinent également en fonction des connaissances sur la taille du vocabulaire, et sur la variabilité des formes à reconnaître (nombre de scripteurs).

III.6.3.1 Approches Globales

Dans cette section nous regroupons les travaux de recherches selon le type de classifieur employé pour la reconnaissance des mots arabes manuscrits.

III.6.3.1.1 par modèles de Markov cachés (HMM)

Vu le grand succès des HMMs dans le domaine de la reconnaissance de la parole et la reconnaissance de l'écriture latine, plusieurs chercheurs ont les utilisés pour la reconnaissance des mots arabes manuscrits. Dans [93], les auteurs proposent un système de reconnaissance de mots arabes manuscrits basé sur une segmentation implicite et une modélisation markovienne des caractères. Pour réduire la variabilité des styles d'écriture, l'image du mot passe d'abord par une étape de squelettisation, puis une normalisation horizontale pour obtenir une largeur moyenne constante de caractères et une normalisation verticale pour uniformiser la hauteur des ascendants et descendants.

L'extraction des caractéristiques est directement basée sur la représentation image en utilisant les valeurs des pixels comme des primitives de base. Une fenêtre rectangulaire est glissée de droite à gauche en respectant l'orientation de l'écriture arabe à travers l'image de niveau de gris normalisée, pour générer le vecteur de caractéristique. La Figure III.7 donne un exemple d'extraction de caractéristiques par une fenêtre glissante. Pour tester la fiabilité du système proposé, la base de données IFN/ENIT des mots arabes manuscrits à été utilisée.

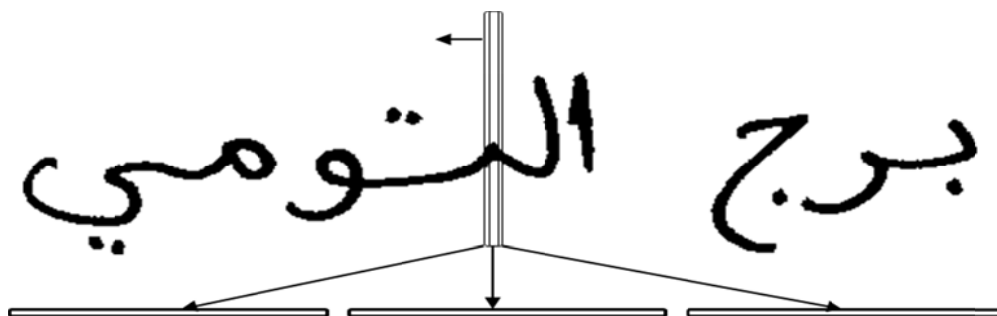


Figure III.7 : Extraction des primitives avec une fenêtre glissante de trois colonnes [93].

Parmi les systèmes de reconnaissance qui ont obtenus des taux de reconnaissance élevés, basés sur la technique de segmentation implicite par fenêtre glissante, on trouve le travail d'El-Hajj [94], qui propose une méthode avancée qui prend en considération les inclinaisons possibles dans l'écriture. Cette méthode est composée de deux étapes, l'une de reconnaissance et l'autre de combinaison en post-traitement. L'étape de reconnaissance est basée sur l'extraction des vecteurs de caractéristiques toujours selon la technique des fenêtres glissantes mais avec des inclinaisons différentes, permettant d'obtenir dans la phase de classification, et pour chaque inclinaison, un ensemble de mots candidats comme résultat.

L'étape de combinaison permet de fusionner en post-traitement, les résultats obtenus pour produire le candidat le plus approprié [95]. Le système a été testé sur la base IFN/ENIT de noms des villes tunisiennes.

Khorsheed [73], a présenté une méthode globale pour la reconnaissance hors ligne des mots arabes manuscrits, en utilisant un HMM 1D avec des caractéristiques structurelle, extraites de l'image du mot. Le HMM est composé de multiples modèles de caractères, où chaque modèle est un HMM de type droite à gauche, qui représente une lettre de l'alphabet arabe. Après le prétraitement, le graphe du squelette du mot est décomposé en une suite de segments dans l'ordre dans lequel le mot est écrit après une étape d'approximation linéaire. La méthode proposée est évaluée en utilisant des échantillons extraits d'un manuscrit ancien.

Benouareth [80], montre expérimentalement que les D-HMM (HMM à durée d'état explicite) peuvent significativement améliorer la capacité exigeante du SC-HMMs (HMM semi continu) pour traiter des tâches de reconnaissance de formes très difficiles, comme la reconnaissance des mots arabes manuscrits. Par ces expériences, il montre que la segmentation en bandes verticales non-uniforme (le découpage s'appuie sur l'analyse de l'histogramme de projection vertical) donne de meilleurs résultats que la segmentation en bandes uniformes (largeur de bande fixe). Ses expériences suggèrent que la loi gamma se comporte mieux que la loi gaussienne et la loi de poisson pour modéliser la durée dans les états des HMM sur une tâche de reconnaissance d'écriture arabe manuscrite[42].

Kessentini [85], propose un système de reconnaissance hors-ligne de mots arabes manuscrits multi-scripts et met en évidence l'apport d'une modélisation par des modèles de Markov cachés multi-flux. Le système proposé, opère indépendamment de la nature du script. Pour cela, il procède sans segmentation en combinant des caractéristiques bas niveaux basés sur les directions et les densités des pixels. Des résultats significatifs sont obtenus, en ne combinant que 2 flux sur la base de référence de mots arabes IFN/ENIT.

Une autre approche basée sur les HMMs est proposée par Dong Xiang [96], consiste à modéliser les caractères par des HMMS, ensuite les modèles des mots sont obtenus par concaténation des modèles des caractères qui les constituent. Deux types de caractéristiques: les densités de pixels noirs et les concavités, sont extraites de l'image du mot en utilisant la technique de fenêtre glissante de droite à gauche. L'apprentissage est réalisé à l'aide de la méthode de Baum-Welch et la décision finale et prise par l'utilisation de l'algorithme de Viterbi.

III.6.3.1.2 Par réseaux de neurones (RNN)

Beaucoup de chercheurs, ont utilisé les réseaux de neurones pour développer des systèmes de reconnaissance hors ligne des mots arabes manuscrits. Dans [97], Alm'addeed a proposé un système de reconnaissance de mots arabes manuscrits en utilisant les réseaux de neurones. La première étape est une étape de prétraitement, qui consiste à enlever un peu de variation dans les images qui n'affectent pas l'identité du mot. La deuxième étape, est l'extraction de caractéristiques. Ce travail traite des caractéristiques globales (des ascendants, descendants, des signes diacritiques, nombre des boucles, nombre de segments, points diacritiques au dessus, et points diacritiques au dessous) qui sont extraites de l'image du mot. Le réseau de neurones utilise ces caractéristiques globales pour reconnaître le mot arabe. Ce système est évalué sur une base de données qui contient 70 mots arabes différents écrits par 100 scripteurs différents.

Dans [98], on propose un système à base de réseaux de neurones de type Perceptron Multicouches, pour assurer le tri des chèques arabes et la reconnaissance des noms de villes tunisiennes. Le système proposé, s'articule autour de quatre modules distincts ; un module de codage des entrées du réseau c'est à dire une étape d'extraction de primitives structurelles, un module d'apprentissage par l'algorithme de la rétro-propagation de gradient, un module de vérification de l'apprentissage dite "généralisation sur apprentissage" et un module de reconnaissance ou dit de généralisation. L'évaluation expérimentale de ce système a été réalisée en utilisant deux bases de données : une base de données des chèques et une base de données de noms de villes tunisiennes (IFN/ENIT).

Une autre approche basée sur les réseaux de neurones de type perceptron multicouches est décrite en [83], après une étape de prétraitement, la transformée en Cosinus discrète (DCT) est utilisée pour décrire le mot arabe manuscrit, et extraire un vecteur de primitives, qui sera utilisé ensuite pour les étapes d'apprentissage et de reconnaissance du réseau de neurones. Pour évaluer les performances de ce système de reconnaissance, les expériences sont effectuées sur la base des noms de villes tunisiennes (IFN/ENIT).

Les méthodes exploitant la théorie de l'apprentissage automatique de réseaux de neurones profonds (Deep Learning) ont prouvé leur efficacité pour la reconnaissance de l'écriture manuscrite. Dans [99] on a proposé les DBN (Deep Belief Networks) pour capturer la distribution des données en représentant les données à différents niveaux d'abstraction en utilisant plusieurs couches cachées. Cette méthode, utilise des caractéristiques plus bas niveaux (pixel) et forme une représentation distribuée d'elles, qui sera capable de discriminer à des niveaux plus élevés tels que les bords, contours et éventuellement, des mots du texte manuscrit.

Les progrès récents dans le domaine de l'apprentissage automatique, ont conduit à des réseaux récurrents multidimensionnels MDLSTM (Long Short Term Memory), qui sont capables d'apprendre les caractéristiques directement, à partir de la représentation en niveau pixels des images de mots sans l'application des approches classiques d'extraction de primitives et sans l'étape ardue de la segmentation en caractères et même sans aucune opération de prétraitement. Alex Graves [100], présente un système très performant pour la reconnaissance des mots arabes manuscrits, qui est le vainqueur de la compétition internationale pour la reconnaissance de l'écriture manuscrite ICDAR 2009. L'idée la plus importante du classifieur MDLSTM, est d'associer à chaque dimension du problème deux réseaux de neurones récurrents. L'un des réseaux parcourt la dimension dans une direction, et le second dans l'autre direction. Ainsi dans une image, il lie la dimension horizontale et la dimension verticale. Un second concept important de cette approche est la hiérarchisation du réseau. Cette hiérarchisation permet d'abstraire l'information à différents niveaux, et la transformation d'une image 2D en un 1D des étiquettes de séquences en utilisant l'algorithme de CTC (Connectionist Temporal Classification). Cette couche de sortie comporte $M+1$ neurones, où chaque neurone représente une classe du problème (une lettre dans le cadre de la reconnaissance de l'écriture arabe) et le dernier est un symbole joker, représentant une absence de décision ou un blanc. Cela permet par exemple d'éviter de prendre des mauvaises décisions sur des cas complexes.

III.6.3.1.3 Par les (SVMs)

Le SVM (support vector machine), est une méthode de classification particulièrement bien adaptée pour traiter des données de très haute dimension, telles que les textes et les images. Ces avantages ont poussé les chercheurs à les employer pour la reconnaissance de l'écriture arabe manuscrite. Hassiba Nemour [87], propose un système de reconnaissance des mots arabes manuscrits basé sur la combinaison de la transformée de Ridgelet et les SVMs. La transformée de Ridgelet est utilisée pour générer des primitives pertinentes des mots arabes manuscrits, tandis que l'étape de classification est basée sur l'implémentation un-contre-tous du SVM multi classes. Des tests expérimentaux ont été effectués sur un vocabulaire de vingt-quatre mots extraits de la base de données des noms des villes tunisiennes IFN/ENIT. Une autre méthode de reconnaissance des mots arabes manuscrits, est proposée en [88]. Le système applique une méthode d'extraction de primitives très simple basée sur la technique de Zoning et un classifieur basée sur un SVM avec un noyau polynôme normalisé. La base de données, IFN / ENIT a été utilisée pour évaluer sa performance en comparaison aux deux systèmes décrits dans d'autres études. D'après les auteurs le système proposé exige moins de traitement de calcul dans l'étape d'extraction de caractéristique et permet d'obtenir des résultats satisfaisantes.

III. 7 Compétitions sur la reconnaissance de l'écriture arabe manuscrite

III.7.1 Compétition ICDAR2005

La première compétition pour les systèmes de reconnaissance du manuscrit arabe a été basée sur les images de la base de données IFN/ENIT, et les résultats de cette compétition ont été présentés à la conférence internationale de l'analyse et de la reconnaissance du document (ICDAR) 2005 [39]. Les cinq systèmes soumis ont utilisé pour l'apprentissage les quatre sous ensembles {a, b, c, d} de la base IFN/ENIT, et ils sont évalués par les organisateurs de la compétition, sur un sous ensemble (e) (le sous ensemble de test e est inconnu pour tous les participants, actuellement intégré dans la version v2.0p2e de la base de données IFN/ENIT).

Dans le tableau III.5, les cinq systèmes participants à la première compétition sont présentés, avec les méthodes de reconnaissance utilisées et les taux de reconnaissance pour le sous ensemble de test e. Le système UOB de Ramy El-Hajj [94], est le système qui a remporté la compétition ICDAR 2005. Cette première compétition a été une motivation pour les organisateurs et pour les participants pour travailler à améliorer les systèmes de reconnaissances de manuscrits arabes.

Système	Description	Top1	Top5	Top 10
ICRA	NN (Two-TierApproach)	65.74	83.95	87.75
SHOCRAN	Confidentiel	35.70	51.62	51.62
TH-OCR	Méthodes statistiques	29.62	43.96	50.14
UOB	HMM	75.93	87.99	90.88
REAM	HMMs planaires	15.36	18.52	19.86
ARAB-IFN	SC-1D-HMM	74.69	87.07	89.77

Tableau III.5 : Résultats de reconnaissance des systèmes participants à la compétition ICDAR 2005.

III.7.2 Compétition ICDAR 2007 [101]

La deuxième compétition ICDAR 2007 a été également organisée avec le même principe que celui de la première. Deux changements ont été faits, le premier concerne les ensembles d'apprentissage et de test utilisés et le deuxième, concerne les méthodes d'évaluation (l'ajout de l'évaluation du temps de calcul). Dans cette compétition, on a demandé aux participants d'utiliser les sous ensembles a, b, c, d et e pour l'apprentissage, et le test est effectué sur deux nouveaux sous ensembles :

- Le sous ensemble {f} collecté à partir d'une contribution des scripteurs tunisiens qui n'ont pas

participé à la collection des sous ensembles {a, b, c, d, e}. Ce sous ensemble est divisé en trois sous ensembles {fa, ff, fg} afin de rapprocher leur distribution en mots, à celle de l'ensemble d'apprentissage. Deux autres sous ensembles {t, t1} sont générés à partir de l'ensemble {f}, ils sont utilisés pour l'évaluation du temps de calcul des différents systèmes.

– Le sous ensemble {s} collecté au niveau de l'université de Sharjah à l'UAE, est utilisé pour étudier l'aptitude des systèmes à reconnaître l'écriture issue des autres régions (leur pouvoir de généralisation).

Les résultats ont été présentés à la conférence internationale de l'analyse et reconnaissance du document (ICDAR) 2007. Dans cette compétition, huit groupes différents ont soumis un total de 14 systèmes de reconnaissance (quelques groupes ont participé avec plus d'un système), dont deux participants ont figuré déjà lors de la compétition 2005 : ICRA de A. Abdulkadr [102] et UOB-ENST de Ramy El-Hajj [94]. Pour la première fois, on a constaté l'intérêt de groupes de recherches industriels, présentant 43% des participants. Cette concurrence, a montré une amélioration considérable des taux de reconnaissance, qui ont atteint les 87% comparé aux taux obtenus lors de la première compétition, qui était de 76%. Le tableau III.6, montre les résultats des différentes évaluations menées sur les 14 systèmes, cette évaluation inclut également une mesure du temps de calcul.

SIEMENS (ID 08), est le système qui a présenté le meilleur taux de reconnaissance. Il combine trois méthodes d'extraction de primitives (qui ne sont pas indiquées dans le manuscrit) et s'appuie sur un reconnaiseur de lettres à base de HMM. Ce système est le résultat d'une série d'adaptations appliquées sur un système de reconnaissance de l'écriture manuscrite latine.

En termes de temps de calcul, le système CEDAR (ID 05), a obtenu les meilleurs résultats.

Système	ID	Set d	Set e	Set f _a	Set f _r	Set f _g	Set f			Set s			Temps (ms)	
		Top1	Top1	Top1	Top1	Top1	Top1	Top5	Top10	Top1	Top5	Top10	Set t	Set t
MITRE	01	66.34	64.89	62.61	63.79	63.90	61.70	81.61	85.69	49.91	70.50	76.48	10886.4	15815.0
CACI	02	40.45	37.73	12.21	12.71	13.26	11.95	26.44	34.51	8.01	17.17	23.78	852.140	1171.25
	03	70.62	68.62	15.95	17.62	17.92	15.79	22.33	21.34	14.24	19.39	20.53	882.625	1040.46
	04	48.68	44.04	14.64	15.42	16.09	14.28	29.88	37.91	10.68	21.74	30.20	888.171	1099.53
CEDAR	05	68.07	57.37	59.84	60.67	60.86	59.01	78.76	83.70	41.32	61.98	69.87	34.171	41.71
MIE	06	93.63	86.67	84.38	85.21	85.56	83.34	91.67	93.48	68.40	80.93	83.73	188.439	210.55
SIEMENS	07	91.23	84.27	83.90	84.84	84.97	82.77	92.37	93.92	68.09	81.70	85.19	39.218	61.87
	08	94.58	87.77	88.41	89.26	89.72	87.22	94.05	95.42	73.94	85.44	88.18	109.406	125.31
UOB-ENST	09	90.02	81.80	80.58	82.43	82.67	79.10	87.69	90.21	64.97	78.39	82.20	680.781	754.06
	10	92.12	83.52	83.08	84.51	84.56	81.65	90.81	92.35	69.61	83.79	85.89	2174.97	2447.5
	11	92.38	83.92	83.39	84.93	85.18	81.93	91.20	92.76	69.93	84.11	87.03	2172.55	2424.47
	12	93.32	85.13	83.23	84.79	85.29	81.81	88.71	90.40	70.57	79.85	83.34	2191.23	2430.78
ICRA	13	88.33	83.87	82.74	83.68	84.14	90.07	90.07	92.15	72.22	82.84	86.27	359.687	402.34
PARIS V	14	89.90	80.24	81.36	83.45	83.82	91.09	91.09	91.09	64.38	78.12	82.13	383.078	472.18

Tableau III.6 : Résultats de reconnaissance des systèmes participants à la compétition ICDAR 2007.

III.7.3 Compétition ICDAR 2009 [103]

En 2009, la troisième compétition pour les systèmes de reconnaissance du manuscrit arabe a été organisée avec le même principe que celui de la deuxième. Cette compétition, a comparé 17 systèmes de reconnaissance soumis par 7 participants :

- Quatre systèmes à base de réseau de neurones : trois systèmes soumis par MDLSTM [100], qui procèdent directement sur l'image en pixels, et le système LSTS présenté par Snoussi-Maddouri, qui combine une vision globale et locale du mot à base des réseaux de neurones transparents.
- Menasri a soumis 3 systèmes (A2iA) à base d'une approche hybride PMC/HMM (ce système 'A2iA' figurait déjà lors de la compétition 2005).
- les autres systèmes sont à base des HMMs:
- le système REGIM soumis par Abdelkarim ElBaati. Ce système est basé sur la restauration de l'ordre temporel du tracé hors d'un mot.
- Le système UOB-ENST soumis par Chafic Mokbel et Ramy Al-Hajj, qui a participé aussi aux compétitions ICDAR 2005 et 2007. En variante 2009, quatre variantes du système l'UOB-ENST à base des HMMS et de fenêtres glissantes inclinées ont été présentées.
- Le système LITIS-MIRACL soumis par Yousri Kessentini, est à base des modèles de Markov cachés multi-Flux.
- Quatre variantes du système RWTH-OCR on été soumis par Philippe Dreuw.

Le tableau III.7 présente la comparaison des résultats de ces différents systèmes. Le système 11 (MDLSTM) a remporté la compétition en 2009 par l'obtention du meilleur taux de reconnaissance. Le système 10, soumis par le même groupe MDLSTM, a obtenu les meilleurs résultats en termes de temps de calcul.

Système	ID	Set d	Set e	Set f _a	Set f _r	Set f _g	Set f			Set s			Temps (ms)	
		Top1	Top1	Top1	Top1	Top1	Top1	Top5	Top10	Top1	Top5	Top10	Set t	Set t
UOB-ENST	01	92.52	85.38	83.57	84.77	85.09	82.07	89.74	91.22	69.99	81.44	84.68	812.69	841.25
	02	89.06	81.85	79.49	80.90	81.11	78.16	89.06	91.88	65.61	81.44	85.95	2365.48	2755.01
	03	89.84	83.52	80.89	82.15	82.17	79.55	90.60	92.16	67.83	83.47	86.65	2236.58	2754.08
	04	92.59	86.28	85.42	86.96	87.21	83.98	91.85	93.00	72.28	85.19	87.92	2154.48	2651.57
REGIM	05	79.52	63.53	58.81	59.27	60.42	57.93	73.43	78.10	49.33	65.10	71.14	1564.75	1712.15
Ai2A	06	93.90	87.25	86.73	88.54	89.36	85.58	92.57	94.12	70.44	82.01	84.87	1056.98	956.82
	07	94.92	82.21	83.53	84.86	84.67	82.21	91.24	94.47	66.45	80.52	83.13	519.61	1616.82
	08	97.02	91.68	90.66	91.92	92.31	89.42	95.33	95.94	76.66	88.01	90.28	2583.64	1585.49
MDLTM	09	99.72	98.64	92.59	93.79	94.22	91.43	96.11	96.61	78.83	87.98	90.40	115.24	122.97
	10	99.60	97.60	92.58	94.03	94.40	91.37	96.24	96.61	78.89	88.49	90.27	114.61	122.05
	11	99.94	99.44	94.68	95.65	96.02	93.37	96.46	96.77	81.06	88.94	90.72	371.85	467.07
RWTH-OCR	12	99.91	98.71	86.97	88.08	87.98	85.51	93.32	94.61	71.33	83.66	86.52	17845.12	18641.93
	13	99.79	98.29	87.17	88.63	88.68	85.69	93.36	94.72	72.54	83.47	86.78	17845.12	18641.93
	14	99.79	98.29	87.17	88.63	88.68	85.69	93.36	94.72	72.54	83.47	86.78	17845.12	18641.93
	15	96.72	91.25	86.97	88.08	87.98	83.90	-	-	65.99	-	-	542.12	560.44
LITIS-MIRACL	16	93.04	85.46	83.29	84.51	84.35	82.09	90.27	92.37	74.51	86.14	88.87	143269.81	145157.23
LSTS	17	18.58	14.75	15.34	16.00	15.65	15.05	29.58	35.76	11.76	23.33	29.62	612.56	685.42

Tableau III. 7 : Résultats de reconnaissance des systèmes participants à la compétition ICDAR 2009.

III.7.4 Compétition ICFHR 2010

La quatrième compétition pour les systèmes de reconnaissance du manuscrit arabe, a été basée toujours sur les images de la base de données IFN/ENIT, et les résultats de cette compétition ont été présentés à la conférence internationale sur les Frontières de la Reconnaissance d'Écriture manuscrite (ICFHR 2010) [104]. Cette compétition a comparé 6 systèmes soumis par 4 groupes de recherches différents (quelques groupes ont participé avec plus d'un système). Il faut noter que tous les systèmes soumis sont à base des modèles de Markov cachés (HMM).

A. UPV-PRHLT: Ce système est soumis par Adrià Giménez de l'université de Politècnia de Valencia (UPV), Valencia, Spain. Ce groupe de recherche a présenté deux systèmes basés sur les HMMs de Bernoulli [105]. La seule différence entre les deux variantes de ce système, est la façon dont la fenêtre glissante est appliquée.

Les systèmes UPV-BHMM sont basés sur une fenêtre glissante d'une largeur adéquate pour mieux capturer le contexte d'image à chaque position horizontale de l'image de mot.

B. REGIM: ce groupe de recherche a participé par un système soumis par Mahdi Hamdani et Adel M. Alimi. Ce système est basé sur les HMMS et est une version améliorée du travail présenté en [106]. Cette amélioration consiste en une optimisation de l'architecture des HMMs (nombre des états) par l'utilisation de l'algorithme Particle Swarm Optimization (PSO).

Les caractéristiques utilisées, sont basées sur la transformation de la représentation en pixels, extraites des images normalisées à l'aide de la transformée de Karhunen-Loève.

C. CUBS-AMA: Le système CUBS-AMA a été présenté par Safwan Wshah et Venu Govindaraju. Ce système est un mélange de gaussiennes HMM basé sur la boîte à outils HTK (la boîte à outils des Modèles de Markov cachés). Les séquences des vecteurs de caractéristiques basées sur le Gradient, la structure, et les concavités, sont créées en utilisant une fenêtre glissante. Le système obtient de meilleurs résultats, après la réduction des primitives et l'application de différents états pour chaque modèle.

D. RWTH-OCR: Philippe Dreuw, Christian Plahl, Patrick Doetsch, et Hermann Ney de l'université de RWTHAachen, Allemagne, ont soumis les deux systèmes RWTH-OCR. Ces systèmes sont basés sur les modèles de Markov cachés (HMM). Le tableau III.8 présente la comparaison des résultats de ces différents systèmes. Le système UPV-PRHLT 2 a remporté la compétition 2010 par l'obtention du meilleur taux de reconnaissance. Le système CUBS-AMA, a obtenu les meilleurs résultats en termes de temps de calcul.

G-ID	S-ID	Approach	set d	set e	set f _a	set f _r	set f _g	set f			set s		
			top1	top 1	top 1	top 1	top 1	top 1	top 5	top 10	top 1	top 5	top 10
UPV PRHLT	1	HMM	98.90	96.04	89.43	90.76	90.63	87.91	93.71	94.72	78.45	87.35	90.40
	2		99.38	98.03	93.46	94.30	94.02	92.20	95.72	96.29	84.62	91.42	93.32
REGIM	3	HMM	94.12	86.62	80.60	81.36	81.52	79.03	89.35	91.34	68.44	81.99	84.98
CUBS-AMA	4	HMM	89.97	80.80	81.75	83.35	83.55	80.32	88.26	88.96	67.90	78.58	79.87
RWTH-OCR	5	HMM	99.99	99.77	92.27	93.12	93.46	90.88	95.35	96.04	81.06	90.08	92.63
	6		99.66	98.84	92.35	93.35	93.55	90.94	95.31	96.00	80.29	89.83	91.80

Tableau III. 8 : Résultats de reconnaissance des systèmes participants à la compétition ICFHR 2010.

III.7.5 Compétition ICDAR 2011

La cinquième compétition pour les systèmes de reconnaissance du manuscrit arabe, a été basée toujours sur les images de la base de données IFN/ENIT, et les résultats de cette compétition ont été présentés à la conférence internationale de l'analyse et reconnaissance du document (ICDAR) 2011 [107]. Cette compétition a comparé les taux de reconnaissance des 4 systèmes soumis par 4 participants.

A. JU-OCR : ce système est soumis par Gheith Abandah et Fuad Jamour, il est destiné à reconnaître un vocabulaire illimité et est basé sur une segmentation explicite en graphèmes.

B. CENPARMI: ce système est soumis par muna Khayyat, Louisa Lam et Ching Y. Suen du centre de recherche (CENPARMI) de l'université de Concordia, Montréal, Québec, Canada. Il utilise trois ensembles de caractéristiques dont, les caractéristiques du gradient, les primitives de Gabor et les descripteurs de Fourier. Les auteurs utilisent trois SVMs différentes, chacune a été entraînée sur un type de caractéristiques différent. Les trois classifieurs sont divisés en deux groupes ; l'un est primaire et l'autre est secondaire.

C. RWTH-OCR : le système de reconnaissance d'écriture arabe manuscrite RWTH-OCR a été soumis par Patrick Doetsch, Philippe Dreuw, Mahdi Hamdani, Christian Plahl, et Hermann Ney de l'université de RWTH Aachen, Allemagne. Ce système est une adaptation du système RWTH ASR (utilisé pour la reconnaissance de la parole) pour la reconnaissance de l'écriture arabe manuscrite.

D. REGIM : ce système est soumis par Mahdi Hamdani et son groupe de Recherche (REGIM) de l'école nationale des ingénieurs de Sfax (ENIS). Ce système est basé sur Les HMMs, et est

une version améliorée du système proposé. L'amélioration consiste en une optimisation de l'architecture des HMMs (Nombre d'états) par l'application de l'algorithme PSO (Particle Swarm Optimization).

Les résultats montrent, que les systèmes de reconnaissance de l'écriture arabe manuscrite dans cette cinquième compétition n'ont fait aucun progrès. Seulement un seul des systèmes participants, a montré un taux de reconnaissance élevé. Le tableau III.9 présente la comparaison des résultats de ces différents systèmes. Le système 3 (RWTH-OCR) a remporté la compétition ICDAR 2011 par l'obtention du meilleur taux de reconnaissance.

G-ID	S-ID	Approche	set d	set e	set f _a	set f _r	set f _g	set f			set s		
			top1	top 1	top 1	top 1	top 1	top 1	top 5	top10	top 1	top 5	top 10
JU-OCR	1	RF	75.49	63.75	64.96	66.77	67.64	63.86	80.18	84.65	49.75	66.86	72.46
CENPARMI-OCR	2	SVM	99.90	99.91	40.00	40.00	40.00	40.00	69.33	74.00	35.52	54.56	63.84
RWTH-OCR	3	HMM	99.67	98.61	92.35	92.20	92.81	92.20	95.73	96.15	84.55	91.99	93.52
REGIM	4	HMM	94.12	86.62	80.60	81.36	81.52	79.03	89.35	91.34	68.44	81.99	84.98

Tableau III. 9 : Résultats de reconnaissance des systèmes participants à la compétition ICDAR 2011.

III. Conclusion

Ce chapitre nous a permis de présenter notre domaine d'intérêt : la reconnaissance de l'écriture arabe Manuscrite. Nous avons en premier lieu exposé les différentes propriétés morphologiques de l'écriture arabe manuscrite, ainsi que les difficultés rencontrées lors de la conception d'un système performant de sa reconnaissance. Nous avons présenté également un état de l'art des travaux de recherche concernant les étapes que subit l'image du mot manuscrit après son acquisition jusqu'à la phase finale de la prise de décision, en plus nous avons présentés quelques compétitions internationales sur le domaine de la reconnaissance de l'écriture arabe manuscrite.

CHAPITRE IV

Classification

Chapitre IV Classification

IV.1 Introduction

La classification est un algorithme qui consiste, à élaborer une règle de décision, qui transforme les attributs caractérisant les formes à reconnaître en appartenance à une classe. Dans le domaine de la reconnaissance de l'écriture manuscrite, l'entrée de l'algorithme est l'ensemble des attributs caractéristiques d'un mot, la sortie est la classe de ce mot. Si l'entrée est constituée de plusieurs données, la sortie de l'algorithme est la liste des classes correspondantes. Le processus, qui réalise cette opération est appelé classifieur. Dans ce chapitre, nous allons présenter deux types de classifieurs : les k plus proches voisins flous et les séparateurs à vastes marges (SVM) qui sont employées dans la réalisation de notre système de reconnaissance de mots arabes manuscrits.

IV.2 Apprentissage

L'apprentissage automatique est défini selon Mitchell [108] comme, « Un sous-domaine de l'intelligence artificielle qui s'intéresse à conférer aux machines la capacité de s'améliorer à l'accomplissement d'une tâche, en interagissant avec leur environnement». Il se divise principalement en deux grandes catégories : l'apprentissage supervisé et l'apprentissage non-supervisé.

L'apprentissage supervisé consiste principalement à définir des règles qui permettent de classer des objets ou des formes dans des classes, en utilisant des variables quantitatives ou qualitatives caractérisant ces formes. On dispose au départ d'un échantillon représentatif de l'ensemble des objets à reconnaître, appelé base d'apprentissage dont les classes de ces objets sont bien connues par le module d'apprentissage. Cette phase consiste à analyser les ressemblances entre les éléments d'une même classe et les dissemblances entre les éléments de classes différentes pour trouver la meilleure partition de l'espace des représentations. Les paramètres décrivant cette partition, permettent la conception d'un modèle qui sera utilisé ensuite par le module de décision pour classer les formes qui lui sont présentées.

Dans le cas de l'apprentissage non supervisé, on dispose d'un grand nombre de formes non étiquetées, et le module de classification va se charger de désigner automatiquement les formes appartenant à une même classe sans aucune intervention de supervision.

IV.3 Méthodes de classification

Les méthodes de classification peuvent être classées en quatre catégories : les techniques d'appariement de modèles (Template matching) basées sur l'estimation d'un degré de similitude entre deux vecteurs dans l'espace des attributs. Ces techniques peuvent être divisées en trois approches : l'appariement direct, appariement élastique et déformable et appariement par relaxation. Les techniques statistiques et structurelles consistent à utiliser des fonctions de décision statistiques déterminant la probabilité d'un modèle observé (modèle statistique, structurel ou autres) pour être un élément d'une classe prédéfinie. Plusieurs approches bien connues font partie de cette catégorie telles que : le classifieur Bayésien, les modèles de Markov cachés (HMMs), les séparateurs à vaste marge (SVMs), K-plus-Proche-Voisin (K-NN), et les arbres de décisions. Le quatrième type des méthodes de classification sont les réseaux de Neurones Artificiel (ANNs), qui sont des modèles informatiques inspirés du fonctionnement des neurones du cerveau humain. Dans de le domaine de la reconnaissance de l'écriture arabe manuscrite, la majorité de ces classifieurs sont utilisés. Nous intéressons dans notre travail à deux types de ces méthodes: le classifieur k Plus proches voisins Flou et les séparateurs à vaste marge (SVM).

IV.4 la méthode des K plus proches voisins flous

L'algorithme des k-plus-proches-voisins (k-nn : pour k-neighrast neighbors en anglais), est l'un des algorithmes les plus simples de la classification automatique supervisée. Cette méthode est basée sur l'hypothèse que les points proches dans l'espace de primitives, ont une forte probabilité d'appartenir à la même classe. Son principe de décision consiste tout simplement à calculer la distance d'une forme inconnue x , à tous les échantillons fournis. Puis on sélectionne les k plus proches échantillons et on affecte x , à la classe majoritaire parmi ces k échantillons. Le choix de l'entier k et de la métrique de distance jouent un rôle primordial dans la méthode des k-plus proches voisins. L'une des difficultés qui se posent lors de l'utilisation de cette technique est de donner à chacun des échantillons étiquetés une importance égale pour décider des appartenances de classe de la forme à reconnaître, quel que soit leur typicité. Pour remédier à ce problème, des extensions diverses et des améliorations de la règle de K-NN ont été effectuées par beaucoup de chercheurs. Parmi ces améliorations, on trouve le k plus proches voisins flous proposé par Keller [109] qui a introduit la théorie des ensembles flous sur la technique de K-NN pour développer une version floue de l'algorithme.

Le principe de la méthode k plus proche voisin flou, est similaire à la règle des k plus proches voisins classiques. Il se base sur une étape de recherche des k points voisins les plus proches du prototype à reconnaître. On examine parmi les K voisins retournés, le taux de mélange des classes. Le K-PPV flou permet en plus, d'attribuer suivant les distances aux classes des k prototypes voisins, des degrés d'appartenance à ces classes. Le point à classer x_i , se voit attribuer un coefficient d'appartenance μ_{ji} à chaque classe j. Celui-ci est fonction des distances et des coefficients d'appartenance de ses k plus proches voisins. Ces coefficients doivent vérifier l'appartenance à l'intervalle $[0,1]$ de μ_{ji} , pour tous les i et tous les j.

Les coefficients d'appartenance d'un nouveau point x_i à la classe j est donnée par :

$$u_i(x) = \frac{\sum_{j=1}^k u_{ij} \left[\frac{1}{\|x - x_j\|} \right]^{\frac{2}{m-1}}}{\sum_{j=1}^k \left[\frac{1}{\|x - x_j\|} \right]^{\frac{2}{m-1}}} \quad (\text{IV.1})$$

Tel que m est un entier, et u_{ij} est le coefficient d'appartenance à la classe C_j , =1 si appartient aux K plus proches voisins, de la i^{eme} observation, parmi les k plus proches voisins de x_i sinon=0. La variable m quant à elle, détermine l'importance de la contribution de la distance dans le calcul de la fonction d'appartenance (contrôle l'efficacité de l'ampleur de la distance). m est le paramètre de fuzzification : si m croît, la contribution des voisins est d'avantage pondéré et la notion de distance perd de son importance, si m tend vers l'unité, la contribution des voisins les plus proches sera favorisée, ainsi la notion de distance prend de l'importance ; si m vaut 2, on pondère la contribution de chaque voisin par l'inverse de la distance respective, au carré, qui sépare une observation de l'observation à classer. L'algorithme des k plus proches voisins flous peut se résumer comme suit:

Début

Entrées: x , d'une classe inconnue

Soit k , $1 \leq k \leq n$.

Initialiser $i=1$.

Faire jusqu'a ce que (les k plus proches voisins de x seront trouvés)

Calculer la distance entre x et x_i

Si ($i \leq k$) alors

Inclure x_i dans l'ensemble des k plus proches voisins

Sinon si (x_i est plus proche de x que tout le plus proches voisins précédents) alors

Supprimer la plus éloignée des k plus proches voisins

Inclure x_i dans l'ensemble des k plus proches voisins

Fin de Si

Fin de Faire

Initialiser $i=1$.

Faire jusqu'à ce que (la fonction d'appartenance de x est affectée dans toutes les classes)

Calculer $u_i(x)$ en utilisant la formule (IV.1)

Incrémenter i .

Fin de Faire

Fin

IV.5 Machines à vecteurs supports (SVM)

Les machines à vecteurs de support (SVM), ou appelées encore les séparateurs à vaste marge est un algorithme d'apprentissage qui a été appliqué avec succès dans une grande variété de problèmes de régression et de classification. La robustesse, la précision et la capacité de généralisation font du SVM un choix attirant dans de nombreuses tâches d'apprentissage automatique. Les SVMs sont de nouvelles techniques de la théorie statistique de l'apprentissage [110], proposées par Vapnick en 1995 et développées à partir de la théorie de la Minimisation du risque structurel (Structural Risk Minimization SRM). Ce type d'algorithmes d'apprentissage automatique consiste à mettre en œuvre la stratégie de maintenir le risque empirique fixe et en minimisant l'intervalle de confiance. Les SVMs sont dans leur origine conçus pour la classification binaire et la régression. Aujourd'hui, elles sont utilisées dans différents domaines de recherche et d'ingénierie telle que le diagnostic médical, le marketing, la biologie, et la reconnaissance des formes.

IV.5.1 Principe

Cet algorithme est une méthode de classification binaire, qui consiste à trouver une frontière de décision ou hyperplan qui sépare les données positives des données négatives dans l'ensemble des données en commettant un minimum d'erreurs. Le principe de base de cette méthode s'articule autour de deux notions principales : la marge maximale et la fonction noyau, qui existent depuis plusieurs années avant qu'elles ne soient mises en commun pour construire le classifieur SVM. La marge peut être définie comme étant la distance entre l'hyperplan séparateur et les échantillons les plus proches appelés aussi les vecteurs supports. Cette distance doit être maximale pour garantir une bonne généralisation pour mieux classer les nouveaux exemples. La deuxième notion clé d'un classifieur SVM, est la notion de la fonction noyau qui consiste à transformer un problème de séparation non-linéaire dans l'espace de représentation des données en un problème de séparation linéaire dans un espace de représentation de plus grande dimension appelé aussi espace de redescription.

IV.5.2 Notions de base : Hyperplan, marge et support vecteur

Soit un nuage de points (voir Figure IV.1) de natures différentes (points rouges, points bleus). Le premier objectif de la méthode SVM, est de trouver une frontière de décision (hyperplan séparateur) qui puisse séparer le nuage de points en deux régions en commettant un minimum d'erreurs, c'est-à-dire trouver l'hyperplan optimal. En réalité, il existe plusieurs hyperplans séparateurs où leur performances en apprentissage sont identiques c'est-à-dire que le risque empirique est le même ; mais dont leurs pouvoirs de généralisation sont très différents.

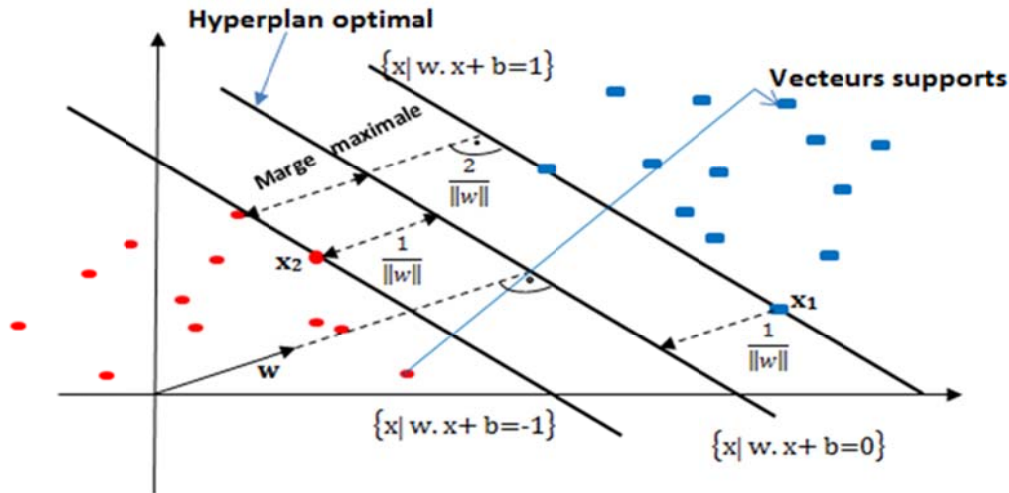


Figure IV.1 : Représentation des notions de Base du SVM.

Pour résoudre ce problème, Vapnik a défini l'hyperplan optimal comme étant l'hyperplan unique qui maximise la marge ou la distance entre les exemples d'apprentissage les plus proches et l'hyperplan séparateur. Ces exemples qui consistent à déterminer l'hyperplan séparateur optimal sont appelés vecteurs supports.

IV.5.3 Les SVMs linéaires

À partir d'un ensemble d'apprentissage (x_i, y_i) où $x_i \in \mathbb{R}^M$, ensemble des données et $y_i \in \{\pm 1\}$, celui des classes, les SVM consistent à trouver l'Hyperplan séparateur Optimal (HSO) qui maximise la distance entre l'hyperplan et les deux classes c'est-à-dire l'hyper plan qui maximise la marge.

L'hyperplan est défini par $\vec{w} \cdot \vec{x} + b = 0$ où (\vec{w}, b) désignent les paramètres de l'hyperplan (respectivement un vecteur normal au plan et le biais). Le classifieur est donné par :

$$f: \vec{x} \in \mathbb{R}^M \rightarrow \text{Sign}(\vec{w} \cdot \vec{x} + b) \in \{\pm 1\} \tag{IV.2}$$

Néanmoins, ce dernier doit satisfaire la condition suivante :

$$\begin{cases} \vec{w} \cdot \vec{x} + b \geq 0 & \text{si } y_i = +1 \\ \vec{w} \cdot \vec{x} + b \leq 0 & \text{si } y_i = -1 \end{cases}$$

De plus, l'hyperplan séparateur optimal (HSO) doit maximiser la marge : $2/\|\vec{w}\|$. La solution est obtenue en résolvant le problème de maximisation de la marge sous les contraintes précédentes, ce qui mène au problème d'optimisation quadratique suivant :

$$\min_{(w,b)} \frac{\|\vec{w}\|^2}{2} \quad (IV.3)$$

$$\text{Tel que: } y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, i \in \langle 1, N \rangle$$

En introduisant les multiplicateurs de Lagrange on obtient le problème dual suivant :

$$\max_{\lambda} W(\lambda) = \frac{-1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j \vec{x}_i \cdot \vec{x}_j + \sum_{i=1}^N \lambda_i \quad (IV.4)$$

Tel que :

$$\sum_{i=1}^N \lambda_i y_i = 0, 0 \leq \lambda_i, \forall i \in \langle 1, N \rangle$$

Qui est un problème d'optimisation quadratique convexe sous contraintes linéaires.

L'Hyperplan séparateur Optimal (HSO) est obtenu par : $\vec{w} = \sum_{i=1}^N \lambda_i y_i \vec{x}_i$ et $b = y_i - \vec{w} \cdot \vec{x}_i$, avec i tel que $\lambda_i \neq 0$. La fonction de classification vaut donc :

$$f(\vec{x}) = \vec{w} \cdot \vec{x} + b = \sum_{i=1}^N \lambda_i y_i \vec{x}_i \cdot \vec{x} + b \quad (IV.5)$$

Pour des données non linéairement séparables ou même pour dans le cas des données linéairement séparables, il peut être recommandé de dépasser quelques contraintes pour agrandir la marge. Ainsi, on améliore la robustesse du processus d'apprentissage en réduisant l'erreur. Pour atteindre cet objectif, on introduit les variables de relaxation ξ_i et le paramètre de régularisation C appelé aussi paramètre de pénalisation du relâchement. Cette dernière est une constante positive à fixer a priori judicieusement. En effet, elle constitue un compromis entre la mauvaise classification et la complexité de la fonction, ou en d'autres termes entre l'erreur empirique et l'erreur de généralisation. Donc, cette constante permet de contrôler l'importance de l'erreur que l'on s'autorise par rapport à la taille de la marge. Le problème d'optimisation primale devient :

$$\min_{(w,b)} \frac{\|\vec{w}\|^2}{2} + C \sum_{i=1}^N \xi_i \quad (IV.6)$$

$$\text{Tel que : } y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i \in \langle 1, N \rangle$$

Ce problème d'optimisation est résolu en utilisant les multiplicateurs de Lagrange et devient :

$$\max_{\lambda} W(\lambda) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j \vec{x}_i \vec{x}_j + \sum_{i=1}^N \lambda_i \quad (IV.7)$$

Tel que :

$$\sum_{i=1}^N \lambda_i y_i = 0, 0 \leq \lambda_i \leq C, \forall i \in \{1, N\}$$

IV.5.4 Les SVMs non linéaires

Une grande majorité des problèmes réels sont des problèmes non linéairement séparables où il n'y a pas de séparation linéaire possible entre les données. De ce fait, le classifieur de marge maximale ne peut pas être utilisé, car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables et le séparateur optimal doit pouvoir prendre une forme plus compliquée. La méthode du noyau, constitue un moyen élégant et efficace pour traiter ce problème. L'idée de base de cette méthode, est de projeter les points d'apprentissage x_i dans un espace T de dimension q , plus élevée que n , grâce à une fonction non-linéaire (ϕ) appelée fonction noyau choisie a priori et d'appliquer la même méthode d'optimisation de la marge dans l'espace T . Il s'agit donc de transformer un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de grande dimension (Figure IV.2). L'espace ainsi obtenu est appelé espace des caractéristiques ou aussi espace transformé. Dans ce nouvel espace de caractéristiques, la fonction objective à optimiser devient:

$$\max_{\lambda} W(\lambda) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j \langle (x_i), (x_j) \rangle + \sum_{i=1}^N \lambda_i \quad (IV.8)$$

Où : $\langle (x_i), (x_j) \rangle$ représente le produit scalaire des deux images des vecteurs x_i et x_j dans le nouvel espace et dont le résultat est un scalaire.

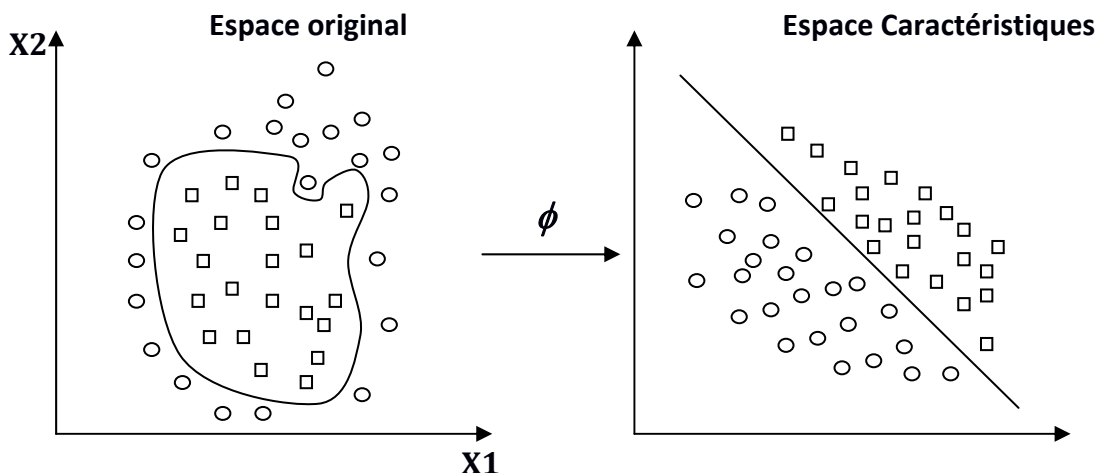


Figure IV.2 : Changement d'espace.

Dans le calcul de l'optimum de la fonction, on utilise une astuce appelée "Noyau" (Kernel), au lieu de calculer $\phi(x_i)$, $\phi(x_j)$ et leur produit scalaire, on calcule plutôt une fonction $K(x_i, x_j)$ qui représente à la fois les deux transformations (qui peuvent être inconnues) et leur produit scalaire. Cette fonction, permet de surmonter le problème de détermination de la transformation (ϕ) et permet d'apprendre des relations non linéaires par des machines linéaires. En pratique, il existe certains noyaux qui sont très utilisés et qui sont considérés comme standards. Une fois le noyau choisi, la fonction objective peut être calculée par la formule suivante :

$$\max_{\lambda} W(\lambda) = \frac{-1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j k(x_i, x_j) + \sum_{i=1}^N \lambda_i \quad (\text{IV.9})$$

$$\text{La fonction de décision devient: } H(x) = \sum_{i \in S} \lambda_i y_i k(x_i, x_j) + b \quad (\text{IV.10})$$

Où S représente l'ensemble des vecteurs supports.

Il n'est cependant pas nécessaire de connaître explicitement la fonction (ϕ) : toute fonction qui satisfait les conditions de Mercer [111] peut être utilisée comme noyau.

IV.5.5 Les fonctions noyaux

Du fait que les données apparaissent dans tous les calculs uniquement sous forme de produits scalaires $\langle \phi(x_i) \cdot \phi(x_j) \rangle$, pour éviter le calcul coûteux de ces produits scalaires, il suffit de trouver une façon efficace de calculer ce produit. Cela est réalisé, en faisant appel à une fonction noyau $k(x_i, x_j)$ (fonction symétrique continue satisfaisant la condition de Mercer définie par $K(x_i, x_j) = \langle \phi(x_i) \cdot \phi(x_j) \rangle$).

Ainsi il n'est pas nécessaire de faire la projection (ϕ), l'espace d'arrivée appelé aussi espace de redescription reste virtuel. La fonction noyau (plus facile à calculer) donne le même résultat dans l'espace d'entrée. Ceci est appelé 'the kernel trick'. Cette fonction peut être vue comme une mesure de similarité non linéaire. Toutes les considérations décrites précédemment seront vérifiées puisque nous maintenons toujours la séparation linéaire, mais dans un espace augmenté. Le problème qui reste à l'utilisateur est celui de choisir une fonction noyau adéquate. Il existe quelques familles paramétrables connues (voir Tableau IV.1). Il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application (il s'agit d'exploiter le maximum de connaissances préalables dont on dispose sur le problème étudié et sur les données). Plus que la fonction prendra en compte les informations disponibles sur le problème considéré, meilleurs devraient être les résultats. Dans le tableau suivant on cite quelques noyaux usuels :

Fonction noyau	Forme fonctionnelle
Linéaire	$k(x_i, x_j) = x_i^T x_j$
Polynomiale	$k(x_i, x_j) = (x_i \cdot x_j + c)^n$
A base radiale	$k(x_i, x_j) = \exp\left(-\frac{\ x_i - x_j\ ^2}{\sigma^2}\right)$
Sigmoïde	$k(x_i, x_j) = \tanh(a(x_i \cdot x_j) - b)$
Laplace	$k(x_i, x_j) = \exp(- \gamma \ x - y\)$

Tableau IV.1 : Les fonctions noyau les plus courantes.

IV.5.6 Les hyper-paramètres

En plus des paramètres qui sont ajustés, avant la réalisation de la phase d'apprentissage en utilisant un modèle SVM et qui seront non modifiables, il existe aussi, d'autres paramètres appelés hyper-paramètres qui peuvent être optimisés durant la phase d'apprentissage. Un exemple de ce type de paramètres est la constante de régularisation, qui sert à déterminer le compromis désiré entre la maximisation de la marge entre les classes et la tolérance du modèle aux erreurs d'apprentissage. Si C prend une valeur très grande, nous avons une haute pénalité pour des points non séparables et nous pouvons stocker beaucoup de vecteurs de support et un sur-apprentissage (overfitting). Et si C est très petite, nous pouvons avoir un sous-apprentissage (underfitting). Il existe, aussi des hyper-paramètres propres à chaque type de noyaux. Par exemple la forme générale d'un noyau polynomial, est donnée par l'expression suivante :

$$k(x_i, x_j) = (Ax_i \cdot x_j + B)^p \quad (\text{IV. 11})$$

Ce type de noyau est caractérisé par les deux paramètres A et B qui prennent toujours la valeur "1". Cependant, P est un hyper-paramètre qui doit être ajusté pendant la phase d'apprentissage. Un autre noyau très utilisé est le noyau Gaussien qui possède la forme suivante :

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right) \quad (\text{IV. 12})$$

L'hyper-paramètre (σ), consiste à régler la largeur de la gaussienne. Si la valeur de (σ) est très grande, la similarité d'une observation par rapport à les observations voisines sera assez élevée, alors si σ prend une valeur proche de 0, l'observation ne sera similaire à aucune autre.

IV.5.7 SVM Multi-classes

Les SVMs ont été initialement conçus pour effectuer la classification binaire; Cependant, les applications de la classification binaire sont très limitées, et la plupart des applications réelles impliquent la classification multi-classes, particulièrement dans le domaine de la reconnaissance des formes. Bien entendu, plusieurs recherches ont été développées pour proposer des algorithmes permettant d'étendre les SVMs à résoudre les problèmes multi-classes. Mais l'inconvénient majeur de ces propositions, est qu'elles sont très lentes par rapport à la version binaire. Donc, le problème de classification multi-classes par les SVMs, reste toujours un domaine de recherche Actif. Dans la Figure VI.3, on présente un exemple de classification SVM à trois classes. Dans la section suivante, nous présentons les différentes approches proposées pour l'extension SVM aux problèmes de classification multi-classes.

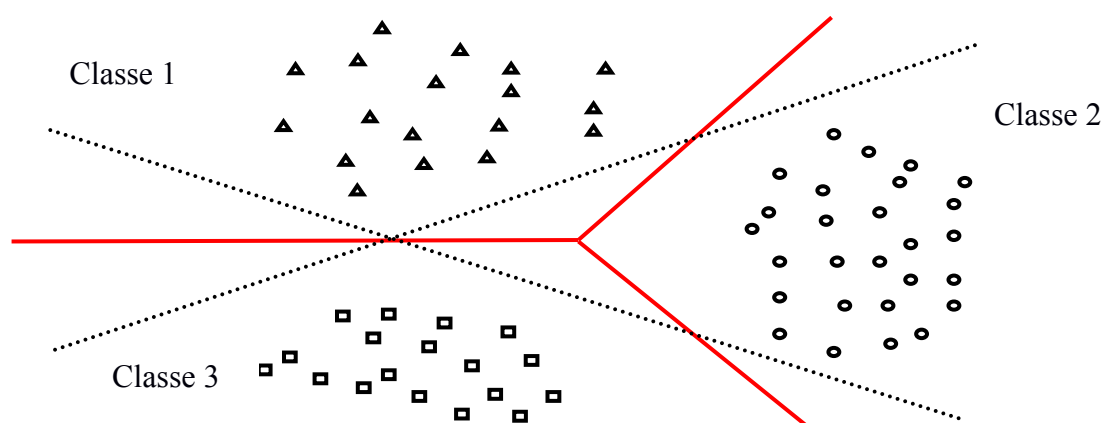


Figure IV.3 : Exemple de classification SVM 3-classes extrait de [112].

IV.5.7.1 L'approche Un Contre Tous

L'approche un contre tous, est considérée comme étant la plus ancienne et la plus simple des méthodes de décomposition. Elle consiste, à utiliser autant de classifieurs qu'il y a de classes. Pour entraîner chaque classifieur, à distinguer les échantillons de sa classe de ceux de toutes les autres classes. Toutes les observations de la classe en question sont considérées positives (+1), alors que toutes les observations n'appartenant pas à la classe sont considérées négatives (-1). A la fin de cette phase, on dispose de k hyperplans (k est le nombre de classes). L'inconvénient majeur de cette méthode, est qu'on peut avoir plusieurs sorties positives pour la même observation de test par des classifieurs différents. La solution pour ce problème est d'utiliser un vote majoritaire pour attribuer cette observation à une classe. Alors, la décision d'affecter une observation x à une classe k , est réalisé en utilisant la règle suivante : $\text{Arg max } (w_i \cdot x + b_i)$. La critique qu'on peut donner à l'approche un contre tous réside dans sa complexité d'apprentissage,

vu que le nombre des exemples d'apprentissage devient très important. En effet, chaque classifieur est entraîné en utilisant tous les exemples d'apprentissage disponibles.

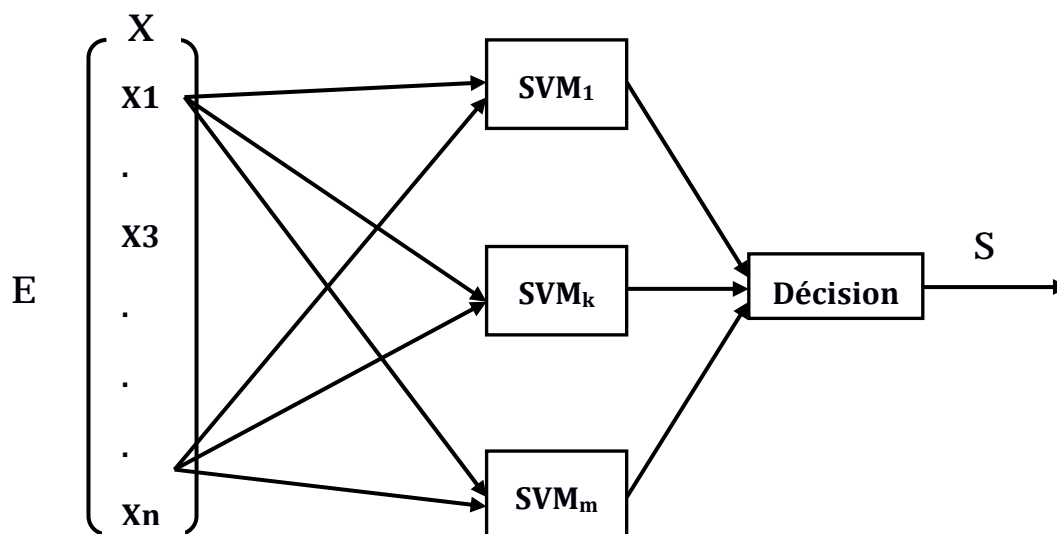


Figure IV.4 : Architecture d'un système de classification basée sur la méthode Un-Contre-Tous

IV.5.7.2 L'approche Un Contre Un

Une autre méthode de décomposition très simple, est la méthode 'un contre un'. Ordinairement attribuée à Kner et ses co-auteurs [113] qui l'ont proposée pour les réseaux de neurones, elle consiste à utiliser un classifieur pour chaque paire de classes. Pour un problème à M classes, on a $M(M-1)/2$ classifieurs. Ce classifieur, considère les données de la première classe comme des exemples positifs et les données de la deuxième classe comme des exemples négatifs. La forme à reconnaître est soumise à tous ces classifieurs 1-vs-1, la classe remportant le plus de votes remporte la décision. Cette dernière s'obtient habituellement en effectuant un vote majoritaire (max-wins voting). La voix de chaque classifieur peut être pondérée par une fonction de la valeur de la sortie calculée.

IV.5.7.3 DAG-SVM

Une amélioration de la méthode un contre un, a été proposée par Platt [114]. Cette technique utilise le même principe que celui de la méthode un-contre-un dans la phase d'apprentissage. Donc, elle utilise $m(m-1)/2$ modèles de classification binaire. Mais, la phase de test est basée sur l'utilisation d'un graphe de décision orienté acyclique comme son nom l'indique (Directed Acyclic Graph). Dans cette phase, chaque modèle vote contre la classe non choisie. Le résultat de cette action, consiste à rejeter l'une des classes à la fin de chaque classification binaire.

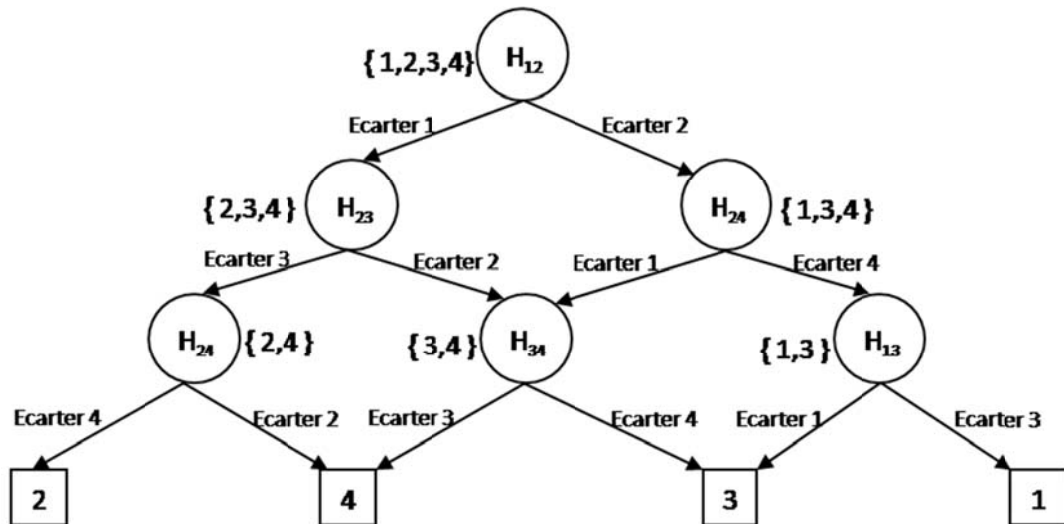


Figure IV.5 : Une classification de trois classes par DAG-SVM.

Donc, on obtient la classe finale après $m-1$ étapes. La Figure IV.5 présente un exemple de classification de trois classes par la méthode DAG-SVM. Cette approche donne un taux de classification similaire à celui de la méthode un-contre-un, mais elle présente l'avantage d'être la méthode la plus rapide.

IV.6 Conclusion

Nous avons présenté au cours du présent chapitre, une étude théorique générale sur l'étape de classification, qui joue un rôle primordial dans tout système de reconnaissance de l'écriture manuscrite. Nous avons présenté ainsi deux types de classifieurs, qui sont employés dans notre système de reconnaissance de l'écriture arabe manuscrite : le classifieur k plus proches voisins flous et les séparateurs à vaste marge (SVMs). Nous avons aussi décrit plusieurs approches de classification utilisées en reconnaissance de l'écriture. Par ailleurs, ces classifieurs peuvent être combinés pour améliorer les performances des classifieurs individuels. L'idée principale est d'exploiter au maximum la complémentarité des modélisations, afin d'obtenir des décisions plus robustes, et compenser les faiblesses de chaque classifieur. Dans le chapitre suivant nous allons introduire la théorie de la combinaison de classifieurs, et nous mettrons l'accent sur les différents algorithmes et techniques de fusion de classifieurs en soulignant leurs avantages et leurs inconvénients.

CHAPITRE V

Combinaison de classifieurs

Chapitre V Combinaison de classifieurs

V.1 Introduction

Le domaine de la reconnaissance de formes se caractérise aujourd'hui par la présence d'un grand nombre de méthodes de classification. Cependant, et malgré les avancés dans ce domaine, on se rend compte qu'il n'existe aucun classifieur qui peut manifester une supériorité incontestable sur les autres classifieurs dans tous les problèmes et toutes les situations. Dans nombreuses recherches récentes ont montré, et sans aucun doute que plutôt de chercher à optimiser un seul classifieur en choisissant les meilleurs configurations et les meilleurs caractéristiques pour un problème donné de reconnaissance de formes, il est plus judicieux de combiner des méthodes de reconnaissance. La combinaison de classifieurs est un ensemble de méthodes basées sur la coopération de plusieurs classifieurs, en combinant leurs décisions pour résoudre un problème de classification donné. Ces méthodes doivent produire des résultats plus fiables que ceux d'un classificateur individuel. L'utilisation simultanée de plusieurs classifieurs en même temps permet généralement d'en cumuler les avantages sans en cumuler les inconvénients. En effet, l'exploitation des différentes informations, que les classifieurs fournissent en utilisant des techniques de combinaison de classifieurs, peut aboutir à une classification plus précise. Même si le classifieur est moins performant, la connaissance de son comportement permet d'apporter des informations complémentaires qui peuvent être utilisées pour désigner la vraie classe de la forme à reconnaître.

Dans ce chapitre, nous présentons un état de l'art des systèmes de combinaison de classifieurs. Nous présentons également les stratégies de prises de décision possibles lorsque plusieurs classifieurs interviennent dans un processus de reconnaissance et les spécificités relatives à chaque stratégie. Nous nous focalisons en particulier sur la combinaison parallèle de classifieurs et nous présentons un panorama des principales méthodes mises en œuvre dans le domaine de la reconnaissance de l'écriture manuscrite.

V.2 Pourquoi combiner plusieurs classifieurs?

La combinaison des classifieurs, occupe aujourd'hui une place très importante dans la communauté scientifique pour la simple raison qu'elle se montre au fil des études particulièrement très performante. Notamment, il a été démontré à travers plusieurs travaux de recherche que la combinaison de classifieurs, est une solution efficace à de nombreux problèmes auxquels se confrontent les algorithmes qui induisent des classifieurs uniques. La première raison qui semble la plus immédiate pour la combinaison de classifieurs, concerne l'amélioration

des performances de la classification. L'idée de base, est de combiner les sorties de plusieurs classifieurs simplement dans l'espoir d'obtenir une décision plus fiable. Intuitivement, la confrontation de l'avis de plusieurs experts à propos d'un même problème, permet de compenser les erreurs de quelques-uns d'entre eux par une majorité de bonnes prédictions données par les autres.

Une seconde raison pour vouloir combiner des classifieurs, est la complexité du problème traité. Si le problème est trop complexe pour être traité par un seul classifieur, il peut être très envisageable de le décomposer en plusieurs sous-problèmes qui seront plus simples et peuvent être donc ensuite résolus séparément par des classifieurs différents. La prise de décision finale est ensuite effectuée par combinaison de ces classifieurs.

Une dernière raison a été bien expliquée par le fameux théorème d'optimisation 'no free lunch' [115], Ce théorème démontre, qu'il n'existe aucun classifieur unique, capable de résoudre tout type de problèmes. En résumé, on peut vouloir combiner plusieurs classifieurs pour les raisons suivantes [116] :

- Ils permettent l'amélioration de la prise de décisions en s'appuyant sur l'avis de plusieurs experts dans un domaine donné au lieu d'un seul.
- Ils consistent à élargir l'ensemble des solutions possibles par la conception des modèles plus complexes que l'on ne pourrait pas obtenir avec des classifieurs individuels.
- Ils permettent d'éviter les optima locaux.
- Ils offrent la capacité de traiter des problèmes trop complexes en les décomposant en plusieurs sous problèmes.
- Ils s'avèrent plus génériques que les classifieurs individuels, et permettent une compréhension plus efficace d'un grand nombre de problèmes.

V.3 Types de sorties d'un classifieur

On peut distinguer trois types de sorties d'un classifieur. Soit un classificateur (e) qui reçoit une entrée (x) et fournit une sortie (C_i) tel que ($i=1..m$), (m) est le nombre de classes.

V.3.1 Type classe

La sortie de type classe, est la sortie la plus simple qu'un classifieur puisse fournir, elle consiste à ne donner que la proposition de la classe qu'un classifieur attribue à la forme à reconnaître, sans aucune autre information supplémentaire. Il s'agit alors de l'unique catégorie que le classifieur estime être la plus probable pour l'entrée proposée. Le classificateur donne une

étiquette (C_i) indiquant la classe de l'entrée traitée. $e_j(x) = C_i$ ($i \in \{1..m\}$). Les classifieurs qui fournissent un tel type de sortie, seront appelés classifieurs de type classe.

V.3.2 Type rang

$$e_j(x) = (r_{1,j}, r_{2,j}, \dots, r_{N,j}) \quad (V.1)$$

$r_{i,j}$ est le rang attribué à la classe C_i par le classifieur e_j . Dans ce type de sortie, les classifieurs fournissent en sortie une liste ordonnée de classes. Leurs préférences sont exprimées par un rang associé à chacune des classes. La classe placée au premier rang de la liste proposée par le classifieur, est considérée comme la plus probable pour la forme à reconnaître x , par contre la classe du dernier rang est la moins probable.

V.3.3 Type mesure

$$e_j(x) = (m_{1,j}, m_{2,j}, \dots, m_{N,j}) \quad (V.2)$$

$m_{i,j}$ est la mesure attribuée à la classe i par le classifieur e_j . Ce type de sortie indique le niveau de confiance du classifieur dans sa proposition. Le classifieur fournit donc un vecteur de mesure de taille N . Cette mesure, normalisée ou non, peut prendre la forme d'une distance, une valeur de confiance, une probabilité a posteriori, un score, une fonction de croyance, une possibilité, une mesure floue, etc.

V.4 Les méthodes de combinaison des classifieurs

L'intérêt croissant des chercheurs pour la combinaison des classifieurs, a entraîné la mise au point de nombreux schémas traitant les données de manières différentes. Généralement, trois approches pour la combinaison de classifieurs peuvent être envisagées: parallèle, séquentielle et hybride. Cependant, malgré l'existence d'un grand nombre de des schémas de combinaison, la détermination de la meilleure architecture reste un problème ouvert.

V.4.1 Approche séquentielle

La combinaison séquentielle, appelée également combinaison série, est une architecture organisée en niveaux successifs de décision qui servent à réduire d'une manière progressive le nombre de classes possibles (La liste de classes candidats est ainsi progressivement réduite jusqu'à ce qu'il ne reste qu'une seule décision possible), où chaque niveau de décision est caractérisé par la présence d'un seul classifieur qui doit prendre en compte la réponse fournie par le classifieur placé en amont, pour traiter un rejet ou confirmer une décision obtenue sur la forme présentée à son entrée (Figure V.1). Cette approche peut être considérée comme un filtrage

progressif des décisions, dans la mesure où elle permet de diminuer au fur et à mesure l'ambiguïté sur la classe proposée. Cela permet généralement de diminuer le taux d'erreur globale de la chaîne de reconnaissance [117]. Néanmoins, l'inconvénient majeur de cette approche de combinaison, réside dans sa sensibilité à l'ordre d'exécution des classifieurs, qui est primordial et influe sur le résultat final [118].

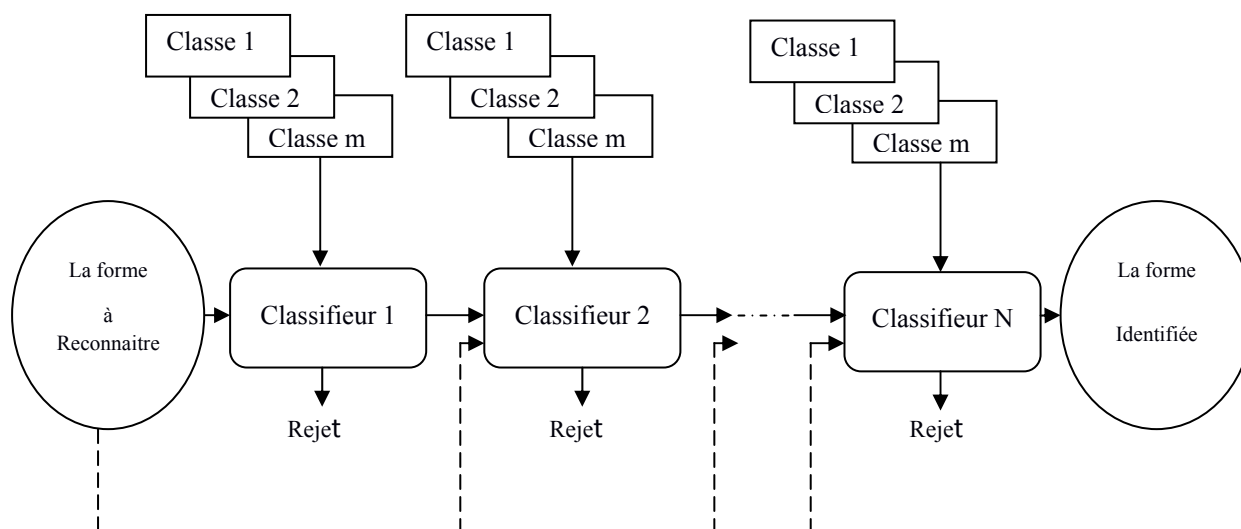


Figure V.1 : Combinaison séquentielle de classifieurs.

En effet, les classifieurs placés en amont doivent être robustes, c'est-à-dire que la vraie classe de la forme à reconnaître doit apparaître dans les listes successives proposées. Car un échec de décision du premier classifieur provoque une erreur, qui va se propager de façon irréversible dans toute la série des classifieurs utilisés. Donc la combinaison séquentielle, exige une certaine connaissance a priori du comportement de chacun des classifieurs. Dans cette approche, chaque classifieur est ajusté en fonction du classifieur placé en amont, ce qui peut conduire à un ré-paramétrage des classifieurs suivants en cas d'une simple modification du premier classifieur. Les travaux rentrant dans le cadre de l'écriture arabe manuscrite adoptant cette approche ne sont pas nombreux, parmi lesquels on a trouvé dans la littérature, le travail d'Al Ohali [119] en 2002 qui a proposé un système de reconnaissance des sous-mots en utilisant une combinaison séquentielle composée d'un classifieur global neuronal et un classifieur analytique markovien.

V.4.2 Approche parallèle

A la différence de l'approche séquentielle, dans l'approche parallèle, les classifieurs de base opèrent indépendamment les uns des autres et prennent leurs décisions sans tenir compte du reste du comité. Les décisions individuelles sont ensuite fusionnées à l'aide d'un opérateur de combinaison. Cette fusion peut être effectuée soit de manière démocratique, dans le sens où elle

ne favorise aucun classifieur par rapport à un autre (vote majoritaire par exemple), soit au contraire dirigé et, dans ce cas, on attribue à la réponse de chaque classifieur un poids de pondération en fonction de ses performances. Dans cette approche de combinaison l'ordre d'exécution des classifieurs de base n'intervient pas. La Figure V.2 fournit le schéma du principe de la combinaison parallèle des classifieurs.

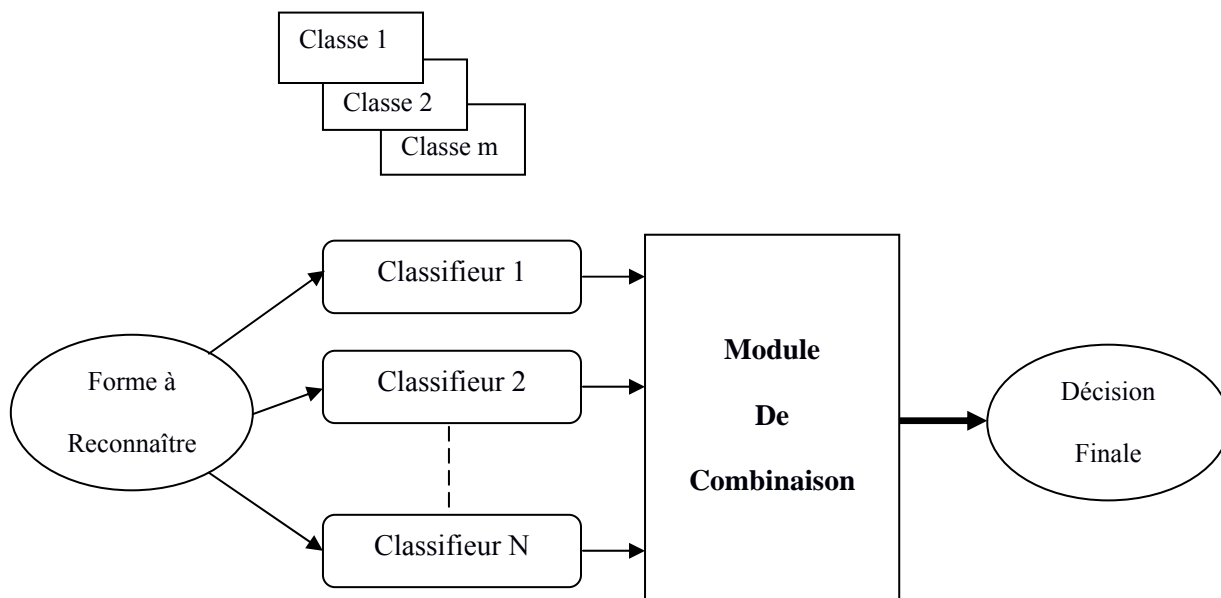


Figure V.2 : Principe de la combinaison parallèle de classifieurs.

L'un des inconvénients majeurs de l'approche parallèle, est son besoin de l'activation en même temps de tous les classifieurs de base, qui doivent participer de manière concurrente et indépendante à la prise de décision finale. Dès lors se posent les problèmes concernant la qualité des connaissances fournies par les classifieurs élémentaires et le degré de confiance qu'on peut accorder à chacun d'eux.

V.4.3 Approche hybride

L'approche hybride, est une combinaison à la fois des architectures séquentielles et parallèles afin de tirer pleinement avantage de chacun des classifieurs utilisés et de chacune des deux méthodes. La Figure V.3, présente un exemple de combinaison hybride, qui combine un classifieur en série avec deux classifieurs en parallèle. Cette architecture, illustre deux aspects de la combinaison qui sont d'une part la réduction de l'ensemble des classes possibles et d'autres parts la recherche d'un consensus entre les classifieurs afin d'aboutir à une décision unique. Ce type de configuration, nécessite une bonne connaissance des comportements de chacun des classifieurs utilisés afin de choisir la topologie la plus adéquate. On peut trouver comme topologies [28]:

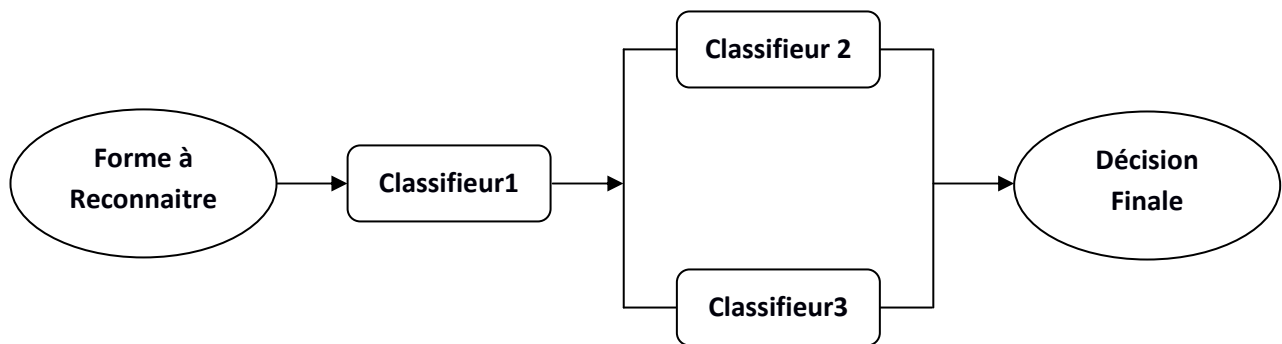


Figure V.3 : Combinaison hybride de classifieurs.

- Les topologies avec bouclage : ce type de topologie consiste à renvoyer le résultat de la combinaison à l'entrée du système de classification afin de modifier la sortie. Ce qui rend le système global souvent plus performant qu'un système classique ; cependant, l'analyse et le paramétrage de ce genre de méthodes s'avèrent plus difficile.
- Les topologies avec interaction entre les différents classifieurs : les classifieurs de base dans ce genre de topologie interagissent les uns sur les autres afin, éventuellement, de modifier leurs résultats avant la phase finale de fusion.
- Les topologies avec utilisation du prototype : dans ces topologies de combinaison, les données d'entrées initiales sont utilisées en même temps que les réponses des classifieurs.

Peu de travaux ont exploité la combinaison hybride dans le cadre de la reconnaissance de l'écriture arabe manuscrite ; parmi lesquels on peut citer le travail de Touj[120]. Les chercheurs ont proposés deux méthodes hybrides, la somme ascendante et la somme descendante.

V.5 Catégorisation des méthodes de combinaison parallèle

De nombreux travaux ont démontré que la combinaison des classifieurs dans ses différentes architectures (séquentielles, parallèles ou hybrides), améliore nettement les performances du système de reconnaissance par rapport à chacun des classifieurs individuels [121]. Cependant, parmi les différentes architectures permettant la combinaison d'un ensemble de classifieurs, l'architecture parallèle est celle qui est considérée comme la plus favorite de la plupart des chercheurs, à cause de sa simplicité de mise en œuvre, sa capacité à exploiter les réponses des classifieurs à combiner sans connaître précisément leurs comportement et de son efficacité prouvée à travers beaucoup de travaux de recherches. La Figure V.4 montre une présentation détaillée des méthodes de combinaison parallèle. Différentes taxonomies des méthodes de combinaison ont été proposées dans la littérature basant sur certain nombre de critères comme le type de sorties des classifieurs combinés, la capacité d'apprentissage des règles de combinaison ou les stratégies de combinaison choisies.

Les taxonomies que nous venons de présenter sont adoptées du travail de Zouari [117], elles permettent de distinguer les méthodes de combinaison en fonction de différents critères (type de sorties des classifieurs, possibilité d'apprentissage de la combinaison, stratégies de combinaison, structure des classifieurs, ...).

Au premier niveau, on distingue les méthodes de sélection des méthodes de fusion. Si dans les méthodes de fusion, tous les classifieurs élémentaires participent à la prise de décision finale, la situation est entièrement différente dans le cas des méthodes de sélection qui consistent à sélectionner le meilleur sous-ensemble de classifieurs en fonction des résultats des classifieurs simples, et la décision finale peut être prise soit par le meilleur classifieur uniquement, soit par plusieurs classifieurs.

Le deuxième niveau, concerne la distinction entre les méthodes de sélection statique et les méthodes de sélection dynamique. À la différence des méthodes statiques qui prennent en compte les mêmes classifieurs pour reconnaître toutes les formes, les méthodes dynamiques sont adaptatives dans la mesure où elles cherchent le meilleur sous-ensemble de classifieurs en fonction des données présentées en entrée.

Dans les méthodes de fusion, on distingue les méthodes dites figées ou non paramétriques, où on combine les sorties des classifieurs dans un schéma dont les paramètres sont invariables ; des méthodes avec apprentissage qui cherchent à apprendre, sur les données disponibles, des paramètres nécessaires à la combinaison.

Enfin, la complexité de ces méthodes peut varier en fonction du type des sorties de classifieurs à combiner (sortie de types classe, rang ou mesure). Dans les sections suivantes, les méthodes de fusion seront présentées selon la taxonomie proposée ci-dessus.

V.6 Combinaison sans apprentissage

Ces méthodes n'utilisent que des informations du premier ordre (sorties de classifieurs), elles sont faciles à implémenter et ne nécessitent pas de phase d'apprentissage. Elles peuvent être divisées en type classe, rang et mesure.

V.6.1 Type classe

Dans cette combinaison, chaque classifieur fournit en sortie une réponse sur l'appartenance de la forme inconnue à une classe. Toutefois, il s'agit de la seule information qu'on pourra utiliser. La combinaison d'un ensemble de classifieurs de ce type de sorties est souvent basée sur le principe du vote.

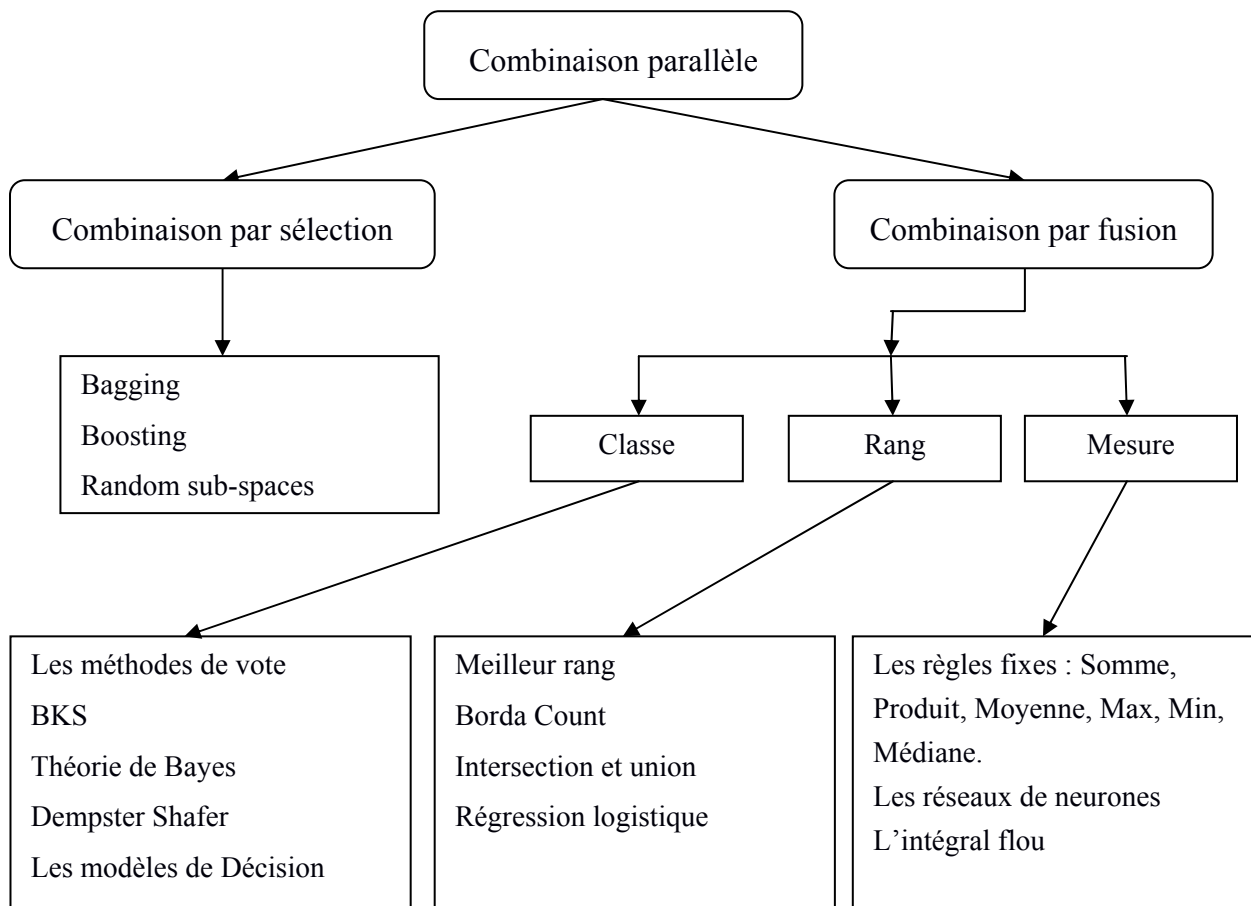


Figure V.4 : Classification des méthodes de combinaison parallèles.

Les stratégies de vote peuvent être appliquées à un système de multiple classifieurs, en supposant que chaque classifieur élémentaire donne une étiquette de classe unique en tant que sortie et aucune de données d'apprentissage sont disponibles. Ces méthodes sont les plus simples à mettre en œuvre, où chaque classifieur donne une proposition de classe ou effectue un rejet exprimant qu'il est incapable de déterminer la classe de la forme à reconnaître.

La plupart des méthodes de vote utilisent un seuil qui représente la proportion des classifieurs devant voter la même classe pour qu'elle soit la décision finale, la classe qui a reçu un nombre de vote supérieur à ce seuil est considérée la décision finale, sinon rejet. En effet, la méthode de vote majoritaire avec seuil est considérée comme le cas général des méthodes de votes, qui peut s'exprimer par :

$$E(x) = \begin{cases} C_i & \text{si } \sum_i e(i) = \max_{c_i \in \{1, \dots, M\}} \sum_j e(j) \geq \lambda K \\ \text{rejet} & \text{sinon} \end{cases} \quad (V.3)$$

Où K est le nombre de classifieurs à combiner et λ est le seuil qui représente la proportion du nombre de classifieurs devant répondre la même classe pour que cette classe soit la classe finale sinon rejet. Si $\lambda = 0$, la classe qui a reçu le plus grand nombre de vote, est considérée la classe finale, sinon rejet. Cette règle s'appelle le vote à la pluralité. Cette méthode a la forme de vote la

plus simple et est donc la plus facile à appliquer. Néanmoins, il y a un rejet si toutes les classes ont le même nombre de votes. Dans ce cas, les risques de conflit sont particulièrement importants.

Dans le cas où $\lambda = 1$, la classe finale est choisie si tous les classifieurs votent cette réponse, sinon la décision finale est un rejet. Cette méthode est appelée majorité unanime et son inconvénient majeur est qu'elle peut produire un taux de reconnaissance assez faible surtout lorsque le nombre de classifieurs augmente.

Si $\lambda \geq 0.5$, plus de la moitié des classifieurs doivent voter la même classe pour qu'elle soit la décision finale, sinon la décision finale est un rejet. C'est à dire si au moins k classifieurs sont d'accords, k pouvant être définie comme suit :

$$k = \begin{cases} \frac{L}{2} \text{ si } L \text{ est pair} \\ \frac{L+1}{2} \text{ si } L \text{ est impair} \end{cases} \quad (\text{V.4})$$

Où L représente le nombre total des classifieurs.

Cette méthode est appelée vote par majorité simple, ou des informations supplémentaires comme les performances individuelles des classifieurs sont utilisées pour résoudre le cas d'un conflit.

Les méthodes présentées ci-dessus, se composent d'une seule étape. Il existe d'autres méthodes de vote qui nécessitent plusieurs étapes. On trouve la procédure run-off et la méthode de Condorcet, qui consistent à utiliser des traitements itératifs au cours des différentes étapes permettant de réduire l'ensemble des classes participant au vote.

V.6.2 Type rang

Les méthodes de type rang ont été développées essentiellement pour remédier aux problèmes des méthodes de vote. Ce type de méthodes consiste, à combiner des listes de propositions de classifieurs ordonnées de manière décroissante selon leur préférence. Les rangs attribués par chaque classifieur élémentaire peuvent être considérés comme des votes pour toutes les classes. Pour un problème de N classes, chaque classifieur attribue un poids N à la classe placée au début de la liste, un poids $N-1$ à la seconde classe, etc. La classe placée à la fin de la liste, reçoit un seul vote. Les méthodes de combinaison de ce type conviennent pour des problèmes multi-classes où la bonne classe peut apparaître dans un rang élevé, mais parfois pas au début de la liste.

Dans cette catégorie de méthodes, on trouve:

– **Méthode d'intersection:** cette méthode, consiste à choisir pour chaque classifieur l'ensemble des classes de meilleurs rangs et à réaliser l'intersection entre tous les ensembles obtenus pour la combinaison. Le nombre de classes choisies pour chaque classifieur, doit être suffisamment grand pour avoir une grande probabilité de présence de la vraie classe dans l'ensemble des résultats [28].

– **Méthode union:** la combinaison par la méthode union est réalisée par l'union des meilleurs candidats des classifieurs. Contrairement à la méthode d'intersection, le nombre de candidats de chaque classifieur doit être suffisamment petit pour ne pas avoir un ensemble trop important de résultats.

– **Méthode du meilleur rang:** La méthode du meilleur rang, consiste à attribuer à chacune des classes le rang le plus élevé parmi les rangs proposés par les classifieurs et d'ordonner la liste selon ces rangs. La classe se trouvant finalement en haut de la liste est choisie comme étant la décision finale. Cette méthode est mieux adaptée pour des problèmes à grand nombre de classes et à faible nombre de classifieurs. Son point fort, est qu'elle utilise la meilleure réponse de chaque classifieur. Cependant, elle présente l'inconvénient de pouvoir facilement provoquer des conflits, lorsque plusieurs classes obtiennent le même meilleur rang. Ce problème peut être résolu en utilisant des critères supplémentaires.

- **Méthode de Borda Count et ses variantes:** La méthode la plus utilisée pour combiner des classifieurs de niveau rang est la méthode de Borda Count proposée par Jean Charles de Borda [122]. Cette méthode est basée sur le calcul du rang total pour chacune des classes proposées. Dans une certaine façon, la méthode de Borda Count peut être considérée comme une généralisation de la règle de vote majoritaire et elle est exactement réduite au vote majoritaire dans le cas d'un problème à deux classes. Lors de l'utilisation de cette méthode, on considère les classifieurs comme des votants et les classes comme des candidats. Etant donné L classifieurs, portant chacun sur N exemples et M classes, on trie tout d'abord les résultats de chaque classifieur, par ordre de rang, en attribuant un nombre de points à chacun. Le candidat préféré reçoit N points (votes), le candidat suivant reçoit $N-1$, et ainsi de suite. Le nombre de points du candidat placé à la fin de la liste est 0. La somme de tous les votes des éléments qui ont la même classe, donne le nombre de points pour cette classe. De cette façon, les nombres de votes pour toutes les classes sont obtenus. On exécute ces étapes pour chaque classifieur et on fait une addition des votes obtenus pour les classifieurs précédents par les différentes classes. Au final, on obtient donc un tableau qui donne le nombre total de votes obtenus par chacune des classes dans les différents classifieurs. La classe finale de la forme entrante, est alors celle qui a obtenu le plus de points (votes). La méthode de Borda count est facile à mettre en œuvre et ne nécessite

aucun apprentissage. Le point faible de cette technique est qu'elle traite tous les classifieurs d'une manière égale et ne prend pas en compte leurs capacités individuelles. Notons que des variantes de cette méthode se sont révélées plus fiables dans des applications spécifiques de classification [123], parmi lesquelles on trouve ; la méthode de Borda Médiane, la méthode de Nanson, et la méthode proposée par Parker [124] appelée wBorda.

V.6.3 Type Mesure

Les méthodes de type mesure, consistent à combiner des mesures qui reflètent le degré de confiance des classifieurs sur l'appartenance des exemples à reconnaître en chacune des classes. Dans cette catégorie de méthodes, on trouve:

- **Règles fixes** : les méthodes des règles fixes sont basées sur le principe que les classifieurs sont indépendants et estiment des probabilités a posteriori des classes [125]. Plusieurs règles de combinaison peuvent être obtenues pour l'estimation de la probabilité a posteriori des classes en utilisant la formule de Bayes. Ainsi pour reconnaître une forme x en utilisant k classifieurs pour un problème de M classe, les probabilités a posteriori $P_{ij}(x), i = 1, k; j = 1, M$ peuvent être calculées par la formule générale:

$$q_i(x) = \frac{q'_j(x)}{\sum_j q'_j(x)} \quad (V.5)$$

La règle de combinaison dans ce cas est :

$$E(x) = \begin{cases} \max_j (q_j(x)) \\ \text{rejet} & \text{Sinon} \end{cases} \quad (V.6)$$

Cette règle de décision finale revient à choisir, parmi l'ensemble des classes les plus probables proposées par chacun des classifieurs, la classe pour laquelle la probabilité a posteriori $q'_j(x)$ est donnée par l'une des règles suivantes :

$$\text{Maximum: } q'_j(x) = \max_{i=1}^k (p_{ij}(x))$$

$$\text{Minimum: } q'_j(x) = \min_{i=1}^k (p_{ij}(x))$$

$$\text{Médiane: } q'_j(x) = \text{med}_{i=1}^k (p_{ij}(x))$$

$$\text{Produit: } q'_j(x) = \prod_{i=1}^k (p_{ij}(x))$$

$$\text{Somme : } q'_j(x) = \sum_{i=1}^k (p_{ij}(x))$$

- **Moyenne de Bayes**

Les méthodes bayésiennes, sont utilisées pour fusionner des sorties de classifieurs exprimées en probabilités a posteriori. Parmi les méthodes de fusion les plus simples à appliquer, on trouve la règle moyenne de Bayes. Ainsi, pour reconnaître une forme x , où chaque classifieur e_j propose une probabilité à postérieure $P_j(x \in C_i/x)$ attribuée à la classe C_i ($i=1, \dots, m$), il est possible de calculer la probabilité à postérieure moyenne à partir de tous les classifieurs :

$$P_{moy}(x \in C_i/x) = \frac{1}{L} \sum_{j=1}^L P_j(x \in C_i/x) \quad (V.7)$$

La décision finale est obtenue selon le critère de Bayes : la forme x est classée C_i si $P_{moy}(x \in C_i/x)$ est maximale. La probabilité à postérieure $P_j(x \in C_i/x)$ peut être directement fournie par les classifieurs de type bayésien.

V.7 Les méthodes de combinaison avec apprentissage

Contrairement aux méthodes précédentes, les méthodes de combinaison avec apprentissage sont plus complexes à mettre en œuvre. Elles déterminent des paramètres supplémentaires de combinaison via une base d'apprentissage ou dite base de validation. Cet apprentissage permet notamment de prendre en compte l'efficacité de chacun des classifieurs de l'ensemble. Ces méthodes donnent en général de bons résultats, mais exigent que la base de validation soit différente de la base d'apprentissage pour éviter le problème de sur-apprentissage et de taille suffisamment importante pour augmenter le pouvoir de généralisation.

V.7.1 Type classe

Les méthodes avec apprentissage de type classe, ne sont pas très nombreuses. Les méthodes les plus utilisées sont principalement le vote pondéré, la théorie de Bayes, la théorie de l'évidence de Dempster-Shafer et la méthode d'espace de connaissance du comportement (Behavior Knowledge Sace ou BKS).

- **Vote pondéré** : ce type de vote consiste à pondérer la réponse de chaque classifieur par un certain coefficient, qui indique son importance dans la combinaison. Ce qui nécessite une étape d'apprentissage ou validation pour calculer les poids associées aux différents classifieurs participants. Il existe plusieurs façons d'estimer ces poids. Ils peuvent être calculés à base de taux de réussite des classifieurs, optimisés par un algorithme génétiques...etc.

- **Méthode de Bayes** : cette méthode repose essentiellement sur la théorie des probabilités. Où les imperfections de l'information et les informations elles-mêmes sont modélisées à partir de

distributions de probabilités ou de mesures statistiques à estimer. L'utilisation de cette méthode, consiste à déterminer la classe C_i pour laquelle la probabilité à postériori $P(C_i/e_i=C_1, \dots, e_l=C_l)$ est maximale, c'est-à-dire :

$$E(x)=C_i \quad \text{Si} \quad P(C_i/e_i=C_1, \dots, e_l=C_l)=\max_{m=1}^N P(C_m/e_i=C_1, \dots, e_l=C_l) \quad (V.8)$$

Plusieurs études supposent que les classifieurs sont indépendants [126, 127], et la probabilité à postériori sous cette hypothèse s'écrit :

$$P(C_i/e_i = C_1, \dots, e_l = C_l) = P(C_i) \prod_{l=1}^L \frac{P(C_i/e_l)}{P(C_i)} \quad (V.9)$$

Les probabilités $P(C_i/e_l = C_l)$ peuvent être déterminées à partir de la matrice de confusion obtenue pour les classifieurs sur une base d'apprentissage. Si on note n_{c_i, c_l}^j le nombre d'éléments de cette base pour lesquels le classifieur e_j attribue les éléments de la classe C_i et $n_{., c_l}^j$ le nombre total des éléments attribués par le classifieur e_j à la classe C_l alors :

$$P(C_i/e_i = C_l) = \frac{n_{c_i, c_l}^j}{n_{., c_l}^j} \quad (V.10)$$

- **Méthode de Dempster Shafer:** appelée aussi théorie de l'évidence, est une théorie mathématique qui permet de représenter à la fois l'imprécision et l'incertitude, à l'aide de la fonction de masse, de la fonction de crédibilité et de la fonction de plausibilité. Cette méthode de combinaison consiste à utiliser comme connaissance à priori la performance de chaque classifieur de base. En effet, La règle DS, combine les différents classifieurs en utilisant leurs taux de reconnaissance et leurs taux de substitution comme connaissance à priori. Ainsi, pour une forme d'entrée x , tous les classifieurs ayant la même sortie, sont rassemblés dans un groupe E_k , $k=1, \dots, k'$ (où k' est le nombre de différentes sorties), qui est équivalent à un nouveau classifieur avec de nouveaux taux de reconnaissance et taux de substitution. Successivement, de l'analyse de l'ensemble de classifieurs équivalents E_k , $k=1, \dots, k'$, deux mesures de croyance, sont calculées (voir [128] pour des détails): la mesure de croyance correcte $Bel(A_j)$ et la sortie mal reconnue $Bel(\neg A_j)$. La réponse finale du classifieur combiné est l'étiquette ω_j de la classe pour laquelle la différence est maximale entre les mesures de croyance pour la sortie correcte et la sortie mal reconnue.

$$Bel(A_j) - Bel(\neg A_j) = \max_{i=1}^N Bel(A_i) - Bel(\neg A_i) \quad | \quad i=0, 1, \dots, m \quad (V.11)$$

- **Méthode de connaissance d'espace de comportement (Behaviour Knowledge Space) :** cette méthode, consiste à utiliser un espace appelé espace de connaissance contenant les décisions de tous les classifieurs. Ce qui permet de connaître le comportement des classifieurs d'où son nom espace de connaissance du comportement. Elle permet alors, de prendre en considération les

informations de cet espace et tente à les intégrer directement dans la règle de décision, afin d'obtenir la solution finale. L'hypothèse d'indépendance de classifieurs n'est pas nécessaire. Si le nombre de classifieurs utilisés est L, l'espace de BKS est de dimension L et chaque classifieur à N décisions possibles (N est le nombre de classes).

La méthode BKS, est constituée de deux étapes : l'une d'apprentissage et l'autre de décision. Les résultats fournis par le L classifieurs permettent durant l'étape d'apprentissage, de construire l'espace de connaissance de comportement (BKS). L'inconvénient majeur de la version originale de BKS, est de tirer aléatoirement l'une des classes lorsqu'une unité focale contient plus d'une classe. Pour remédier à ce problème, On pourra choisir la classe proposée par le classifieur ayant fait le moins d'erreur.

V.7.2 Type rang

Les méthodes de type rang les plus citées dans ce cas sont la somme pondérée, la régression logistique, l'intersection et l'union.

- **Somme pondérée** : Cette méthode peut être considérée, comme une généralisation de la méthode de Borda Count. Elle consiste à pondérer les rangs attribués par les classifieurs pour une classe par des coefficients, indiquant la fonction de la crédibilité ou de la confiance accordée à chaque classifieur. Ces pondérations ou poids sont obtenues par une procédure d'apprentissage. Son inconvénient majeur est le fait d'avoir des poids identiques pour toutes les classes.
- **Régression logistique** : les méthodes de régression logistique, utilisent des informations sur le comportement des classifieurs en combinant des sorties de type rang. Elle est considérée comme une généralisation de la méthode de Borda count en utilisant les poids comme information reflétant l'importance relative des classifieurs. Cette méthode utilise la probabilité $\pi(D_i)$ indiquant que la classe finale de x est la classe C_i avec $0 < \pi(D_i) < 1$ et D_i est le vecteur des réponses de classifieurs pour la classe C_i . La régression logistique utilisée pour estimer cette probabilité, a la forme suivante :

$$\pi(D_i) = \frac{\exp(\beta_0 + \beta_1 r_1 + \beta_2 r_2 + \dots + \beta_k r_k)}{1 + \exp(\beta_0 + \beta_1 r_1 + \beta_2 r_2 + \dots + \beta_k r_k)} \quad (V.12)$$

Les méthodes des moindres carrés ou de maximum de vraisemblance peuvent être utilisées pour l'estimation des paramètres de régression β_i .

Après avoir calculé les paramètres β_i , la combinaison des K vecteurs de rang fournis par les classifieurs, consiste à calculer la probabilité $\pi(D_i)$ en utilisant l'équation (V.12).

Les classes finales de la combinaison seront les classes triées selon les valeurs de $\pi(D_i)$. Finalement la classe ayant la probabilité la plus élevées est sélectionnée.

- **Méthode intersection** : ici, chaque classifieur propose une liste de classes ordonnées selon sa préférence, et les intersections entre ces listes, servent à indiquer le résultat final. c'est-à-dire qu'une classe n'appartient à la liste finale que si elle existe dans toutes les listes fournies par les classifieurs de base. On détermine ici pour chaque élément de la base d'apprentissage le rang maximum permettant de garder la bonne classe dans la liste, ce qui exige d'avoir des listes de grandes tailles afin d'augmenter la probabilité d'avoir la bonne classe au final.

- **Méthode union** : Dans la méthode union, chaque classifieur propose aussi une liste de classes ordonnées selon sa préférence. L'union de ces listes forme le résultat final. Une classe qui existe au moins une fois dans une liste, figure dans la liste finale des solutions. Contrairement à l'intersection, dans la méthode d'union, le nombre de classes de chaque classifieur doit être suffisamment petit pour éviter d'avoir un ensemble trop important dans la liste finale. Une procédure min-max peut être utilisée pour sélectionner les seuils des listes des classifieurs. Cette procédure se base sur les performances des classifieurs obtenues sur la base d'apprentissage.

V.7.3 Type mesure

Les méthodes appartenant au type mesure avec apprentissage sont :

- **Règles pondérées** : ce type de règles, consiste à pondérer les mesures $m_{i,j}$ associées aux classes C_i par des coefficients w_j indiquant la vraisemblance associée à chaque classificateur e_j . Ces coefficients seront déterminés par un apprentissage. Chaque probabilité a posteriori P_i d'une classe C_i peut être obtenue par l'une des règles suivantes :

$$P_i = \lambda \sum_{j=1}^L W_j m_{i,j} \quad (\text{V. 13})$$

$$P_i = \prod_{j=1}^L m_{i,j}^{w_j} \quad (\text{V. 14})$$

w_j est le coefficient qui détermine l'importance attribuée au $j^{\text{ème}}$ classifieur dans la combinaison ($j=1, \dots, L$). Si $\lambda = 1$, la 1^{ère} règle est nommée la somme pondérée. Pour $\lambda = 1/L$ il s'agit de la moyenne pondérée.

- **Réseaux de neurones**: La combinaison de classifieurs au niveau mesure, peut aussi être réalisée en utilisant un réseau de neurones en tant qu'opérateur de décision. Les classifieurs de base apprennent au début sur la base entière ou sur une partie de cette base. Ensuite, une base de validation est classée, en utilisant cet ensemble de classifieurs. La réponse à chaque forme est considérée comme une nouvelle donnée à apprendre par le réseau de neurones ce qui lui donne la capacité d'apprendre le comportement des classifieurs. Chaque forme à reconnaître, est alors classée au début par l'ensemble de classifieurs dont les sorties sont ensuite combinées par le

réseau de neurones, pour fournir une décision finale. Les sorties de ces classifieurs de base, sont donc considérées pour le module de fusion comme un nouvel ensemble de caractéristiques de chaque forme de test. On appelle cette approche méta-classifieur [117].

- Décision Template (DT) : L'idée de base de la méthode de combinaison de décision Template (DT), consiste à caractériser chaque classe par un profil de décision le plus typique, et de classer les nouvelles formes en comparant leur décision profils avec le DT de chaque classe, en utilisant une mesure de similarité. L'algorithme d'apprentissage de Décision Template peut se résumer comme suit:

1. Pour $j=1, \dots, c$, calculer la moyenne des profils de décisions $DP(Z_k)$ de tous les membres de ω_j à partir de l'ensemble de données Z :

$$DT_j = \frac{1}{N} \sum_{\substack{Z_k \in \omega_j \\ Z_k \in Z}} DP(Z_k) \quad (V.15)$$

Où : N_j est le nombre des éléments de Z à partir de ω_j .

2. Donnant $x \in \mathbb{R}^n$, construire $DP(x)$ en calculant la similarité S entre $DP(x)$ et chaque DT_j : $\mu_j(x) = S(DP(x), DT_j) j = 1, \dots, C$.

V.8 Conclusion

Ce chapitre, a permet de présenter un état l'art sur la combinaison de classifieurs. Nous avons exposé également un panorama des approches principales de combinaison à savoir: la combinaison séquentielle, parallèle et hybride. Nous avons présenté ensuite, l'avantage de l'approche parallèle pour la combinaison de classifieurs qui est celle que nous avons choisie dans le cadre de notre travail sur la reconnaissance des mots arabes manuscrits. Le choix des approches de combinaison parallèle est justifié par leur efficacité et leur simplicité à la mise en œuvre. Nous avons mené une étude profonde des différentes techniques et règles de combinaison parallèle des classifieurs.

CHAPITRE VI

Contribution à la reconnaissance de mots arabes manuscrits

Chapitre VI Contribution à la reconnaissance de mots arabes manuscrits

VI.1 Introduction

La reconnaissance de l'écriture manuscrite, est l'un des problèmes de reconnaissance de formes les plus étudiés. Aujourd'hui, et malgré plus de vingt ans de recherches approfondies, il n'existe toujours pas de système fiable capable de traiter l'écriture naturelle dans sa globalité. En effet, les travaux publiés montrent que les bons résultats obtenus, sont restreints à des domaines d'application bien limités (adresses postales, chèques bancaires, formulaires) ou à des catégories d'écriture très contraintes, ne représentant qu'un aspect particulier de l'écriture courante et spontanée.

Ce problème devient plus complexe, en particulier pour la reconnaissance de l'écriture dans des scripts cursifs comme l'arabe. Les défis principaux en reconnaissance d'écriture arabe manuscrite résultent de sa nature fortement cursive, les espaces inter et intra mots non-uniformes, des chevauchements et des ligatures, et un grand nombre de marques et de points diacritiques. La reconnaissance de l'écriture arabe manuscrite a reçu l'intérêt de recherche significatif pendant les deux dernières décennies et on a proposé un certain nombre de systèmes de reconnaissance rapportant des taux de reconnaissance très élevés. Malgré ces développements, le problème reste toujours ouvert pour la recherche et l'expérimentation en raison des défis mentionnés ci-dessus. La combinaison de classificateurs, a attiré significativement l'attention des chercheurs ces dernières années pour améliorer les taux de reconnaissance de différents problèmes de classification des formes, surtout dans le cas de la reconnaissance de l'écriture manuscrite en général et la reconnaissance des mots arabes manuscrits en particulier.

Le travail que nous présentons dans ce chapitre, concerne la mise en place d'un système de reconnaissance hors ligne de mots arabes manuscrits basé sur la combinaison parallèle de classifieurs[129]. L'architecture globale de notre système de reconnaissance, est illustrée par la Figure VI.1.

VI.2 Prétraitement

Le prétraitement est une série d'opérations effectuées, sur l'image du mot pour supprimer le bruit introduit lors de l'étape d'acquisition, éliminer les variations et les irrégularités liées aux styles d'écritures et représenter l'image sous une forme appropriée pour l'étape d'extraction de caractéristiques. Les tâches de prétraitement typiques pour la reconnaissance de l'écriture manuscrite et des problèmes similaires de reconnaissance des formes comprennent, la

binarisation, redressement de la ligne de base et la correction d'inclinaison, la normalisation et la squelettisation. Étant donné que les images dans la base de données IFN / ENIT sont déjà binarisées, le prétraitement dans notre travail comprend la correction de pente et l'inclinaison pour éliminer les variations dépendantes des styles d'écriture, et la squelettisation pour réduire la quantité d'informations à traiter. Le processus de prétraitement est schématisé par la Figure VI.2.

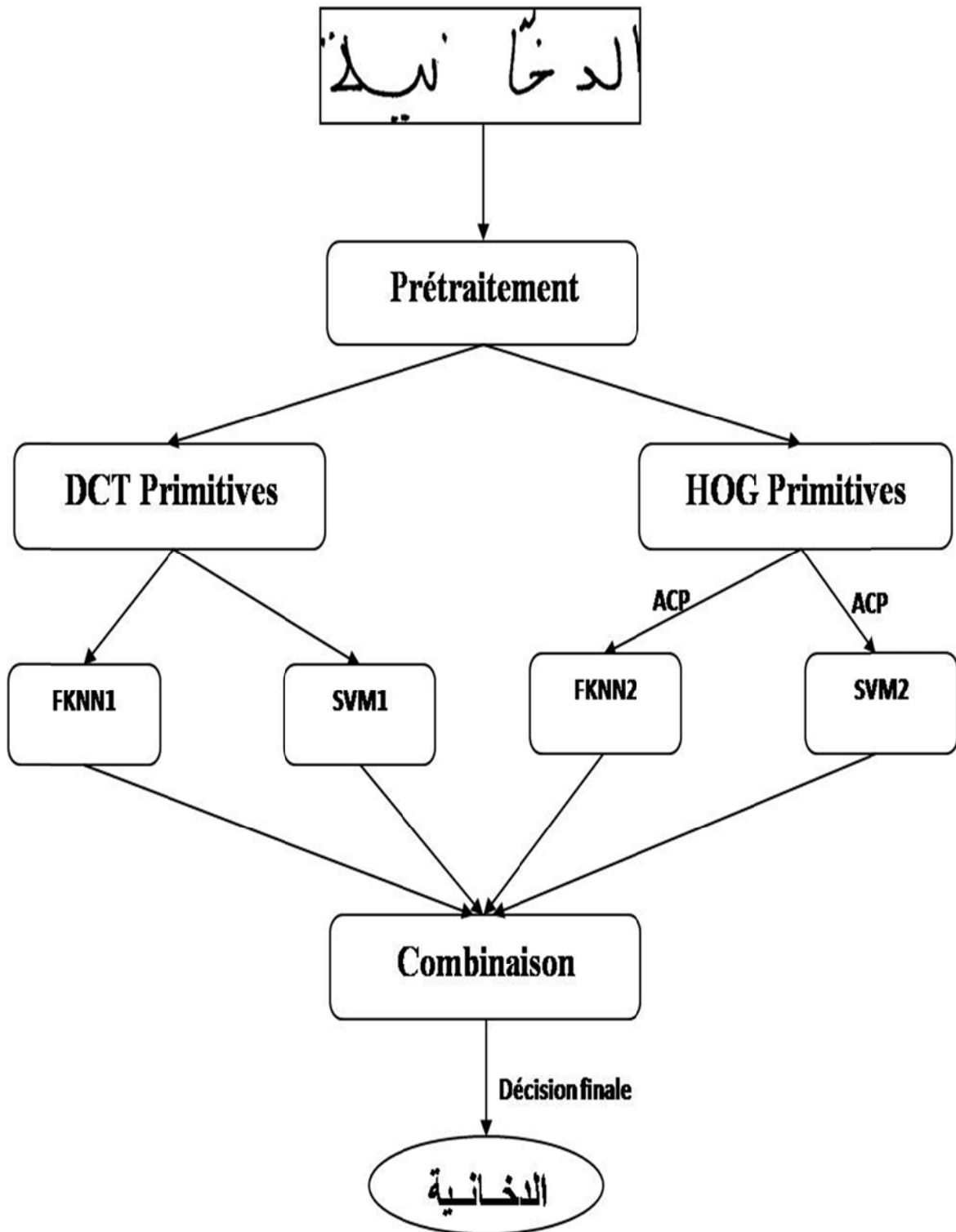


Figure VI.1 : Architecture du système développé.

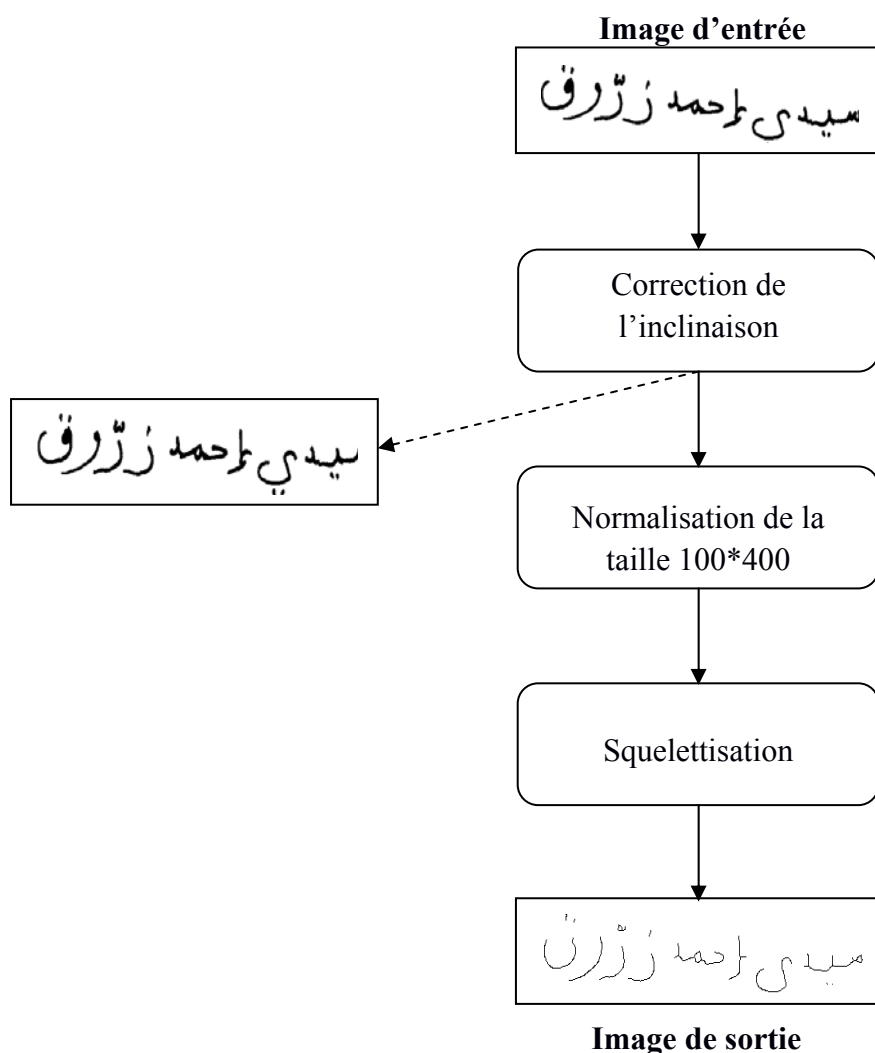


Figure VI.2 : Processus de Prétraitement.

VI.2.1 Correction de l'inclinaison de ligne (Slope correction)

L'inclinaison de la ligne de base du mot est une distorsion qui est souvent présentée pendant la phase de numérisation du support, ou par des défauts d'orientation causés par les habitudes des scribes. L'angle d'inclinaison est l'angle que forme la ligne de base avec l'axe horizontal. Par conséquent, l'objectif de ce prétraitement consiste à mettre horizontalement chaque ligne de l'écriture inclinée, ce qui permet de réduire considérablement la variabilité de l'écriture. Pour redresser les images de mots inclinés, nous avons utilisé une procédure basée sur les histogrammes de projection horizontale adoptée de l'algorithme proposé en [1]. Cette procédure peut se résumer comme suit:

On suppose que la ligne de base est inclinée horizontalement d'un angle entre -15 et +15.

Entrée: squelette de l'image du mot.

Output: image du mot redressée.

Etape1: faire tourner l'image vers la gauche avec certains angles et pour chaque image tournée calculer la valeur de pic de l'histogramme de projection horizontal.

Etape 2: faire tourner l'image vers la droite avec certains angles et pour chaque image tournée calculer la valeur de pic de l'histogramme de projection horizontal.

Etape 3: la comparaison des résultats des étapes 1 et 2, nous indique en quelle direction aura lieu la ligne de base.

Etape 4: faire répéter la rotation de l'image avec des angles dans la direction indiquée dans l'étape 3.

Calculer la valeur maximale de l'histogramme de projection horizontal.

Si la nouvelle valeur maximale > à la valeur maximale précédente

Faire la mise à jour de la valeur maximale

Enregistrer la nouvelle image

Enregistrer le nouvel angle.

Sinon

Incrémenter le compteur de un.

Jusqu'à la valeur maximale des angles sera atteinte (-15 ou +15).

Etape 5: Faire tourner l'image vers le sens contraire de l'angle d'inclinaison calculée dans l'étape 4.

Fin

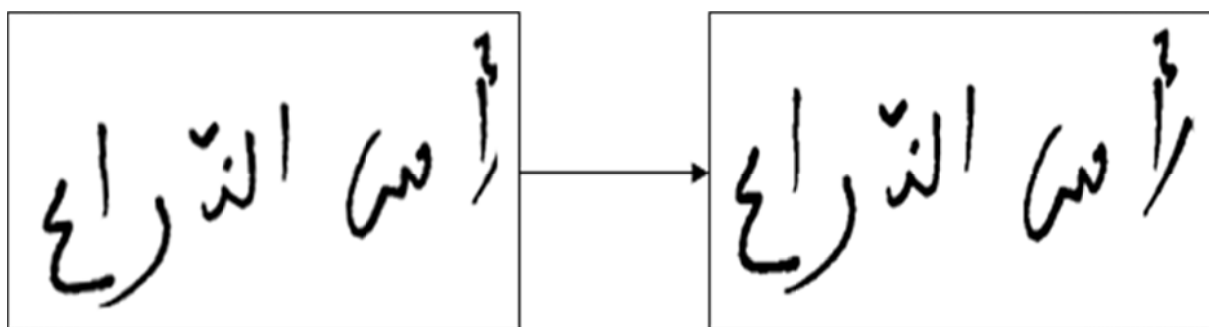


Figure VI.3 : Résultat du slope correction.

VI.2.2 Correction de l'inclinaison des caractères (Slant correction)

L'inclinaison des caractères, est définie comme étant l'angle entre l'axe correspondant à la direction moyenne des caractères et l'axe vertical ; où les lettres peuvent être inclinées vers la droite ou vers la gauche. L'objectif de ce prétraitement, est de transformer le mot de façon à ce que cet axe de direction principale devient vertical. Nous avons proposé une méthode simple pour la correction de l'inclinaison des caractères à base des histogrammes de projection similaire à la méthode proposée pour la correction de l'inclinaison de la ligne de base. Cet algorithme, consiste à utiliser l'histogramme de projection verticale. L'idée est que l'histogramme d'un mot qui est écrit tout droit aura des pics plus grands et plus distincts. Donc, nous pourrions regarder l'histogramme du mot à différents angles d'inclinaison et prendre celui avec les pics les plus grands. Nous le faisons pour des angles entre -20 et 20 degrés, qui est, la gamme la plus commune d'angles d'inclinaison dans l'écriture normale. Pour chaque angle, nous calculons l'histogramme vertical en appliquant une fonction de mesure pour calculer la hauteur des pics. L'angle avec la plus haute mesure, gagnera et sera utilisé comme angle de translation de l'image (shear Transformation) ; pour gagner du temps, nous passons d'abord la gamme avec de grands pas de 4 degrés. Pour chacun de ces angles nous déterminons ceux avec la mesure la plus haute et nous cherchons autour de chacun d'entre eux avec des pas plus petits. Cela donne quelque chose d'équilibre entre la vitesse et la robustesse.

Le but de la procédure de correction de l'inclinaison des lettres, est de rendre le texte plus approprié pour le processus de reconnaissance. Donc, la meilleure façon d'évaluer un algorithme serait de mesurer le changement de la performance du système dans lequel il est incorporé.

VI.2.3 Normalisation de la taille

Pour la normalisation de la taille de l'image du mot, nous avons utilisé une fonction prédéfinie du langage Matlab appelée *Imresize* en lui spécifiant le facteur de normalisation de taille. Ce facteur, est un vecteur contenant le nombre de lignes et de colonnes de l'image de sortie. Ainsi, les images ont toutes la même taille de 100*400 pixels. La fonction *Imresize*, utilise l'interpolation pour déterminer les valeurs des pixels additionnels. La méthode d'interpolation consiste à déterminer la valeur du pixel interpolé, en recherchant le point dans l'image d'entrée qui correspond à un pixel dans l'image de sortie, ensuite on calcule la valeur du pixel de sortie par une moyenne pondérée des pixels voisins de ce point. Cette pondération se base sur la distance séparant chaque pixel du point. Dans notre procédure de normalisation, nous avons utilisé l'interpolation bi-cubique pour déterminer les valeurs des pixels dans l'image de sortie, qui a donné des bons résultats, mais d'autres méthodes d'interpolation peuvent être spécifiées.

VI.2.4 Squelettisation

Cette opération est très importante pour le processus de reconnaissance, car elle permet de réduire la quantité d'information à traiter. Il existe une variété d'algorithmes de squelettisation des images binaires. Dans le cadre de notre projet nous avons choisi l'algorithme de Zhang et Suen[18]. Cet algorithme de squelettisation est une méthode parallèle qui signifie que la nouvelle valeur obtenue dépend uniquement de la valeur de l'itération précédente. Cet algorithme est rapide et simple à mettre en œuvre, et réalisé par deux sous-itérations. Dans la première, un pixel $I(i, j)$ est supprimé si les conditions suivantes sont remplies:

1. Son nombre de connectivité est égal à 1.
2. A au moins deux voisins noirs et pas plus de six.
3. Au moins une des $I(i, j + 1)$, $I(i-1, j)$ et $I(i, j-1)$ sont blancs.
4. Au moins l'un de $I(i-1, j)$, $I(i + 1, j)$ et $I(i, j-1)$ sont blancs.

Dans la deuxième sous-itération les conditions dans les étapes 3 et 4 changent.

1. Son nombre de connectivité est égal à 1.
2. A au moins deux voisins noirs et pas plus de six.
3. Au moins l'un de $I(i-1, j)$, $I(i, j + 1)$ et $I(i + 1, j)$ sont blancs.
4. Au moins une des $I(i, j + 1)$, $I(i + 1, j)$ et $I(i, j-1)$ sont blancs.

Finalement, les pixels satisfaisant ces conditions seront supprimées. Si à la fin de l'une des sous-itérations, il n'y a pas de pixels à supprimer, alors l'algorithme s'arrête.

Le TableauVI.1 présente quelques exemples du processus de squelettisation à l'aide de l'algorithme de Zhang-Suen appliqué à des images des mots arabes manuscrits.

Image originale	Image squelette

Tableau VI.1 : Squelettes des images de mots arabes par l'algorithme de Zhang-Suen.

VI.3 Extraction de primitives

L'extraction de caractéristiques, est probablement le composant le plus important de n'importe quel système de reconnaissance de formes, qui a pour but de trouver une représentation appropriée de la forme à reconnaître. Pour notre problème de reconnaissance de mots arabes manuscrits, nous avons voulu représenter les images des mots par des caractéristiques discriminatives. Pour chaque image de mot, nous extrayons un ensemble de caractéristiques basées, sur la transformée en cosinus discrète (DCT) et l'histogramme de gradients orientés (HOG). Chacune de ces familles de caractéristiques est discutée dans les sections suivantes.

VI.3.1 Extraction des caractéristiques par la transformée de DCT

Après l'application de la DCT sur l'image du mot, le vecteur de primitives est conçu par la lecture zigzag de la matrice de DCT (voire la Figure VI.4). Ceci arrange les coefficients dans un vecteur de 1D de telle façon que les coefficients de basses fréquences contenant des informations pertinentes (appropriées) soient au début du vecteur qui sont utiles dans la reconstitution de l'image originale. En appliquant DCT, l'image du mot à reconnaître est représentée par ce vecteur. Ainsi, l'une des caractéristiques principales de la DCT est sa capacité à convertir l'énergie de l'image dans quelques coefficients. Le nombre de ces coefficients de DCT, est déterminé expérimentalement après plusieurs tests.

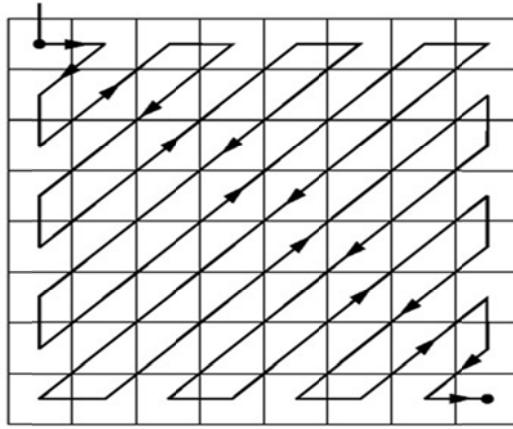


Figure VI.4 : La lecture Zigzag de la matrice de DCT.

VI.3.2 Extraction des caractéristiques par PCA-HOG

L'histogramme de Gradients Orientés, ou HOG, est un descripteur de caractéristiques basé sur le gradient que l'on a à l'origine introduit par Dalal et Triggs pour la détection des humains. Depuis, il a été largement employé à un certain nombre de problèmes de reconnaissance de formes. La procédure que nous avons adoptée pour le calcul du descripteur de HOG est inspirée du travail de [130] pour la détection des humains. Les étapes suivantes décrivent la méthode de construire le descripteur de HOG:

Etape 1 : calcul du gradient, il s'agit de calculer le gradient pour tous les points de l'image ; usuellement deux masques de dérivation sont appliqués sur l'image, un horizontal $[-1, 0, 1]$ et vertical $[-1, 0, 1]^T$.

Etape 2 : La norme et l'orientation du gradient sont calculés en utilisant les équations suivantes:

$$NG = \sqrt{G_h(x, y)^2 + G_v(x, y)^2} \quad (VI.1)$$

$$OG = \arctan \frac{G_h(x, y)}{G_v(x, y)} \quad (VI.2)$$

Où: G_h et G_v représentent le gradient horizontal et vertical; NG et OG représentent la norme et l'orientation respectivement au point (x, y) dans l'image.

Etape 3 : L'image est divisée en $N*N$ cellules et l'histogramme des orientations est calculé pour chaque cellule. Si l'histogramme est divisé en B barres (bins) en fonction de l'orientation, la valeur du $i^{\text{ème}}$ bin V_i pour la cellule C est calculée de la manière suivante:

$$V_i = \sum_{(x, y) \in C} NG(x, y) / OG(x, y) \in \text{bin}_i \quad (VI.3)$$

Etape 4 : L'histogramme de chaque cellule est normalisé.

Etape 5 : Les histogrammes de toutes les cellules sont concaténés pour former le descripteur.

La dimension du descripteur est en fonction de la taille de la cellule N et le nombre d'intervalles dans l'histogramme B (bins).

Dans notre implémentation de l'algorithme, nous divisons chaque image de mot en cellules qui varient de $3*3$ à $15*15$ tandis que l'histogramme est également divisé en $([0, \pi])$ intervalles. Le descripteur final a une taille de $N*N*B$, Qui dans notre cas varie de 81 à 1296 pour des scénarios d'évaluation différents. Afin de réduire la taille du vecteur de caractéristiques nous appliquons la méthode de l'analyse en composante principale (ACP) sur le descripteur de HOG.

VI.4 Classification

La classification est effectuée, pour déterminer la classe de l'image de mot à reconnaître. Les caractéristiques extraites des exemples d'apprentissage (mots dans notre cas) sont utilisés pour former le classifieur à apprendre à distinguer entre les différentes classes. Dans notre étude, nous cherchons à étudier les différentes combinaisons de classificateurs et d'analyser leurs performances respectives sur la reconnaissance des mots. Nous avons choisi la machine à vecteurs de support (SVM) et k-plus proche voisin flou (F-KNN) comme classifieurs de base, qui sont combinés en utilisant différentes règles de combinaison. Des primitives basées sur la DCT et HOG sont calculées à partir des images de la base d'apprentissage, seront présentées à chacun des classifieurs, soit un total de quatre classifieurs. Dans les sections suivantes, nous discutons brièvement de chacun des classifieurs (SVM et F-KNN) et de présenter plus tard, les règles de combinaison.

VI.4.1 K plus proche voisin flou

Le classifieur k plus proche voisin flou est une amélioration de la méthode de classification classique k plus proche voisin, basée sur la théorie des sous-ensembles flous. Le seul paramètre à optimiser durant l'utilisation de ce type de classifieur, est le nombre k qui serait déterminé empiriquement après plusieurs tests en basant sur le taux de reconnaissance.

VI.4.2 SVM multi-classes

Pour la classification SVM multi-classes, nous avons utilisé une librairie nommée LIBSVM qui regroupe un ensemble d'algorithmes de fouille de données (classement, régression...), toutes issues de l'approche par les machines à support de vecteurs (Support Vector Machine), développées par Chih-Jen Lin et les membres de son équipe [131]. Nous avons utilisé la méthode C-SVM (SVM pour l'apprentissage supervisé multi-classes). Pour la classification multi-classes, cette librairie utilise l'approche un contre un (1VS1). Cela nécessite l'apprentissage de $N*(N-1) / 2$ classifieur SVM binaire chacun entraîné sur les primitives extraites des exemples de la base d'apprentissage de deux classes. La classe du mot à reconnaître est désignée selon la stratégie de vote "Max-win". Le classifieur SVM a deux paramètres de réglage, la pénalité (C) et le paramètre de noyau RBF (γ). La recherche de paramètres adaptés est appelée sélection de

modèle, et ses résultats influent fortement sur les performances du classifieur. Les valeurs de ces paramètres ont été déterminées par la méthode 'grid search' en utilisant la méthode de 5-validation croisées. La méthode "grid search", où une recherche systématique consiste à discrétiser l'espace des paramètres à l'aide d'un pas fixe plus ou moins grand. Différentes valeurs de (C, γ) ont été testées et celui au meilleur score de validation croisée est sélectionné pour l'étape de test.

VI.4.3 Règles de combinaison parallèle

Les différentes règles de combinaison de classifieurs employées dans notre système de reconnaissance de mots arabes manuscrits, comprennent le vote majoritaire, les règles: minimum, maximum, somme, moyenne, produit, décision Template, méthode bayésienne et la règle de Dempster-Shafer. Chacune de ces règles de combinaison est brièvement discutée ci-après.

- **vote majoritaire:** la forme à reconnaître est affectée à la classe avec le nombre de votes le plus élevé par l'ensemble de classifieurs.
- **Règle maximum:** Comme son nom l'indique, cette règle prend simplement le maximum des sorties individuelles des classifieurs pour chaque classe et affecte la forme d'entrée à la classe avec le score maximum.
- **Règle minimum:** cette règle consiste à trouver le score minimum de chaque classe entre les classifieurs et affecte la forme d'entrée à la classe avec le score maximum parmi les scores maximums.
- **Règle Somme:** collecte les scores fournis par chaque classifieur de base et attribue la classe avec le score maximum à la forme d'entrée.
- **Règle Produit:** multiplie le score fourni par chaque classifieur de base et la forme d'entrée à la classe le maximum score.
- **Règle moyenne:** trouve la moyenne des scores de chaque classe entre l'ensemble des classifieurs et affecte la forme d'entrée à la classe, avec le score maximum parmi les moyennes.
- **Modèle de décision (Template decision):** Un modèle de décision est créé pour chaque classe et la forme d'entrée est classée en comparant son profil de décision avec les modèles de décision de chaque classe, en utilisant une mesure de similarité.
- **La théorie de Dempster-Shafer:** Inspirée de la théorie de l'évidence de Dempster-Shafer (DS), cette règle emploie des modèles de décision (Template decision) avec un degré de croyance plutôt que d'une mesure de similarité.

- **Règle de Bayes:** Suppose que les classifieurs sont mutuellement indépendants et combine les sorties au niveau abstrait, en utilisant les matrices de confusion des classifieurs membres.

VI.5 Résultats expérimentaux

les performances du système de reconnaissance proposé sont évalués en conduisant une série d'expériences sur la base de données bien connu des mots arabes manuscrits IFN/ENIT [31] comprenant 26,459 mots manuscrits de 946 noms de ville/village tunisiens. Les échantillons d'écriture sont la contribution de 411 auteurs différents et la base de données est divisée en quatre sous-ensembles 'a' 'b', 'c', et 'd'. Trois sous-ensembles 'a' , 'b', et 'c' sont utilisés pour la phase d'apprentissage et le sous-ensemble 'd' est utilisé pour la phase de test. Pour toutes les expériences, chaque image de mot est normalisée à une taille prédéfinie de 100 *400 pixels.

Le Tableau VI.2 montre quelques images de mots arabes manuscrits, extraites de la base IFN/ENIT.

image	Code postal	étiquette
	9112	الفايض
	6051	نحال
	1251	الشرايع
	2241	رأس الذراع

Tableau VI.2 : Quelques images de mots arabes manuscrits extraites de la base IFN/ENIT.

Nous présentons d'abord les résultats des expériences effectuées sur les classifieurs individuels, ensuite nous présentons les résultats obtenus de différentes règles de combinaison de ces classifieurs. Finalement, nous présentons une comparaison des résultats réalisés par l'architecture de combinaison proposée avec des travaux réalisés dans ce domaine.

VI.5.1 Performances des classificateurs individuels

Comme indiqué précédemment, les primitives basées sur la DCT et HOG sont séparément introduits dans deux classifieurs différents, SVM et F-KNN pour construire un ensemble de quatre classifieurs différents. Les résultats sont rapportés en faisant varier le nombre de coefficients DCT dans le cas de caractéristiques à base DCT et en faisant varier le nombre de cellules (N) et le nombre de bins (B) dans le cas de descripteurs de HOG. Les paramètres de réglage pour les deux classifieurs sont déterminés expérimentalement. Le SVM a deux paramètres, la pénalité (C) et le paramètre de noyau RBF (γ). Le classifieur F-KNN a un seul paramètre, le nombre de voisins les plus proches (k) qui est également trouvé expérimentalement.

VI.5.1.1 Le Premier classifieur (k plus proche voisins flou avec la DCT)

D'abord nous fixons le nombre de voisins les plus proches k à 7 et nous varions le nombre de coefficients de DCT. L'avantage majeur de la DCT, est qu'elle compresse l'énergie de l'image en quelques coefficients de DCT, le Tableau VI.3 montre l'effet du nombre de DCT sur le taux de reconnaissance du classifieur F-KNN.

Nombre de coefficients de DCT (k=7)	Taux de reconnaissance %	Taux d'erreur%
400	59.09	40.91
200	78.34	21.66
100	80.22	19.78
80	82.60	17.40
70	82.91	17.09
60	83.28	16.72
65	82.94	17.06
63	83.08	16.92
50	82.57	17.43
40	81.27	18.73
30	80.47	19.53

Tableau VI.3 : Taux de reconnaissance de FKNN1 en fonction du nombre de coefficients de DCT.

D'après les résultats présentés dans le tableau ci-dessus le meilleur taux de reconnaissance est 83,28 % pour 60 coefficients de DCT. Ce qui montre, le grand pouvoir de la DCT à compresser les informations pertinentes en quelques coefficients. Après avoir fixé le nombre de coefficients de DCT égal à 60, la 2^{ème} expérience consiste à étudier la variation du taux de reconnaissance en fonction du k.

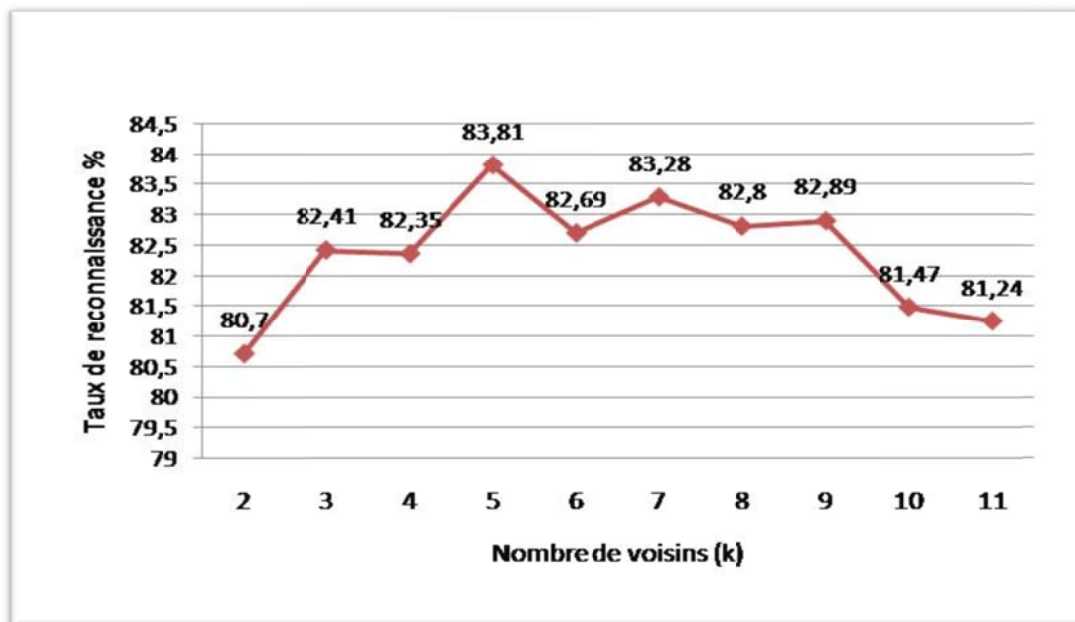


Figure VI.5 : Taux de reconnaissance du FKNN1 en Fonction du nombre de voisins k.

La Figure ci-dessus représente un graphe qui exprime la variation du taux de reconnaissance en fonction du nombre de voisins k. Ce taux varie entre 80,70% pour k=2 et 83,81% pour k=5.

VI.5.1.2 Le deuxième classifieur (k plus proche voisins flou avec le HOG)

Le calcul de HOG descripteur nécessite le réglage de deux paramètres importants, le nombre de cellules par boîte englobante (N) et le nombre d'intervalles (bin) d'orientation (B) produisant un descripteur de dimension $N * N * B$. Pour les expériences, nous considérons un certain nombre de configurations pour les paramètres (N, B). En outre, l'analyse en composantes principales (ACP) est également appliquée sur les caractéristiques du HOG pour réduire la dimensionnalité du descripteur. Le nombre de composantes principales est déterminé empiriquement après plusieurs tests. La première expérience consiste à étudier l'effet du nombre de cellules (N) sur le taux de reconnaissance des mots arabes manuscrits en fixant le nombre de bins (B) à 8 et le nombre de voisins du classifieur FKNN k à 7. Le tableau ci-dessous montre les résultats obtenus.

PCA-HOG	% de PCA	Taux de reconnaissance %	Taux d'erreur%
3*3*8	100%	47.76	52.24
4*4*8	100%	62.34	37.66
5*5*8	40%	72.74	27.26
6*6*8	35%	79.02	20.98
7*7*8	33%	82.84	17.16
8*8*8	30%	87.33	12.67
9*9*8	30%	89.19	10.81
10*10*8	28%	89.60	10.40
11*11*8	25%	90.05	09.95
12*12*8	30%	89.47	10.53
13*13*8	30%	88.43	11.57

Tableau VI.4 : Variation du taux de reconnaissance du FKNN2 en fonction du nombre de Cellules N.

On peut observer du Tableau VI.4 qu'en général, le taux de reconnaissance augmente avec l'augmentation du nombre de cellules (N). Le taux de reconnaissance le plus élevé réalisé est 90,05% pour N=11. Après cette valeur de N le taux de reconnaissance commence à diminuer si on augmente le nombre de cellules. La deuxième expérience consiste à étudier la variation du taux de reconnaissance de mots arabes manuscrits en fonction de la valeur de bins (B), en fixant le nombre de cellules (N) à 11 et le nombre de voisins du classifieur k à 7.

PCA-HOG	Taux de reconnaissance %	Taux d'erreur%
11*11*4	87.74	12.26
11*11*5	88.64	11.36
11*11*6	89.78	10.22
11*11*7	90.04	09.96
11*11*8	90.05	09.87
11*11*9	90.13	09.87
11*11*10	89.90	10.10
11*11*11	88.65	11.35
11*11*12	88.21	11.79
11*11*13	87.34	12.66

Tableau VI.5 : Variation du taux de reconnaissance du FKNN2 en fonction du nombre de bins(B).

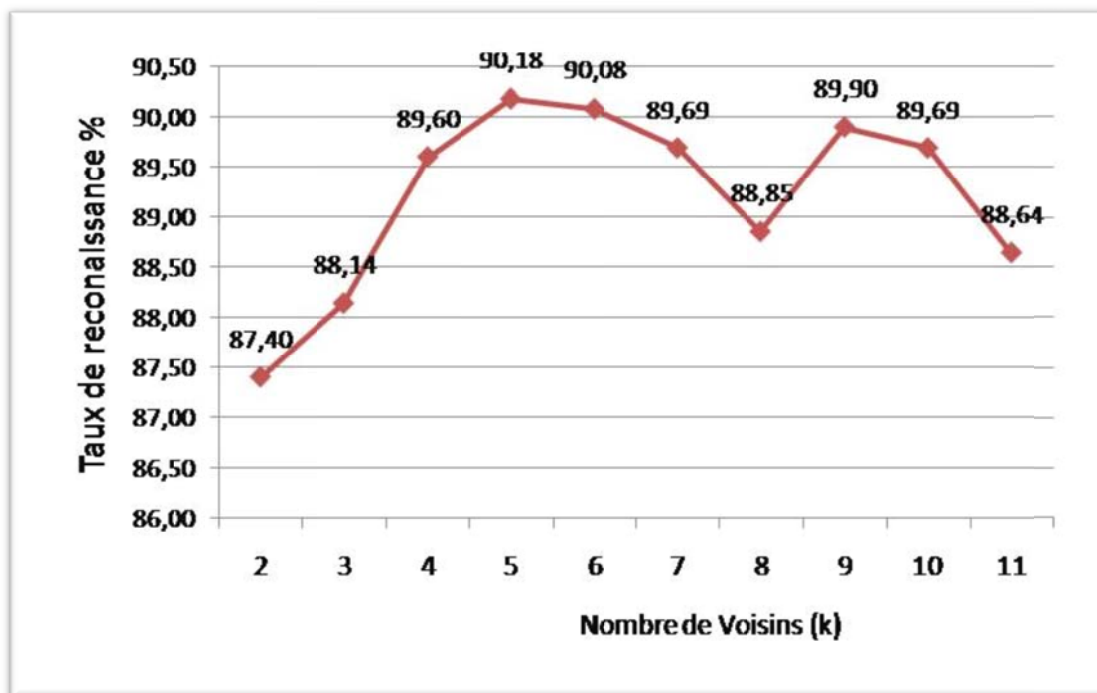
De ce tableau, on peut constater que le taux de reconnaissance augmente faiblement en fonction de l'augmentation du nombre de bins (B), et les meilleurs taux de reconnaissance ont été enregistrés pour B, variant entre 7 et 10. Le taux de reconnaissance le plus élevé est 90,13 % pour B=9.

Le pourcentage de PCA-HOG coefficients à retenir est réalisé empiriquement. Le tableau ci-dessous montre un exemple de détermination de la valeur de pourcentage de PCA-HOG à retenir pour N=11 et B=8 et k=7.

% De coefficients de PCA	Taux de reconnaissance %	Taux d'erreur%	Temps d'exécution en minutes
100%	89.08	10.92	29.57
90%	89.08	10.92	26.25
80%	89.29	10.71	23.90
75%	89.31	10.69	21.92
70%	89.21	10.79	19.81
65%	89.29	10.71	21.40
60%	89.23	10.77	18.67
55%	89.29	10.71	16.92
50%	89.34	10.66	16.45
45%	89.52	10.48	16.06
40%	89.85	10.15	14.81
35%	89.90	10.10	10.65
30%	89.97	10.03	07.92
25%	90.05	09.95	06.01
20%	87.02	12.98	05.67
10%	83.07	16.93	02.77

Tableau VI.6 : Variation du taux de reconnaissance du FKNN2 en fonction du % de PCA.

De ce tableau on peut voir facilement que l'application de l'ACP sur le HOG descripteur permet la réduction considérable de la taille de descripteur sans perdre son pouvoir discriminant. Cette réduction de la taille de descripteur est de $(11 \times 11 \times 8 = 968)$ à 242 coefficients de PCA-HOG (25%). Ce qui conduit à une grande réduction du temps d'exécution pendant la phase de reconnaissance de 29.57 minutes pour tout le descripteur (100% de PCA) à 6.01 minutes pour 25% de PCA avec un taux de reconnaissance égal à 90.05. La Figure VI.6 présente un graphe qui exprime la variation du taux de reconnaissance en fonction du nombre de voisins (k).



FigureVI.6 : Variation de taux de reconnaissance du FKNN2 en fonction du nombre de k.

VI.5.1.3 Le Troisième classifieur (SVM avec DCT)

Le troisième classifieur, est de type SVM multi classes, a pour entrée les coefficients de DCT. Le nombre de ces coefficients de DCT est fixé expérimentalement après plusieurs tests. Nous commençons par 400 coefficients de DCT et nous diminuons ce nombre en remarquant le taux de reconnaissance. Les hyper-paramètres C et γ du SVM, sont fixés empiriquement après plusieurs tests de couple (C, γ) par la méthode 5 cross-Validation. Où le couple qui réalise le meilleur taux de validation sera choisi pour la phase d'apprentissage et la phase de test. Le Tableau VI.7 montre la variation du taux de reconnaissance de mots arabes manuscrits en fonction du nombre de coefficients de DCT, et le Tableau VI.8 illustre la procédure effectuées pour la recherche de meilleur couple (C, γ) par la méthode de 5 cross-validation. De ces expériences, le meilleurs taux de reconnaissance réalisé est égal à 90,43% pour 61 coefficients de DCT et pour $C=100$ et $\gamma=0.05$.

Nombre de coefficients de DCT	Taux de reconnaissance %	Taux d'erreur%
400	42.50	57.50
300	49.33	50.67
200	86.87	13.13
100	88.20	11.80
90	89.31	10.69
80	89.46	10.54
70	89.43	10.57
60	90.21	09.79
65	90.12	09.88
64	90.23	09.77
63	90.27	09.73
62	90.31	09.69
61	90.43	09.57
50	90.13	09.87
40	88.79	11.21
30	85.42	14.58

Tableau VI.7 : Taux de reconnaissance du classifieur SVM1 en fonction du nombre de coefficients de DCT.

Configuration(c, γ)	Taux de validation %	Configuration(c, γ)	Taux de validation %
(1, 0,01)	84.21	(100,0.05)	89.49
(1,0.02)	87.55	(100,0.06)	88.82
(1,0.03)	88.51	(100,0.07)	89.05
(1, 0.04)	88.66	(100,0.08)	88.94
(1,0.07)	88.73	(1000,0.008)	87.44
(10,0.01)	87.96	(1000,0.01)	87.96
(10,0.02)	88,84	(1000,0.02)	88.93
(10,0.03)	89.25	(1000,0.03)	89.12
(10,0.04)	89.36	(1000, 0.04)	89.23
(100,0.01)	87.75	(1000,0.06)	89.19
(100,0.02)	88.77	(1000,0.08)	88.85
(100,0.03)	89.27	(5000,0.02)	88.77
(100,0.04)	89.42	(5000, 0.08)	88.23

Tableau VI.8 : Exemple de détermination des valeurs de C et γ pour le classifieur SVM1.

VI.5.1. 4 Le quatrième classifieur (SVM avec HOG)

Le quatrième classifieur, est un classifieur SVM multi-classes, il a pour entrée, le descripteur de HOG. Donc le but des expériences ci-dessous, est de trouver la meilleur configuration du classifieur SVM et la meilleur configuration du HOG descripteur pour réaliser un taux de reconnaissance satisfaisant des mots arabes manuscrits. La première expérience consiste à étudier la variation du taux de reconnaissance en fonction du nombre de cellules (N) en fixant le nombre de bins B à 8. Le Tableau VI.9 illustre les résultats obtenus.

PCA-HOG	Taux de reconnaissance %	Taux d'erreur%
3*3*8	67.23	32.77
4*4*8	80.72	19.28
5*5*8	87.30	12.70
6*6*8	90.00	10.00
7*7*8	92.40	07.60
8*8*8	93.44	06.56
9*9*8	93.78	06.22
10*10*8	94.33	05.67
11*11*8	94.03	05.97
12*12*8	93.65	06.35
13*13*8	92.63	07.37
14*14*8	91.83	08.17
15*15*8	89.64	10.36

Tableau VI.9 : Taux de reconnaissance du classifieur SVM2 en fonction du nombre de cellules (N).

On peut observer du Tableau VI.9 qu'en général, le taux de reconnaissance augmente avec l'augmentation du nombre de cellules (N). Le taux de reconnaissance le plus élevé réalisé est 94,33% pour N=10. Après cette valeur de N le taux de reconnaissance commence à diminuer si on augmente le nombre de cellules.

Après avoir fixé le nombre de cellules (N) à 10, on va étudier la variation du taux de reconnaissance des mots arabes manuscrits en fonction du nombre de bins (B). Les résultats obtenus seront présentés dans le tableau suivant :

PCA-HOG	% de PCA	Taux de reconnaissance %	Taux d'erreur%
10*10*4	100%	93.26	06.74
10*10*5	60%	93.86	06.14
10*10*6	45%	93.82	06.18
10*10*7	40%	94.13	05.87
10*10*8	35%	94.33	05.67
10*10*9	30%	94.24	05.76
10*10*10	28%	94.16	05.84
10*10*11	25%	94.08	05.92
10*10*12	25%	93.78	06.22
10*10*13	25%	93.26	06.74
10*10*14	25%	92.16	07.84

Tableau VI.10 : Taux de reconnaissance du classifieur SVM2 en fonction du nombre de bins (B).

% De coefficients de PCA	Taux de reconnaissance %	Temps - d'apprentissage	Temps de test En minute
100%	94.17	42.00	2.96
90%	94.13	29.00	2.16
80%	94.19	25.01	1.95
75%	94.21	23.43	2.21
70%	94.17	23.00	1.86
65%	94.17	20.85	1.85
60%	94.19	18.58	1.54
55%	94.13	16.10	1.52
50%	94.22	15.15	1.32
45%	94.23	13.35	1.23
40%	94.12	10.88	1.11
35%	94.33	08.86	01.09
30%	94.24	07.25	0.96
25%	94.04	05.84	0.85
20%	93.84	04.29	0.73
10%	91.43	1.88	0.51

Tableau VI.11 : Taux de reconnaissance du classifieur SVM2 en fonction du % de coefficients de PCA-HOG.

Du Tableau VI.10 on peut constater que le taux de reconnaissance augmente avec l'augmentation du nombre de bins (B). Cependant, au-delà de 8 - 10 bins, les taux de reconnaissance se stabilisent et ne présentent pas de variations significatives. Du Tableau VI.11 on peut observer aussi que l'application de l'ACP sur le HOG descripteur permet la réduction considérable de la taille de descripteur sans perdre son pouvoir discriminant. Cette réduction de la taille de descripteur, est de ($10*10*8=800$) à 280 coefficients de PCA-HOG (35%). Ce qui conduit à une grande réduction du temps d'exécution pendant la phase d'apprentissage de 42 minutes pour tout le descripteur (100% de PCA) à 8.86 minutes pour 35% de PCA et pour la phase de test de 02.96 minutes à 1.09 minutes avec un taux de reconnaissance de 94,33.

Le Tableau VI.12 illustre, la procédure effectuée pour la recherche du meilleur couple (C, γ) par la méthode de 5 cross-validation. De ces expériences, le meilleur taux de reconnaissance réalisé, est égal à 94.33% pour le descripteur de HOG ($10*10*8$) avec $C=100$ et $\gamma=0.02$.

Configuration(c, γ)	Taux de validation %	Configuration(c, γ)	Taux de validation %
(1, 0,01)	86.06	(1000,0.008)	92.08
(1,0.02)	87.60	(1000,0.01)	92.81
(1,0.03)	88.47	(1000,0.02)	92.86
(1, 0.04)	88.84	(1000,0.03)	92.64
(1,0.07)	88.76	(1000, 0.04)	92.08
(10,0.01)	88.21	(1000,0.05)	91.84
(10,0.02)	89.05	(1000,0.06)	91.63
(10,0.03)	89.41	(1000,0.07)	91.54
(10,0.04)	89.53	(1000,0.08)	91.37
(10,0.08)	88.84	(1000,0.008)	92.08
(100,0.02)	93.02	(1000,0.01)	92.81
(100,0.03)	92.61	(1000,0.02)	92.86
(100,0.04)	92.48	(1000,0.03)	92.64
(100,0.05)	92.35	(1000, 0.04)	92.08
(100,0.06)	92.24	(1000,0.05)	92.03
(100,0.07)	91.84	(5000,0.02)	92.78
(100,0.08)	91.67	(5000, 0.08)	91.33

Tableau VI.12 : Exemple de détermination des valeurs de C et γ pour le classifieur SVM2.

La Figure VI.7 résume les taux de reconnaissance les plus élevés obtenus par la DCT et les descripteurs de HOG, en utilisant les deux classifieurs SVM et F-KNN.

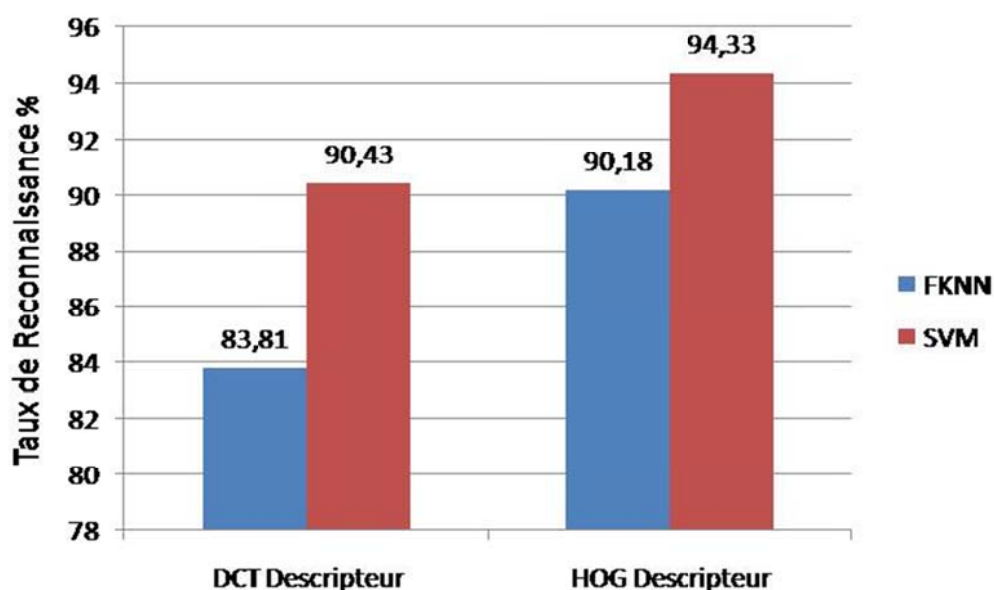


Figure VI.7 : Les taux de reconnaissance maximums des deux classifieurs.

Parmi les deux types de primitives, les caractéristiques à base du HOG surpassent les caractéristiques à base du DCT pour les deux classifieurs. En comparant les deux classifieurs, le classifieur SVM obtient de meilleurs taux de reconnaissance pour les deux familles de caractéristiques.

Dans l'ensemble, le taux de reconnaissance le plus élevé de 94,33% est obtenu en utilisant des descripteurs de HOG et SVM comme classifieur.

Le tableau suivant présente une comparaison des temps d'exécutions des quatre classifieurs pour les phases d'apprentissage et de test.

Classifieurs	Temps	
	Temps d'apprentissage en minutes	Temps de test en minutes
FKNN1	01.22	
FKNN2	06.01	
SVM1	2.50	0.44
SVM2	08.86	1.09

Tableau VI.13 : Comparaison du temps d'exécution des classifieurs.

VI.5.2 Performances de l'ensemble des classifieurs

Après avoir présenté, les performances des classifieurs individuels, nous présentons les résultats de la combinaison parallèle des sorties des quatre classifieurs. Les classifieurs sont combinées en utilisant les règles suivantes : le vote majoritaire, la règle minimum, maximum, somme, moyenne, la règle du produit, modèle de décision, méthode bayésienne et la règle de Dempster-Shafer tel que présenté dans la section VI.4.3. Les taux de reconnaissance de ces combinaisons sont résumés dans la Figure VI.8.

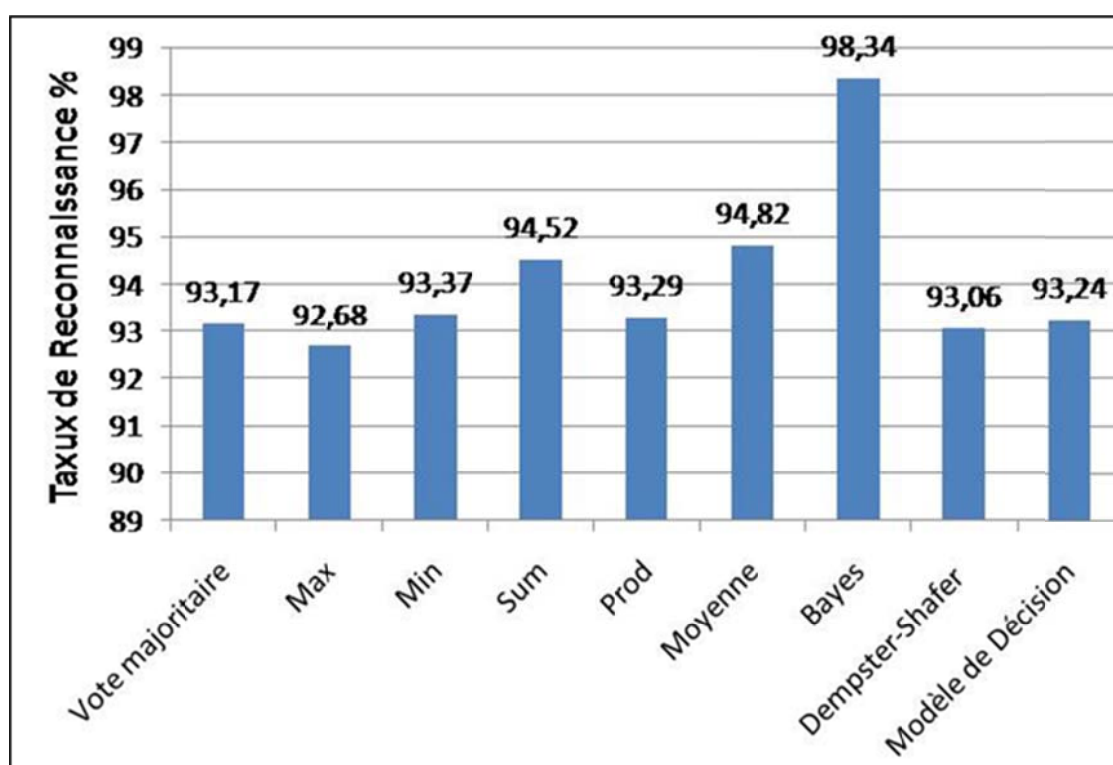


Figure VI.8 : Les taux de reconnaissance de l'ensemble des classifieurs.

À partir de la Figure VI.8, on peut observer que la combinaison de classificateur en utilisant les trois règles : somme, moyenne et la règle de Bayes, a permet d'obtenir de meilleurs taux de reconnaissance par rapport au taux de reconnaissance le plus élevé parmi les classificateurs individuels. L'amélioration la plus significative qu'on peut voir, est dans le cas de la méthode de Bayes qui réalise un taux de reconnaissance de 98,34% contre 94,33% réalisé avec le descripteur de HOG et le classifieur SVM. C'est-à-dire que l'application de la règle de combinaison de Bayes a permet l'amélioration du taux de reconnaissance de mots arabes manuscrits de 4,01% par rapport au meilleur taux de reconnaissance des classifieurs individuels.

VI. 5.3 Comparaison de nos résultats avec d'autres systèmes

Nous présentons également une analyse comparative du système de reconnaissance proposé, avec les travaux de recherche les plus récents dans le domaine de la reconnaissance hors ligne des mots arabes manuscrits. Le Tableau VI.14, présente un aperçu comparatif des différents systèmes évalués sur la même base de données (IFN / ENIT), en utilisant le même protocole d'évaluation que celui utilisé dans notre étude. On peut voir, clairement que la combinaison de classifieurs proposée atteint un taux de reconnaissance maximum de 98,34%, mieux que n'importe quel système existant, validant ainsi, les idées présentées dans cette étude.

Système	Caractéristiques	Classifieur	Taux de Reconnaissance %
Abdel Azeem [91]	Gradient et concavités	Fusion de 3HMMs	97.70
Alalshekmubarak [88]	Technique de zoning	SVM	92.34
Al-Hajj [84]	Distribution et concavités primitives	Combinaison des HMMs	90.96
Alkhateeb [83]	DCT	Réseaux de Neurones	80.75
Benouareth [80]	Primitives statistiques et structurelles	HMM	83.79
L. Chergui [90]	Hu, Zernike et Moments de Tchebychev	Combinaison de réseaux de neurones	90.10
El Abed [78]	Squelette directions et Primitives locales	HMM	89.10
Khalifa [132]	DCT	SVM	91.70
Système proposé	DCT+HOG	Combinaison des SVM and F-KNN	98.34

Tableau VI.14 : Comparaison des systèmes de reconnaissance de mots arabes manuscrits.

Code du mot	libellé	Image réelle	FKNN1	FKNN2	SVM1	SVM2	Décision finale
1251	الشرايع		1251	1251	1251	1251	1251
9174	الرضاع		9174	9174	9174	9174	9174
4134	شماخ		4134	4134	4134	4134	4134
4022	أكودة		8161	4022	7060	4022	4022
1013	المنزه 9		2091	2091	2091	1013	2091
6013	سمباط		9080	6013	9080	6013	6013
8041	قربص		4164	4164	4164	8041	8041
1134	شواط		1134	1134	1134	1134	1134
5154	المحارزة 18		4159	4159	5154	2053	4159
9112	الفايض		9112	9112	9112	9112	9112
3191	رقادة		3191	1111	3191	3191	3191

Tableau VI.15 : Exemples de l'application du système proposé sur des mots arabes manuscrits.

VI.6 Conclusion

Dans ce chapitre, nous avons présenté un système de reconnaissance hors ligne de mots arabes manuscrits basé sur une combinaison parallèle de classifieurs. Ce système opère en quatre grandes phases : prétraitement, extraction des primitives, classification et module de combinaison.

En premier lieu, nous avons introduit les opérations de prétraitements effectuées sur les images des mots afin de les préparer pour l'étape d'extraction de primitives. Dans cette dernière, on a choisi d'utiliser deux types de descripteurs : les coefficients de la DCT, et les descripteurs des histogrammes de gradients orientés (HOG). Le système combine deux types de classifieurs ; un k plus proches voisins flou (F-KNN), et un séparateur à vaste marge (SVM). Chaque classifieur reçoit en entrée, un vecteur de caractéristiques d'un type de descripteur distinct, donnant un total de quatre classifieurs différents. Le système de reconnaissance proposé a été évalué sur la base de données de mots manuscrits arabes, largement utilisée IFN / ENIT. Parmi les classifieurs individuels, SVM a obtenu le taux de reconnaissance le plus élevé de 94,33% en utilisant les descripteurs de HOG. Les quatre classifieurs, ont été ensuite combinés en utilisant différents schémas de combinaison comprenant le vote majoritaire, la règle minimale, la règle maximale, la règle de somme, la règle moyenne, la règle de produit, le modèle de décision, la méthode de Bayes et la règle de Dempster-Shafer.

Les évaluations de ces règles de combinaison, ont révélé que la combinaison en utilisant la règle de bayes permet d'obtenir un taux de reconnaissance des mots de 98,34%, soit environ 4% supérieur au meilleur taux de reconnaissance des classifieurs individuels. Une comparaison de l'approche proposée avec des systèmes de reconnaissance de mots arabes manuscrits, testés aussi sur la base de données IFN/ENIT, montre une supériorité du système proposé sur les méthodes existantes dans la littérature en termes de taux de reconnaissance.

CHAPITRE VII

Nouvel Algorithme de segmentation de l'écriture arabe manuscrite basé sur une approche de Validation SVM

VII Nouvel Algorithme de segmentation de l'écriture arabe manuscrite basé sur une approche de Validation SVM

VII.1 Introduction

La segmentation en caractères, est considérée comme un processus essentiel qui intervient dans toute approche analytique de reconnaissance de l'écriture manuscrite. Car un échec du processus de segmentation provoque une classification erronée des caractères. De toute évidence, le cas le plus difficile dans la segmentation de caractères, est le script cursif tel que l'écriture arabe manuscrite. Sa nature calligraphique se distingue des autres scripts de plusieurs façons. La raison principale d'obtenir de faibles taux de reconnaissance, est expliquée par la particularité du manuscrit arabe. À la différence d'autres langues, les caractéristiques morphologiques de l'écriture arabe manuscrite sont la cause principale de l'échec de son traitement. Dû à la complexité de l'écriture arabe, la segmentation automatique des caractères, devient une tâche très ardue, la segmentation en caractères du texte cursif est la plus dure, la plus cruciale et la plus gourmande en temps de calcul de toutes les étapes d'un système de reconnaissance de l'écriture manuscrite. Dans ce chapitre nous présentons un nouvel algorithme de segmentation de l'écriture arabe manuscrite, basé sur une approche de validation SVM [133].

VII.2 Architecture de l'algorithme proposé

L'objectif que nous nous sommes assignés, s'articule autour du développement d'un nouvel algorithme de segmentation des mots arabes manuscrits, basé sur une approche de validation SVM. La Figure (VII.1), montre la méthodologie, que nous avons adoptée pour la conception de notre algorithme.

Dans un premier temps une phase de prétraitement est réalisée afin d'éliminer autant que possible le bruit, et d'autre part de réduire au maximum la variabilité liée aux styles d'écriture. Suivie d'une étape de détection de la ligne de base qui joue un rôle primordial dans le processus de segmentation. Une étape d'isolation des composantes principales basée sur une procédure d'étiquetage de pixels et quelques règles morphologiques concernant la taille de la composante connexe, sa position par rapport à la ligne de base et le recouvrement d'une composante connexe par une autre. Ensuite, une étape de détermination des points de segmentation primaires (PSP) et un Arabic Morphological Segmenater (AMS) et enfin une étape de validation des (PSPs) basée sur un SVM mono classe.

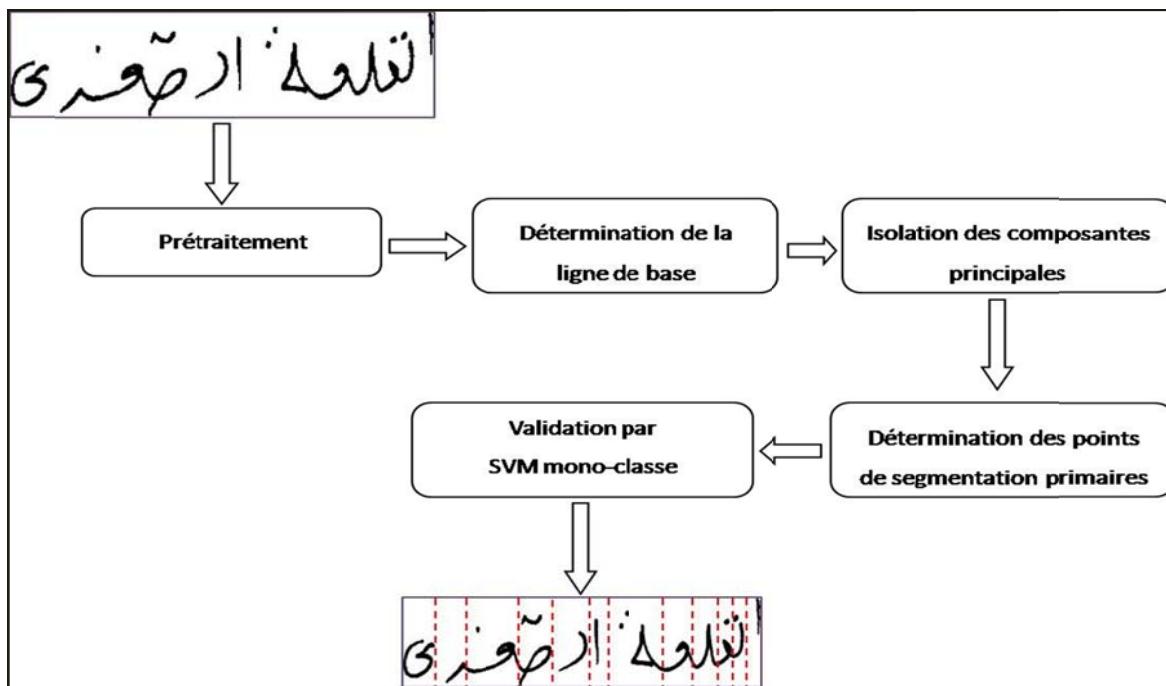


Figure VII.1 : Architecture de l'algorithme proposé.

VII.2.1 Prétraitement

Le prétraitement est une série d'opérations appliquées sur l'image du mot permettent, d'une part, d'éliminer ou de réduire le bruit dans l'image, et d'autre part, de simplifier les traitements ultérieurs. La phase de prétraitement dans l'algorithme proposé comporte les opérations suivantes : Correction de l'inclinaison des caractères et l'inclinaison de l'écriture, normalisation de la taille, et squelettisation. La Figure suivante, montre les opérations de prétraitement effectuées dans notre système:

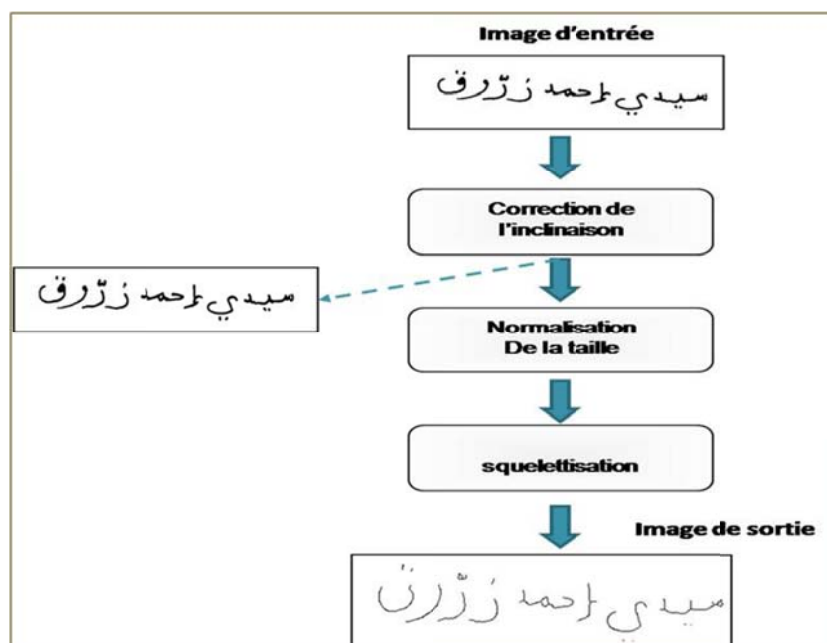


Figure VII.2 : Processus de prétraitement proposé.

VII.2.2 Détection de la ligne de base

La détection de la ligne de base joue un rôle primordial dans la majorité des systèmes de reconnaissance de l'écriture, elle influe fortement sur l'efficacité du processus de segmentation en caractères ; et par conséquent sur les performances du système global. Nous avons proposé une version améliorée de l'algorithme basée sur les histogrammes de projection horizontale qui se résume comme suit:

Entrées

Squelette du mot

Début

Filtrer les composantes connexes <30 pixels

Calculer l'histogramme de projection horizontale

Trouver les 05 premiers pics

Pour chaque composante connexe

Pour chaque valeur de pic

Calculer la distance entre chaque valeur de pic et le point le plus proche de la CC

Calculer la valeur moyenne de toutes les distances

La ligne de base est la valeur de pic avec la moyenne la plus petite.

Fin

Sorties : la ligne de base.

La Figure (VI.3) montre un exemple de la détection de la ligne de base d'un mot arabe manuscrit basée sur l'algorithme décrit ci-dessus.

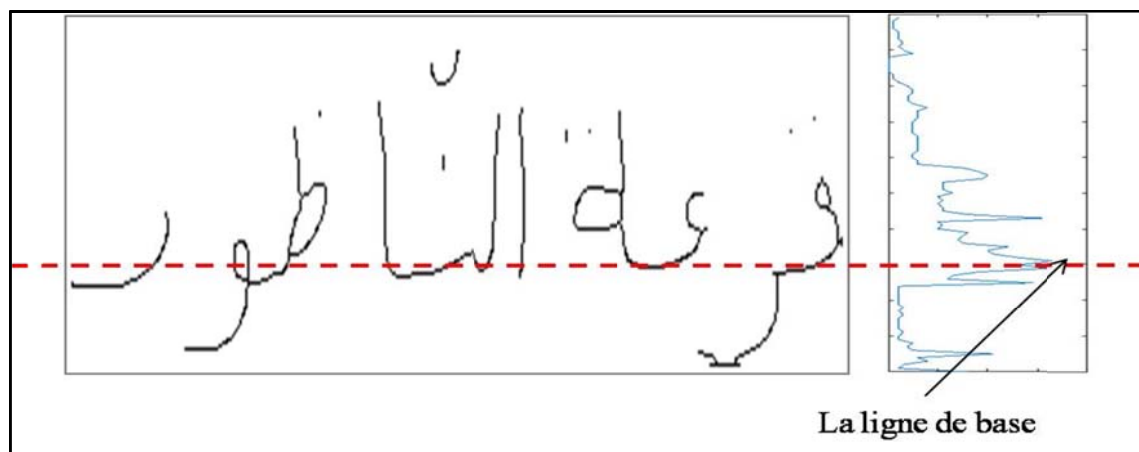


Figure VII.3 : Détection de la ligne de base.

VII.2.3 Filtrage des signes diacritiques

Cette opération, consiste à distinguer les composantes principales des composantes secondaires ou signes diacritiques, qui peuvent beaucoup perturber le processus de segmentation. Cette séparation est effectuée en utilisant la procédure d'étiquetage de pixels et quelques règles morphologiques concernant des critères assez simples: taille, aire, superposition verticale. L'objectif de ce filtrage, est de rejeter la plupart des signes diacritiques, sans rejeter les composantes principales qui peuvent être des lettres ou des pseudo-mots. La Figure (VII.4), décrit la procédure proposée pour le filtrage de signes diacritiques.

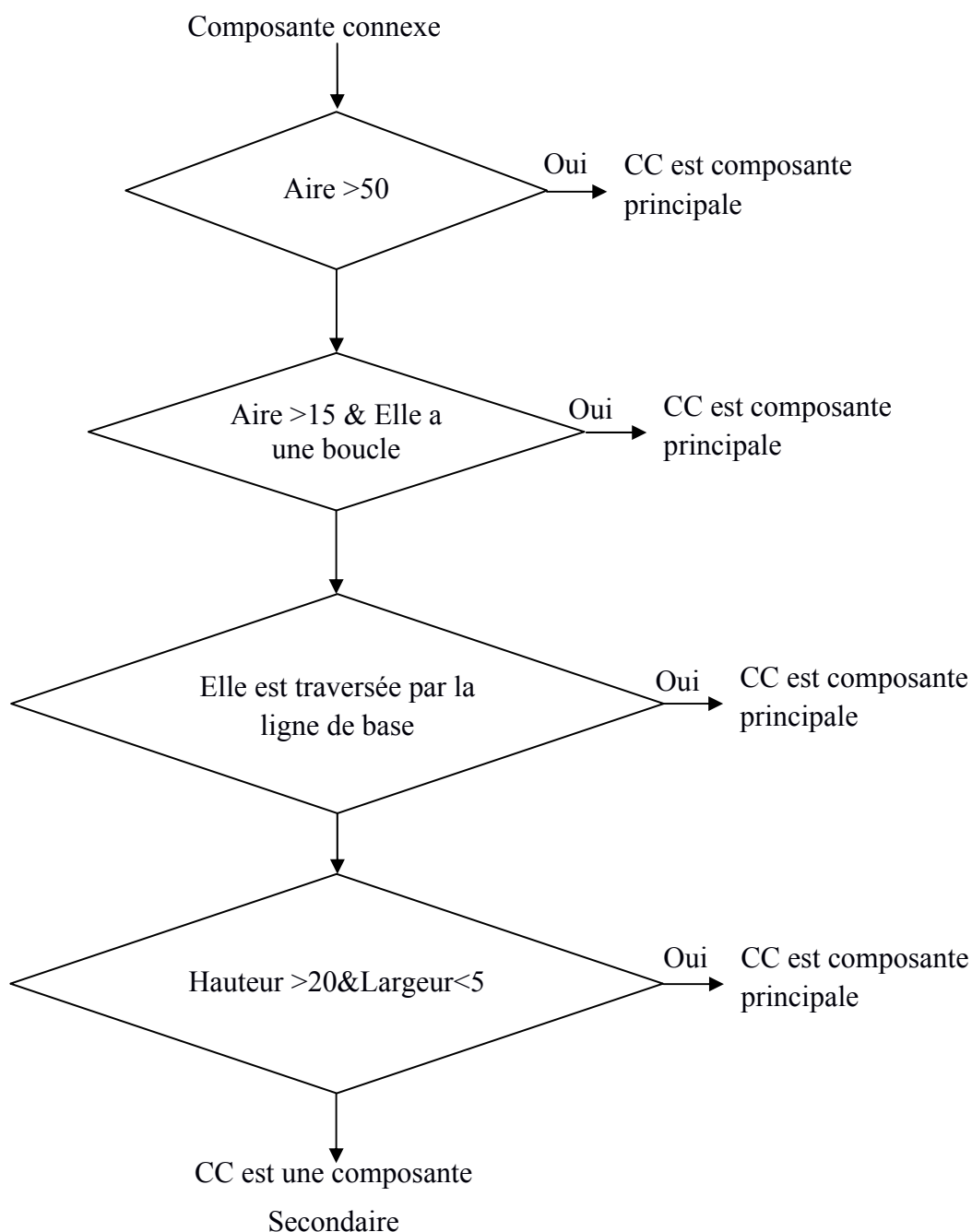


Figure VII.4 : Procédure de filtrage des signes diacritiques.

La figure suivante montre un exemple de la procédure de filtrage appliquée sur des mots arabes manuscrits :

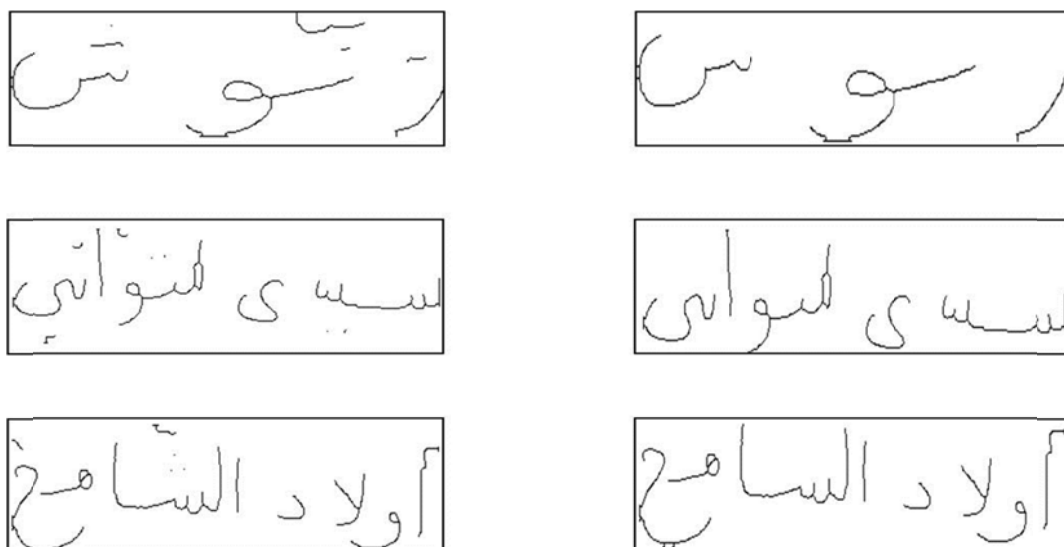


Figure VII.5 : Exemples de la procédure de filtrage des signes diacritiques.

VII.2.4 Détermination des points de segmentation primaires

La détermination des points de segmentation primaires (PSP), est réalisée principalement à l'aide de la recherche des minimums locaux dans l'histogramme de projection verticale de chaque composante connexe principale de l'image du mot. Cette technique se base sur le principe que les séparations entre les caractères manuscrits se manifestent alors sur l'histogramme de projection par la présence de minimums locaux, du moment que la liaison entre les caractères est plus mince. La Figure VII.6, montre un exemple de la procédure de détermination des PSPs d'un mot arabe manuscrit.

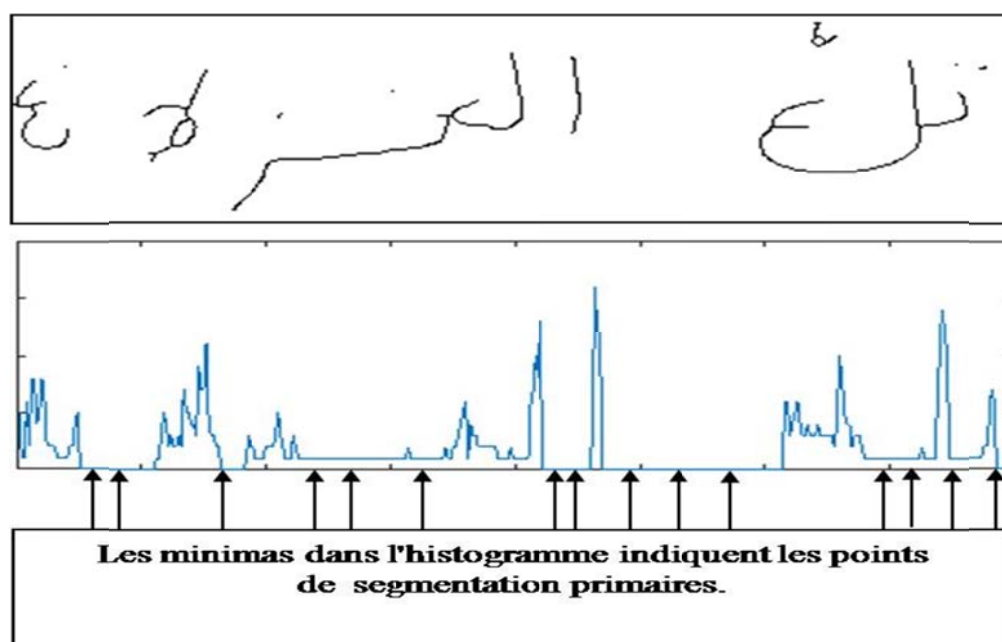


Figure VII.6 : Exemple de détermination des points PSP.

VII.2.5 Arabic Morphological Segmenater (AMS)

Cette appellation est inspirée du travail du Hamad [67]. L'AMS est un ensemble de règles morphologiques qui permettent de réduire autant que possible le nombre de points de segmentation primaires (PSP). Un point PSP est rejeté si :

Règle 1 : Il coupe une boucle.

Règle 2 : Un point PSP est un point extrême ou point de branchement.

Règle 3 : Un point PSP est très proche du début ou de la fin d'une composante principale

Règle 4 : Le point PSP est loin de la ligne de base.

Règle 5 : Un point PSP est très proche d'un point de segmentation réel.

Règle 6 : Il vient en dernier et n'est pas suivi de n'importe quel ascendant ou tout autre descendant.

Règle 7 : S'il y a deux points PSP et qu'il n'y a pas de points caractéristiques entre les deux, Supprimez de la liste celui de gauche.

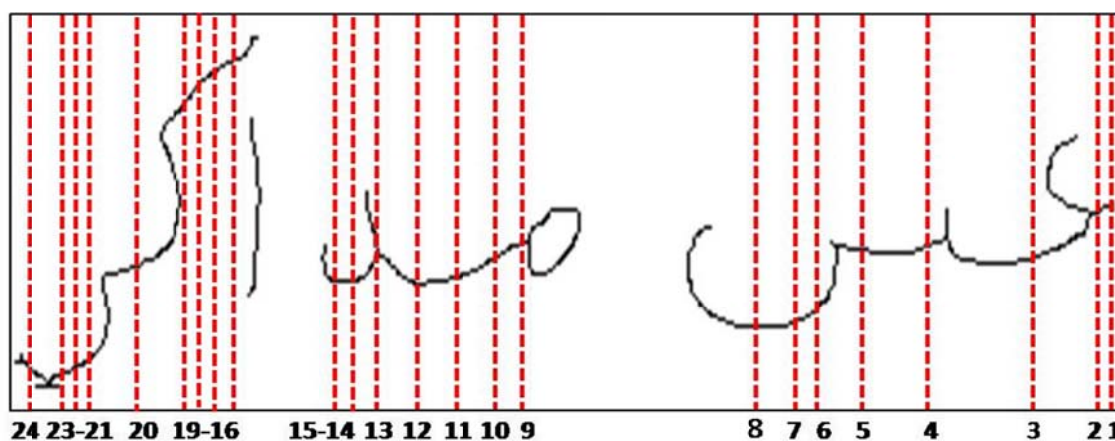


Figure VII.7 : Procédure d'application de L'AMS.

- Les PSPs 1-2 rejetés (R3)
- Les PSPs 3 et 5 sont acceptés.
- Le PSP 4 est rejeté (R7).
- Les PSPs 6,7 et 8 sont rejetés (R4).
- Le PSP 9 est accepté.
- Les PSPs 10-11 sont rejetés (R7).
- Le PSP 12 est rejeté (R7).
- PSP 13 est rejeté (R2).
- Les PSPs 14-15 sont rejetés (R4).
- Les PSPs 16-19 sont rejetés (R4).
- Le PSP 20 est accepté.
- Les PSP 21-23 sont rejetés (R4).
- Le PSP 24 est rejeté (R5).

L'application de l'Arabic Morphological Segmenater (AMS) a permis de réduire le nombre de segmentations primaires de l'image du mot arabe ci-dessus de 24 points à 5 points seulement. La Figure VII.8 montre le résultat de l'application de cette procédure.

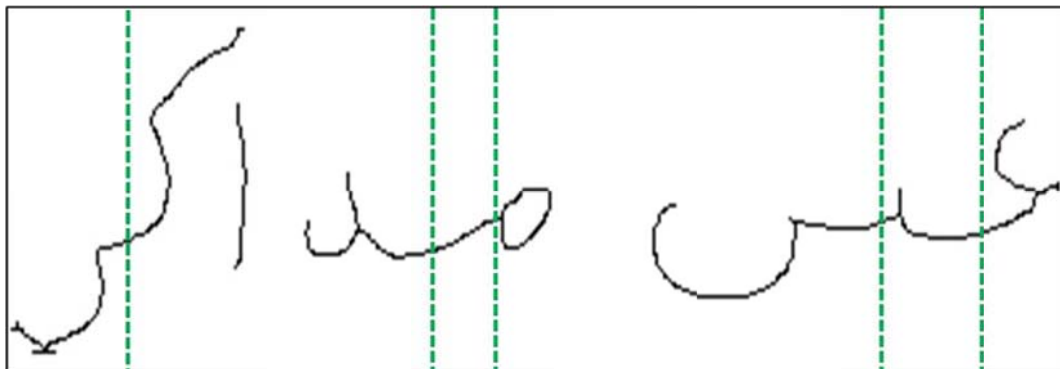


Figure VII. 8 : Résultat de l'application de l'AMS.

VII.2.6 Validation par SVM

Cette étape a pour objectif, de valider les points de segmentations primaires fournies par le module AMS pour garder que les points de segmentations réels et enlever les points de segmentation erronés. Cette étape nécessite une phase d'apprentissage et une méthode d'extraction de primitives basée sur les histogrammes des gradients orientés (HOG). L'extraction de primitives est effectuée sur chaque région de segmentation, déterminée par les points de segmentations primaires, fournies par le module AMS. La Figure suivante, montre un exemple de la détermination des régions de segmentation d'un mot arabe manuscrit.

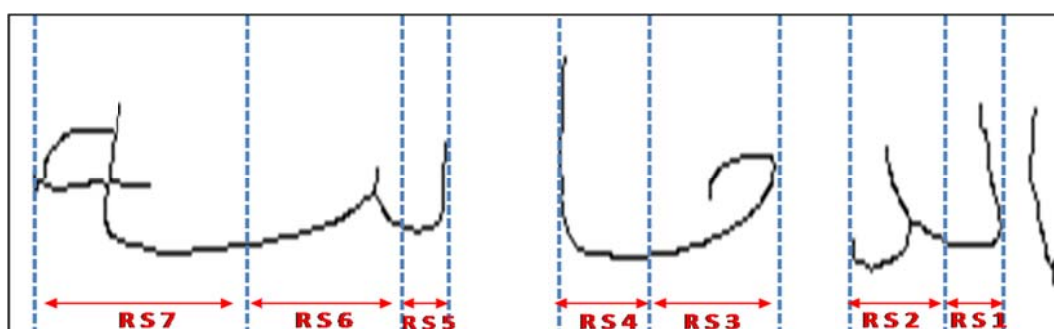


Figure VII.9 : Exemple de détermination des régions de segmentation d'un mot arabe manuscrit.

La région de segmentation est extraite du point de segmentation actuel et le point de segmentation précédant. La base d'apprentissage contient 600 régions de segmentation couvrant l'ensemble des caractères arabes. La validation des points de segmentation, est réalisée en utilisant la bibliothèque LIBSVM [131]. Les hyper paramètres (ν) du SVM mono classe et le

paramètre de noyau RBF (γ), sont fixés empiriquement après plusieurs tests du couple (ν, γ) par la méthode 5-Validation croisée (5-cross-validation). Où le couple qui réalise le meilleur taux de validation sera choisi pour la phase d'apprentissage et la phase de test. Le point de segmentation primaire est accepté comme un point de segmentation réel si la fonction de décision $f(x)$ est supérieure à un seuil prédéfini (t) c'est-à-dire $f(x) \geq t$. Dans le cas contraire le point de segmentation est rejeté. Le seuil (t) est défini par l'équation suivante :

$$t = m + k\sigma \quad (\text{VII.1})$$

Où: m et σ représentent la moyenne et l'écart-type calculés à partir de toutes les valeurs des fonctions de décision lors de la phase d'apprentissage, et le paramètre k permet le contrôle du seuil de décision.

VII.3 Résultats expérimentaux

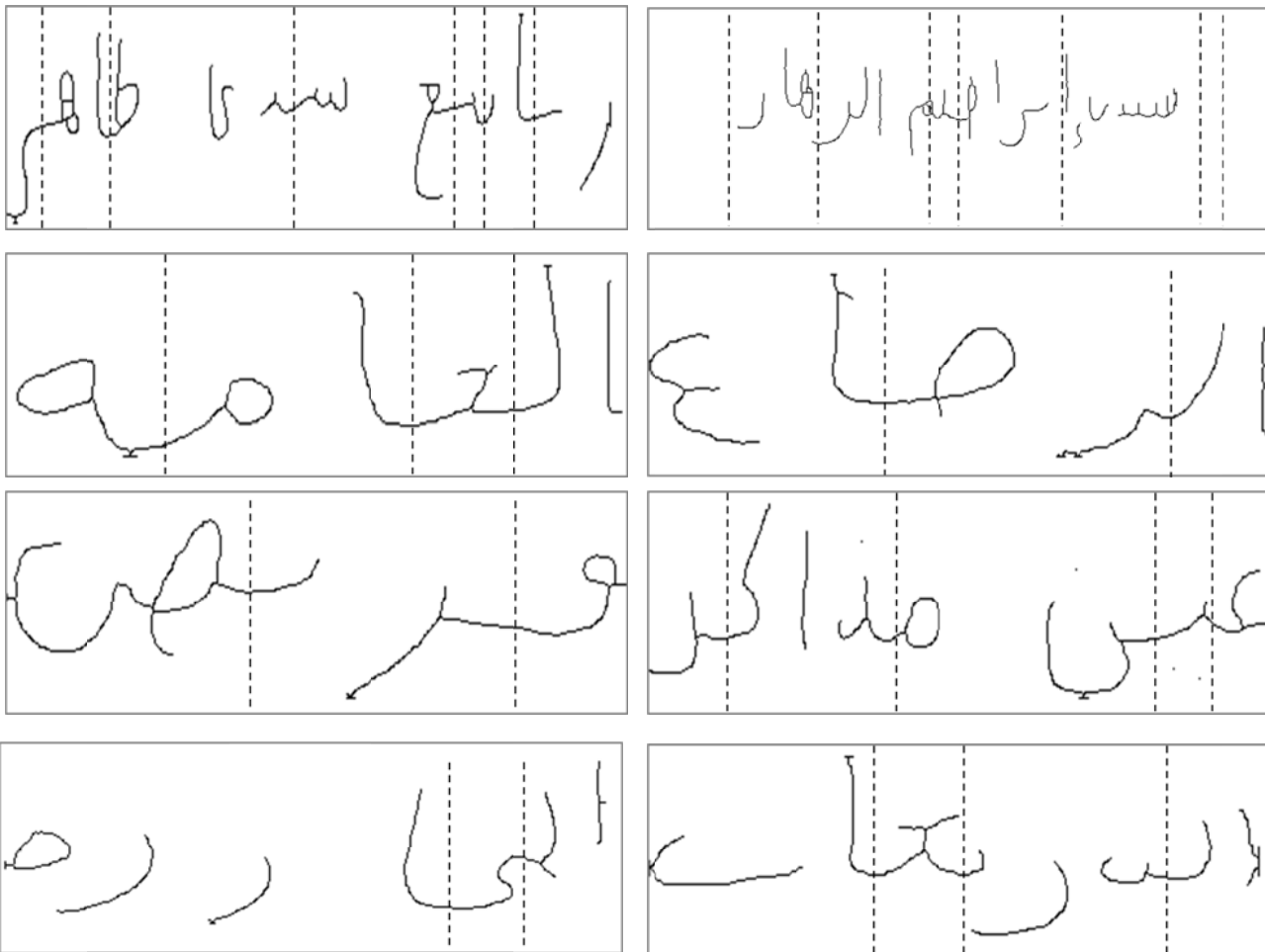
En général, il existe différents types d'erreurs qui peuvent se présenter pendant le processus de segmentation des mots arabes en caractères. Ces erreurs peuvent être récapitulées comme suit :

- **Sur-segmentation (over-segmentation)** : lorsque le nombre de segments est supérieur au nombre réel.
- **Sous segmentation (Under-segmentation)** : lorsque le nombre de segments est moins que le nombre réel.
- **Segmentation erronée (Mis-placed)** : lorsque le nombre de segments est juste, mais les limites sont fausses.

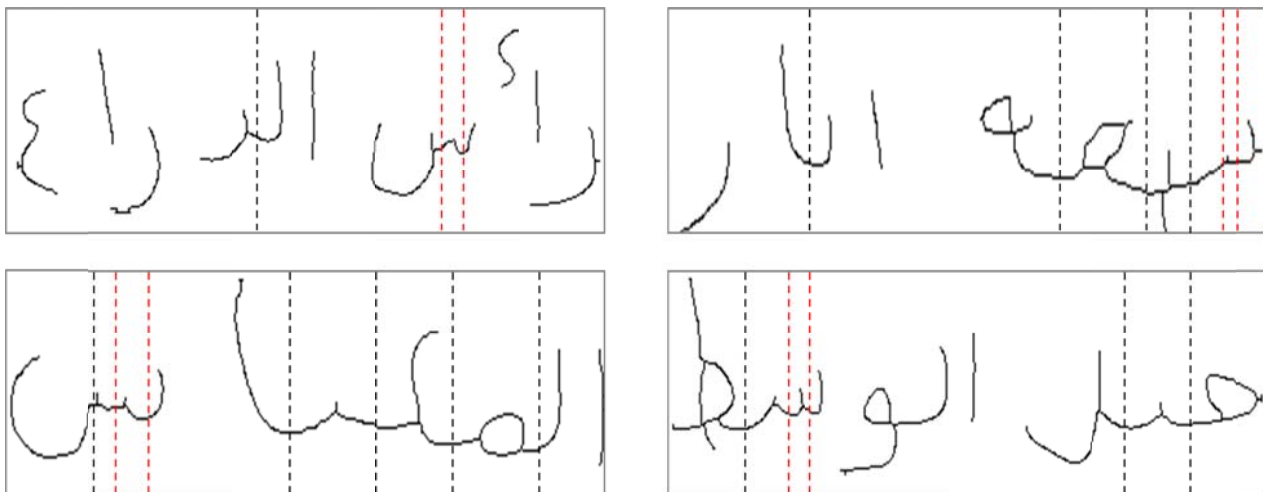
Nous avons testé notre algorithme de segmentation sur une base de test composée de 500 images de mots arabes manuscrits et les résultats sont résumés dans le tableau ci-dessous :

Points de segmentation	Segmentation correcte	Erreur de Segmentation		
		Sur-segmentation	Sous-segmentation	Segmentation erronée
4580	3975	105	50	450
100%	86.79%	2.29%	1.09%	09.83%
(Nu=0.04, $\gamma=0.02$) (t) = 0.65		605		
		13.21%		

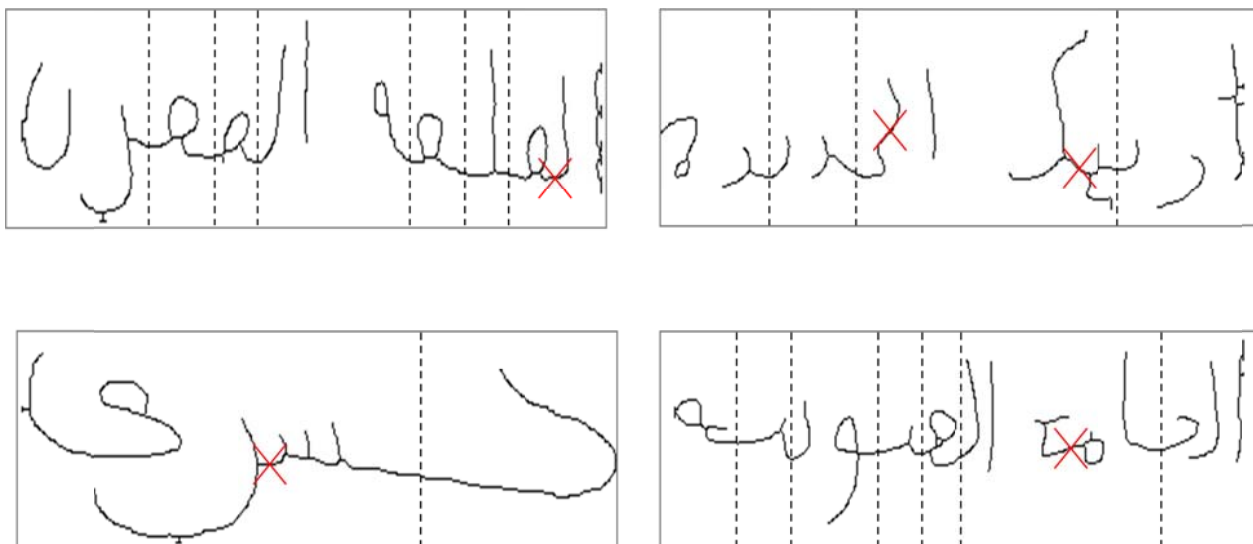
Tableau VII.1 : Résultats de la segmentation.



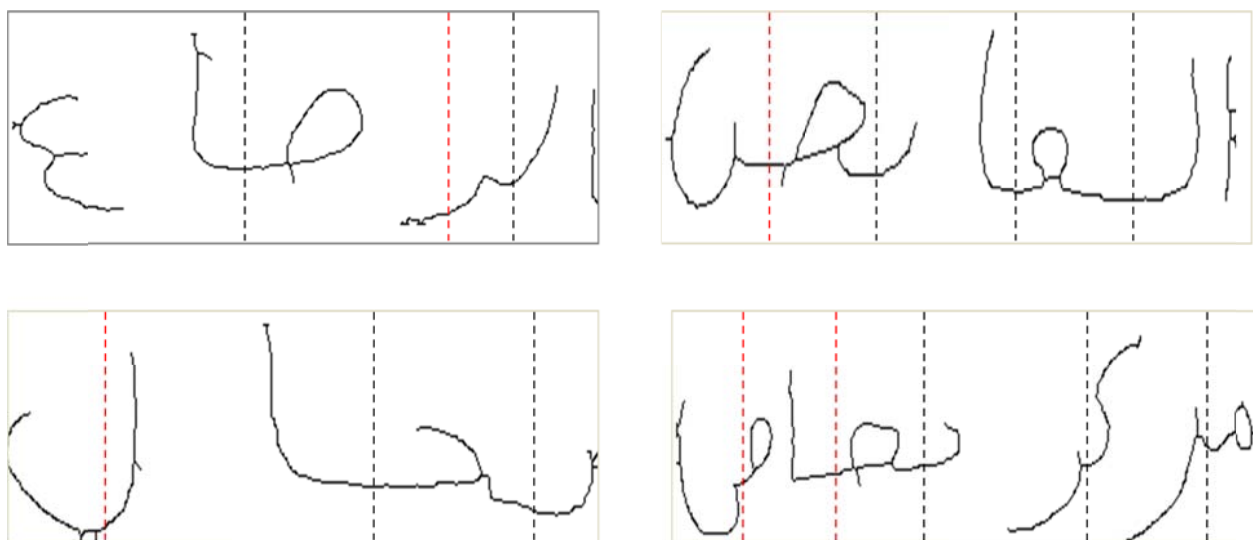
a- Exemples de la bonne segmentation.



b- Exemples de la sur segmentation.



c- Exemples de sous segmentation.



D- Exemples de la segmentation erronée.

Figure VII.10 : Exemples de l'algorithme de segmentation proposé.

La Figure VII.10 montre des exemples de l'application de l'algorithme de segmentation proposé sur des images de la base de données IFN/ENIT.

VII.4 Conclusion

Dans ce chapitre, nous avons présenté un nouvel algorithme de segmentation de mots arabes manuscrits basé sur une approche de validation SVM mono classe. Cet algorithme de segmentation inclut un Arabic Morphological Segmenater (AMS), qui est un ensemble de règles morphologiques dont l'objectif est de fournir au module de validation les meilleurs points de segmentation primaires (PSP) et ce dernier se chargera d'en déduire les points de segmentation réels à l'aide d'une phase d'apprentissage, basée sur une méthode d'extraction de primitives par les histogrammes des gradients orientées (HOG). Nos résultats préliminaires sont encourageants, montrant un taux de bonne segmentation supérieure à 86% et concurrentielle à l'état de l'art, et les erreurs de segmentation sont dues principalement aux positions irrégulières des caractères touchés à cause de la mauvaise qualité des images numérisés. Dans l'ensemble, les résultats sont satisfaisants. Néanmoins, il faut seulement signaler qu'on peut envisager des possibilités d'amélioration et extensions futures par :

- L'Arabic Morphological Segmenater (AMS), du système de segmentation devra être amélioré encore plus.
- L'amélioration de la procédure actuelle d'extraction de primitives ou l'utilisation d'autres primitives pour l'augmentation du taux de réussite de cet algorithme de segmentation.
- Le perfectionnement de la technique de détection de la ligne de base, car elle joue un rôle primordial dans le processus de segmentation.

CONCLUSION GENERALE

Conclusion générale

Le travail présenté dans cette thèse s'inscrit dans le cadre de la reconnaissance automatique hors ligne de l'écriture arabe manuscrite. Deux problématiques ont été abordées, la réalisation d'un système global de reconnaissance de mots arabes manuscrits d'une part, et d'autre part, on a proposé un nouvel algorithme de segmentation en caractères des mots arabes manuscrits. Malgré les efforts et les travaux de recherche réalisés dans le domaine de la reconnaissance de l'écriture arabe manuscrite, aucun système n'a réussi à atteindre la perfection idéale. Donc, c'est un domaine qui est toujours un défi scientifique ouvert aux propositions et aux expérimentations.

Dans ce cadre, nous avons proposé une solution globale au problème de la reconnaissance automatique des mots arabes manuscrits. Cette solution est fondée sur le développement d'une approche basée sur une combinaison parallèle des classifieurs. Le système proposé s'articule autour de quatre modules essentiels : le module de prétraitement qui a pour objectif de préparer l'image du mot aux traitements ultérieurs, un module d'extraction de primitives basé sur deux familles de descripteurs différentes : les coefficients de la DCT, et les histogrammes du gradient orientés (HOG), un module de classification basé sur deux classifieurs distincts : k plus proches voisins flou et les supports à vaste marges (SVM) et le dernier module, est la combinaison parallèle de classifieurs.

Notre travail était focalisé sur la reconnaissance de mots manuscrits arabe représentant les noms des villes et villages tunisiens et qui peut être étendu ultérieurement à la lecture des adresses postales ou la vérification des chèques. Les tests ont été effectués sur la base de données de référence bien connue IFN/ENIT. Ils ont montré la supériorité des systèmes combinant des classifieurs par rapport à des systèmes basés sur un seul classifieur dans le cadre de la reconnaissance de mots arabes manuscrits. Nous avons démontré aussi que la procédure d'extraction de primitives proposée à base des histogrammes de gradients orientés (HOG) s'adapte bien avec les variabilités de l'écriture arabe manuscrite et donne de bon résultats de reconnaissance.

Malgré que les approches globales présentent des excellentes performances dans le domaine de la reconnaissance de l'écriture arabe manuscrite, elles ne fournissent pas une solution définitive au problème de la lecture des textes arabes non-contraints. Ce qui nous a motivés à proposer dans la deuxième partie de ce travail, un nouvel algorithme de segmentation de mots en caractères. Car la segmentation en caractères représente l'obstacle principal à la résolution du problème.

L'algorithme proposé est fondé sur une coopération entre des règles morphologiques spécifiques à l'écriture arabe appelées l'Arabic Morphological Segmenater (AMS) et un module de validation basé sur un classifieur SVM mono-classe, dont la tâche principale consiste à accepter ou rejeter les points de segmentation primaires générées par le module (AMS). Des expériences ont été effectuées sur des mots arabes manuscrits extraits de la base de référence INF/ENIT, ont montré l'efficacité de l'approche proposée.

Les travaux présentés dans cette thèse s'avèrent intéressants et encourageants, mais ils n'ont pas la prétention d'être parfaits, car on peut toujours envisager des possibilités d'extensions futures telles que :

- Dans la poursuite de ce travail, nous avons l'intention d'employer un ensemble de caractéristiques plus discriminatif et nous augmentons le nombre de classifieurs à combiner.
- Nous planifions aussi, de mettre en œuvre un mécanisme de sélection caractéristique/classifieur pour trouver l'ensemble optimal de caractéristiques et de classifieurs pour ce problème.
- l'amélioration du module de segmentation en caractères pour permettre de l'intégrer dans un système complet de reconnaissance de l'écriture arabe manuscrite non contrainte.

BIBLIOGRAPHIE

Bibliographie

- 1- A. Belaïd, Y. Belaïd. “Reconnaissance des Formes : Méthodes et Application”. Inter Editions, Paris, 1992.
- 2- M. Kunt, G. Coray, G. Granlund, J.P. Haton, R. Ingold et M. Kocher. “Reconnaissance Des formes Et analyse de Scènes”. Volume 3, Presses Poly- techniques et Universitaires Romandes, 2000.
- 3- Abdelhak Boukharouba. “Contribution à la segmentation et à la reconnaissance de l’écriture arabe manuscrite”. Thèse de doctorat en sciences, Université de Mentouri Constantine, 2011.
- 4- Laurent Hartert. “Reconnaissance des formes dans un environnement dynamique appliquée au diagnostic et au suivi des systèmes évolutifs”. Thèse de doctorat, Université De Reims Champagne-Ardenne, 2010.
- 5- R. P. W. Duin, A.K. Jain and J. Mao. “Statistical Pattern Recognition: A Review”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1):4–37, 2000.
- 6- Laurent Miclet. “Méthodes structurelles pour la reconnaissance des formes”. Eyrolles, 1984.
- 7- C.Y. Suen. “N-gram statistics for natural language understanding and text processing”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2:164–172, 1979.
- 8- M. Cheriet. “Visual recognition of Arabic handwriting: challenges and new directions. Arabic and Chinese Handwriting Recognition”. Lecture Notes in Computer Science, vol. 4768, pp. 1-21, 2008.
- 9- M. Gilloux and M. Leroux. “Recognition of Cursive Amounts on Postal Cheques”. In first European Conference dedicated to Postal Technologies, pp. 705-712, June 1993.
- 10- B. Farou, S. Hallaci, et H. Seridi. “Système Neuro-Markovien pour la reconnaissance de l’écriture manuscrite arabe à vocabulaire limité”. Proc. 2nd Conférence Internationale sur l’informatique et ses Applications (CIIA'09), Saida, Algeria, May 3-4, 2009.
- 11- C. Olivier, H. Miled, K. Romeo, and Y. Lecourtier. “Segmentation and coding of Arabic Handwritten words”. ICPR'96, Vol 3, pp 264-268, Vienna, Austria, Aug 1996.
- 12- K.M. Sayre. “Machine recognition of handwritten words: a project report”. Pattern Recognition, vol.5, pages 213-228, 1973.
- 13- S. Madhvanath, and V. Govindaraju. “The Role of Holistic Paradigms in Handwritten Word Recognition”. IEEE Transactions on pattern analysis and machine intelligence, vol.23, no.2, February 2001.
- 14- C. Fang. “Deciphering Algorithms for Degraded Document Recognition”. Ph.D thesis. 1997.

- 15- N. Otsu, "A threshold selection method from grey scale histogram".IEEE Trans. on Syst. Man and Cyber. vol 1, pp 62-66, 1979.
- 16- J. A.Canny. "A Computational Approach to Edge Detection". IEEE Transactions on Pattern Analysis and Machine Intelligence, 8:679-714 (1986).
- 17- C.J. Hilditch. "Linear skeleton from square cupboards". Machine Intelligence, vol. 4, pp. 404-420, 1969.
- 18- T. Y.Zhangand Y. Suen. "A Fast Parallel Algorithms for Thinning Digital Patterns". Communication of the ACM, Vol 27, No. 3,Maret 1984, pp.236-239.
- 19- C. Singh, N. Bhatia, and A. Kaur. "Hough transform based fast skew detection and accurate skew correction methods". Pattern Recognition, Volume 41, Issue 12, December 2008, Pages 3528–3546.
- 20- M.M. Haji. "Farsi Handwritten word recognition using continuous hidden Markov models and structural features". Master Thesis, Shiraz University, Iran, 2005.
- 21- S.N Srihari, E. Cohen, J.J.Hull and L. Kuan. "A system to locate and recognize ZIP codes in handwritten addresses". IJRE pp 37–45, Vol 1, 1989.
- 22- L.S. Oliveira, F. Bortolozzi, C.Y. Suen. "Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy". IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol. 24, No. 11, pp. 1448-1456. 2001.
- 23- H.A. Beghdadi et M. Senouci. "Réseaux de neurones théorie et pratique". Office Des Publication Universitaire, 2000.
- 24- H. K. Ekenel, R. Stiefelhagen. "Analysis ofLocal Appearance-based Face Recognition: Effects of Feature Selection and Feature Normalization". Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06).
- 25- N. Dalal, and B. Triggs. "Histograms of Oriented Gradients for Human Detection". IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893, 2005.
- 26- A. Brakensiek ET G. Rigoll. "Handwritten address recognition using Hidden Markov Models". In A. Dengel, et al. Adaptive READ Research Project, LNCS 2956, pp. 103–122, Springer Berlin Heidelberg, 2004.
- 27- W. Ding, C. Suen, ET A. Krzyzak. "A new courtesy amount recognition module of a check reading system". In Proceedings of the 19th International Conference on Pattern Recognition-ICPR 2008, p.1–4, 2008.
- 28- Leila Chergui. "Combinaison de classifieur pour la reconnaissance de mots arabes manuscrits". Thèse de Doctorat, Université Mentouri Constantine, 2012.

- 29- N. Kharm, M. Ahmed, and R. Ward. "A new comprehensive database of handwritten Arabic words, numbers, and signatures used for OCR testing". Canadian Conference on Electrical and Computer Engineering, vol. 2, pp. 766–768, 1999.
- 30- Y. Al-Ohali, M. Cheriet, and Ch. Y. Suen. "Databases for recognition of handwritten Arabic cheques". Pattern Recognition, 36(1):111 – 121, 2003.
- 31- M. Pechwitz, S. Snoussi Maddouri, V. Märgner, N. Ellouze, H. Amiri. "IFN/ENIT Database of Handwritten Arabic Words". In CIFED'02, 2002.
- 32- . N. Srihari, H. Srinivasan, P. Babu, and C. Bhole. "Handwritten Arabic word spotting using the CEDARABIC document analysis system". In Proc. Symposium on Document Image Understanding Technology, SDIUT'05, pages 123–132, College Park, MD, Nov. 2005.
- 33- S.A. Mahmoud, I. Ahmad, M. Alshayeb, W.G Al-Khatib, M.T. Parvez, V. Margner, G.A. Fink. "KHATT: An open Arabic offline handwritten text data".dx.doi.org/10.1016/j.patcog.2013.08.009, 2013.
- 34- A. Mezghani, S. Kanoun, M. Khemakhem, H. ElAbed. "A Database for Arabic Handwritten Text Image Recognition and Writer Identification". International Conference on Frontiers in Handwriting Recognition (ICFHR), pp.399, 402, 2012.
- 35- A. Amin. "Off-Line Arabic Character Recognition: The State Of The Art". Pattern Recognition, Vol. 31, No. 5, pp. 517-530, 1998.
- 36- M. S. Khorsheed. "Off-Line Arabic Character Recognition – A Review". Pattern Analysis & Applications 5, pp.31 – 45, 2002.
- 37- N. E. Ben Amara, F. Bouslama. "Classification of Arabic script using multiple sources of information: State of the art and perspectives".. IJDAR'2003, 5: 195 – 212, 2003
- 38- L. M. Lorigo and V. Govindaraju. "Offline Arabic Handwriting Recognition: A survey". IEEE Trans. on Pattern Analysis and Machine Intelligence, 28(5), pp. 712 – 724, 2006.
- 39- V. Märgner, M. Pechwitz, H. El Abed. "ICDAR 2005 Arabic Handwriting Recognition Competition". In 8th International Conference on Document Analysis and Recognition, ICDAR'05, pp.70 – 74, 2005.
- 40- M.T. Parvez."Offline Arabic Handwritten Text Recognition: A Survey". ACM Computing Surveys, Vol.45, No.2, Article23, February2013.
- 41- S. Mahmoud, I. AbuHaiba, and R. Green."Skeletonization of Arabic characters using clustering based skeletonization algorithm (CBSA)".Pattern Recognition., vol. 24, no. 5, pp. 453-464, 1991.

- 42- M.M. Altuwaijri and M.A. Bayoumi. "A Thinning Algorithm for Arabic Characters Using ART2 Neural Network". IEEE transactions on circuits and systems-II: analog and digital, vol.45, no.2, February 1998.
- 43- M. A. Ali. "An efficient thinning algorithm for Arabic ocr systems". Signal & Image Processing: An International Journal (SIPIJ) Vol.3, No.3, June 2012.
- 44- L. Lorigo, V. Govindaraju. "Segmentation and Pre-Recognition of Arabic Handwriting". Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition, ICDAR'05. IEEE Computer Society, 2005.
- 45- F. Bouchareb, M. Bedda and S. Ouchetati. "New Preprocessing Methods for Hand-Written Arabic Word". Asian Journal of Information Technology, 5: 609-613, 2006.
- 46- K. Bouriel, F. Bouafif Samoud, H. El Abed, and S. SnoussiMaddouri. "Stratégies d'évaluation et de comparaison de méthodes d'extraction de la ligne de base de mots manuscrits arabes". Colloque International Francophone sur l'Ecrit et le Document, Octobre 2008, France.
- 47- S. Snoussi-Maddouri, F. Bouafif-Samoud, K. Bouriel, N. Ellouze, and H. Al Abed. "Baseline extraction: comparison of six methods on IFN/ENIT database". Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition (ICFHR), Montréal-Canada, 2008.
- 48- F. Farooq, V. Govindaraju, M. Perrone. "Pre-processing methods for handwritten Arabic documents". In: ICDAR05, pp. 267–271(2005).
- 49- A. Vinciarelli et J. Luetin. "A new normalization technique for cursive handwritten words". Pattern Recognition Letters, 22(9): 1043–1050, July 2001.
- 50- P. Burrow. "Arabic handwriting recognition". Master Thesis, School of Informatics, University of Edinburgh, England, 2004.
- 51- N. Azizi, T. Sari, L. Souici-Meslati, et M. Sellami. "Une architecture de combinaison floue de classifieurs neuronaux pour la reconnaissance de mots arabes manuscrits". Proceedings du 3^{ème} Colloque International Francophone sur l'Ecrit et le Document (CIFED), Hammamet-Tunisie, pp. 89-96, 2002.
- 52- D. Motawa, A. Amine, and R. Sabourin. "Segmentation of Arabic cursive script". In Proceedings of the International Conference on Document Analysis and Recognition.625–628, 1997.
- 53- S. Wshah, Z. Shi, and V. Govindaraju. "Segmentation of Arabic handwriting based on both contour and Skeleton segmentation". In proceedings of the 10th international conference on document analysis and Recognition (ICDAR).793–797, 2009.

- 54- A. Hamid and R. Haraty. "A Neuro-Heuristic Approach for Segmenting Handwritten Arabic Text". In: ACS/IEEE International Conference on Computer Systems and Applications, p.p. 0110, 2001.
- 55- A.M. Elgammal, M.A. Ismail. "A graph-based segmentation and feature extraction framework for Arabic text recognition". 6th International Conference on Document Analysis and Recognition, pp.622–626(2001).
- 56- B. Al-Badr, R.M. Haralick. "Segmentation-free word recognition with application to Arabic". In Proceedings of the Third International Conference on Document Analysis and Recognition, vol.1, 355–359(1995).
- 57- Romeo-Parker, K. R. K. Miled, H., and Y. Courtier. "A new approach for Latin/Arabic character segmentation". In proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR). 874–877, 1995.
- 58- M. Syiam, T. M. Nazmy, A. E. Fahmy, H. Fathi, K. Ali. "Histogram clustering and hybrid classifier for handwritten Arabic characters recognition". Proc. Of the 24th IASTED International Multi-Conference.
- 59- Yusra Osman. "Segmentation algorithm for Arabic handwritten text based on contour analysis". International Conference on Computing, Electrical and Electronics Engineering (ICCEEE), 2013.
- 60- H. Miled, c. Olivier, m. Cheriet, k. Romeo-pakker. "Une méthode rapide de reconnaissance de l'écriture arabe manuscrite". In seizième Colloque Gretsi, pp. 857-860, 1997.
- 61- T. Sari, L. Souici, and M. Sellami. "Off-line handwritten Arabic character segmentation algorithm: ACSA". In IWFHR'02: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition, page 452. IEEE Computer Society, 2002.
- 62- B. Timsari, H. Fahimi. "Morphological approach to character recognition in machine-printed Persian words". In: Proceeding of SPIE. Document Recognition III. San Jose (1996).
- 63- M. Elzobi, A. Al-Hamadi, Z. Al Aghbari, L., Dings, and A.Saeed. "Gabor Wavelet Recognition Approach for Off-Line Handwritten Arabic Using Explicit Segmentation". Advances in Intelligent Systems and Computing 233, 2014.
- 64- A.A. Atici, F.T. Yarman-Vural. "A heuristic algorithm for optical character recognition of Arabic script". Signal Processing 62 (1), 87–99(1997).
- 65- M. Elzobi, A. Al-Hamadi, Z. Al Aghbari, L., Dings. "IESK-ARDB: a database for handwritten Arabic and an optimized topological segmentation approach". International Journal on Document Analysis and Recognition (IJDAR), 1–14(2012).
- 66- X. Ding, And H. liu. "Segmentation Driven offline handwritten Chinese and Arabic script recognition". In Arabic and Chinese Handwriting Recognition. Lecture Notes in Computer Science, vol.4768. Springer, 196–217, 2008.

- 67- H. A. Al Hamad, and R. Abu Zitar. "Development of an efficient neural-based segmentation technique for Arabic handwriting recognition". *Pattern. Recognition.* 43, 2773–2798(2010).
- 68- R.M. Abdeen, A. Afifi, AB. El-Sisi. "An Enhanced Technique for Offline Arabic Handwritten Words Segmentation". 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015.
- 69- H. Miled, M. Cheriet, and C. Olivier. "Markovian modeling of Arabic cursive handwriting: an analytical approach". *Proceedings of Vision Interface*, pp. 255-262, 1998.
- 70- M. S. Khorsheed, and W. F. Clocksin. "Structural Features of Cursive Arabic Script". *Proceedings of the British Machine Conference*, pages 42.1-42.10. BMVA Press, September 1999.
- 71- A. Benouareth. "Reconnaissance de l'écriture arabe manuscrite par une approche hybride". Mémoire de magister, Labo. LRI, Département d'informatique, Université d'Annaba, Algérie, Février 2000.
- 72- N. Azizi, T. Sari, L. Souici-Meslati, M. Sellami. "Une architecture de combinaison floue de classifieurs neuronaux pour la reconnaissance de mots arabes manuscrits". CIFED'02, 7^{ème} Colloque International Francophone sur l'Écrit et le Document, pp. 89-96, Hammamet, Tunisie, Octobre 2002.
- 73- M. S. Khorsheed. "Recognizing handwritten Arabic manuscripts using a single hidden Markov model". *Pattern Recognition Letters*, Vol. 24, N°. 14, pp. 2235-2242, October 2003.
- 74- S. Al-Ma'adeed, D. Elliman, C. Higgins: "Off-line recognition of handwritten Arabic words using multiple hidden Markov models", *Knowledge-Based Systems*, Vol. 17, N°. 2-4, pp. 75-79, May 2004.
- 75- . Farah, L. Souici, and M. Sellami. "Arabic word recognition by classifiers and context". *Journal of Computer Science and Technology*, vol. 20, no. 3, pp. 402-410, 2005.
- 76- I. Gasmi, H. Merouani, et L. Souici-Meslati. "Combinaison de classifieurs". *Proceeding of the 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunication (SETIT)*, Sousse-Tunisia, 2005.
- 77- N. Farah, L. Souici, and M. Sellami. "Classifiers combination and syntax analysis for Arabic literal amount recognition". *Engineering Applications of Artificial Intelligence*, Elsevier, vol. 19, pp. 29-39, 2006.
- 78- H. El Abed, and V. Märgner. "Comparison of different preprocessing and feature extraction methods for offline recognition of hand-written Arabic words". *Proc. Ninth International Conference on Document. Parana*, 2007.
- 79- F. Nouar, M.E. Aissaoui, et H. Seridi. "Approche globale pour la reconnaissance de mots arabes manuscrits par combinaison parallèle de classifieurs". *Proceedings des Journées des Jeunes Chercheurs en Informatique (JCI)*, Guelma-Algeria, 2008.

- 80- A. Benouareth, A. Ennaji, and M. Sellami. "Semi-continuous HMMs with explicit state duration for unconstrained Arabic word modeling and recognition". *Pattern Recognition Letters*, vol. 29, no. 12, pp. 1742-1752, 2008.
- 81- A. Broumandnia, J. Shanbehzadeh, and M. Varnoosfaderani. "Persian/Arabic handwritten word recognition using M-band packet wavelets transform". *Image and Vision Computing*, Elsevier, vol. 26, pp. 829-842, 2008.
- 82- I. Zavorin, E. Borovikov, E. Davis, A. Borovikov, and K. Summers. "Combining different classification approaches to improve off-line Arabic handwritten word recognition". In *proceedings of SPIE -The International Society for Optical Engineering*, Volume 6815; 2008.
- 83- J.H. AlKhateeb, J. Jiang, J. Ren, F. Khelifi, and S.S. Ipson. "Multiclass. Classification of Unconstrained Handwritten Arabic Words Using Machine Learning Approaches". *The Open Signal Processing Journal*, vol. 2, pp. 21-28, 2009.
- 84- R. Al-Hajj, L. Likforman-Sulem, and C. Mokbel. "Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition". *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (7), pp. 1165–1177, 2009.
- 85- Y. Kessentini, T. Paquet, and A. Ben Hamadou. "Off-line handwritten word recognition using multi-stream hidden Markov models". *Pattern Recognition Letters* 31(1): 60-70 (2010).
- 86- J. Chen, H. Cao, R. Prasad, A. Bhardwaj, P. Natarajan, "Gabor features for offline Arabic". *DAS '10 Proceedings of the 9th IAPR International Workshop on Document Analysis Systems .handwriting recognition*, Pages 53-58, 2010.
- 87- H. Nemmour, Y. Chibani. "Handwritten Arabic word recognition based on Ridgelet transform and support vector machines". In *Proceedings of International Conference on high Performance Computing and Simulation (HPCS)*, Pages 357-361, 2011.
- 88- A. Alalshekmubarak, A. Hussain, and Q.F. Wan. "Off-Line Handwritten Arabic Word Recognition Using SVMs with Normalized Poly Kernel". *Neural Information Processing, Lecture Notes in Computer Science*, vol. 7664, pp. 85-91, 2012.
- 89- S. Nemouchi, L. SouiciMeslati, N. Farah. "Classifiers Combination for Arabic Words Recognition: Application to Handwritten Algerian City Names". *Image and Signal Processing, Lecture Notes in Computer Science*, vol. 7340, pp 562-570, 2012.
- 90- L. Chergui, M. Kef, and S. Chikhi S. "Combining Neural Networks for Arabic Handwritten Recognition". *The international Arab Journal of Information Technology*, Vol. 9, no. 6, 2012.
- 91- S. Abdel Azeem, and H. Ahmed. "Effective technique for the recognition of offline Arabic handwritten words using hidden Markov models" *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 16, no.4, pp. 399-412, 2013.

- 92- B. El qacimy, M. Ait kerroum, A. Hammouch. "Word-Based Arabic Handwritten Recognition Using SVM Classifier with a Reject Option". In proceeding of 15th International Conference on Intelligent Systems Design and Applications (ISDA), 2015.
- 93- M. Pechwitz and V. Maergner. "HMM based approach for handwritten Arabic word recognition using the IFN/ENIT- database". In ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition, page 890. IEEE Computer Society, 2003.
- 94- R. El-Hajj, L. Likforman-Sulem, C. Mokbel. "Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling". In 8th International Conference on Document Analysis and Recognition, ICDAR'05, pp.893 – 897, 2005.
- 95- R. El-Hajj and C. Mokbel. "HMM-based Arabic handwritten cursive recognition system". In Research Trends in Science and technology RTST 05, Beyrou, March 2005.
- 96- D. Xiang, H. Yan, X. Chen, Y. Cheng. "Offline Arabic handwriting recognition system based on HMM". In proceedings of 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), 2010.
- 97- S. Al-Ma'adeed. "Recognition of off-line handwritten Arabic words". Thesis of Doctorate, university of Nottingham, England, 2004.
- 98- K. Bouriel, S. Maddouri, K. Hamrouni. "Un système neuronal pour la reconnaissance de mots arabes manuscrits". In Proceeding of the 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunication (SETIT), Sousse-Tunisia, 2005.
- 99- M. Elleuch, N. Tagougui, M. Kherallah. "Arabic handwritten characters recognition using Deep Belief Neural Networks". 12th International Multi-Conference on Systems, Signals & Devices (SSD), 2015.
- 100- A. Graves, M. Liwicki, S. Fernández, R Bertolami, H. Bunke, J Schmidhuber. "A novel connectionist system for unconstrained handwriting recognition". IEEE transactions on pattern analysis and machine intelligence 31 (5), 855-868, 2009.
- 101- V. Märgner, H. El Abed. "ICDAR 2007: Arabic Handwriting Recognition Competition". In 9th International Conference on Document Analysis and Recognition, ICDAR'07, 2007.
- 102- A. Abdulkadr. "Two-tier approach for Arabic offline handwriting recognition". In 10th Inter. Workshop on Frontiers in Handwriting Recognition (IWFHR), pages 161–166, 2006.
- 103- H. Al Abed and V. Märgner. "ICDAR 2009-Arabic handwriting recognition competition", International Journal on Document Analysis and Recognition (IJ DAR), Springer, vol. 14, pp. 3-13, 2011.
- 104- H. Al Abed and V. Märgner. "ICFHR 2010—Arabic handwriting recognition competition". In: Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 709–71, November 2010.

- 105- A. Giménez, I. Khoury, A. Juan. “Windowed Bernoulli mixture HMMs for Arabic handwritten word recognition”. In: Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition (ICFHR), Kolkata, India, and November 2010.
- 106- M. Hamdani, H. ElAbed, T.M. Hamdani, V. Märgner, A.M. Alimi. “Improving a HMM-Based off-line handwriting recognition system using MME-PSO optimization”. In: Proceedings of the Document Recognition and Retrieval XVIII (2011).
- 107- V. Märgner, H. El Abed. “ICDAR 2011—Arabic handwriting recognition competition”. In: Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR) (2011).
- 108- T. Mitchell. “Machine learning”. Edition MIT press, USA, 1997.
- 109- J.M. Keller, M.R. Gray, and J.A. Givens. “A Fuzzy K Nearest Neighbor algorithm”. IEEE transaction on systems, man, and cybernetics, vol.SMC-15, pp. 580-585, 1985.
- 110- V. N. Vapnik. “Statistical learning theory”. Wiley, New York, 1998.
- 111- J. Mercer. “Functions of positive and negative type and their connection with the theory of integral equations”. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 209:415–446, 1909.
- 112- O. Kadri, L. H. Mouss, M. D. Mouss, A. Abdelhadi. “Reconnaissance des Formes par SVM pour le Diagnostic du Système de Pasteurisation d’une Usine de Lait”. Revue des Sciences et de la Technologie –RST, Volume 4 N°1 /janvier 2013.
- 113- S. Knerr, L. Personnaz, J. Dreyfus. “Single-layer learning revisited: A stepwise procedure for building and training a neural network”. In: Soulié F.F., Héroult J. (eds) Neurocomputing. NATO ASI Series (Series F: Computer and Systems Sciences), vol 68. Springer, Berlin, Heidelberg.
- 114- J. Platt, N. Cristianini, and J. Shawe-Taylor. “Large margin dags for multiclass classification”. In Advances in Neural Information Processing Systems 12, pages 547–553. MIT Press, 2000.
- 115- D.H. Wolpert ET W.G. Macready. “No free lunch theorems for optimization”. IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pages 67–82, 1997.
- 116- Simon Bernard. “Forêts Aléatoires : De l’Analyse des Mécanismes de Fonctionnement à la Construction Dynamique”. Thèse de Doctorat, Université de Rouen, France, 2009.
- 117- Héli Khoufi Zouari. “Contribution à l’évaluation des méthodes de combinaison parallèle de classifieurs par simulation”. Thèse de Doctorat, Université de Rouen, France, 2004.
- 118- S. Günter. “Multiple classifier systems in offline cursive handwriting recognition”. Thesis of Doctorate, University of Bern, Suisse, 2004.
- 119- P. Ahmed and Y. Al-Ohali. “Arabic character recognition: progress and challenges”. Journal of King Saudi University, vol. 12, pp. 85-116, 2000.
- 120- S.M. Touj, N. Ben Amara, and H. Amiri. “A hybrid approach for off-line Arabichandwriting recognition based on planar hidden Markov modeling”. Proceedings of the 9thInternational

- Conference on Document Analysis and Recognition (ICDAR), Curitiba-Brazil, vol. 2, pp. 964-968, 2007.
- 121- Salima Nebti. "Reconnaissance de Caractères Manuscrits par Intelligence Collective". Thèse de Doctorat, Université Ferhat Abass-Sétif, 2013.
- 122- L.I. Kuncheva. "Combining Pattern Classifiers: Methods and Algorithms". John Wiley & Sons, New Jersey-United States of America (USA), 2004.
- 123- M. Van Erp, and L. Schomaker. "Variants of the Borda count method for combining ranked classifier hypotheses". Proceedings 7th International Workshop on frontiers in handwriting recognition (7th IWFHR): September 11-13, 2000, Amsterdam.
- 124- J. Parker. "Rank and response combination from confusion matrix data". Information Fusion, 2:113-120, 2001.
- 125- H. Zouari, L. Heutte, Y. Lecourtier, A. Alimi. "Un panorama des méthodes de combinaison de classificateurs en reconnaissance de formes". RFIA'2002, Angers, France, vol. 2, pp. 499-508, 2002.
- 126- L. Xu, A. Krzyzak, and C. Suen. "Methods of combining multiple classifiers and their applications to handwriting recognition". IEEE Transactions on Systems, Man, and cybernetics, 22(3):418-435, 1992.
- 127- J. Kittler, M. Hatef, R. Duin, and J. Matas. "On combining classifiers". IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3):226-239, 1998.
- 128- I. Bloch. "Information combination operators for data fusion: a comparative review with classification". IEEE Transactions on Systems Man Cybernetics - Part A: Systems Humans, 26:52-67, 1996.
- 129- R. Zaghdoudi, H. Seridi. "Combination of Multiple Classifiers for Off-Line Handwritten Arabic Word Recognition". The international Arab Journal of Information Technology, Volume 14, No. 5, September 2017.
- 130- S. Chen, C. Liu. "Precise Eye Detection Using Discriminating HOG Features". Computer Analysis of Images and Patterns, Lecture Notes in Computer Science vol. 6854, pp 443-450, 2011.
- 131- Chih-Chung Chang and Chih-Jen Lin, LIBSVM: "a library for support vector machines". ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011
- 132- M. Khalifa, and Y. Bing Ru. "A Novel Word Based Arabic Handwritten Recognition System Using SVM Classifier". Advanced Research on Electronic Commerce, Web Application, and Communication, Communications in Computer and Information Science, vol. 143, pp. 163-171, 2011.
- 133- R. Zaghdoudi, H. Seridi. "Nouvel Algorithme de segmentation de l'écriture arabe manuscrite basé sur une approche de Validation SVM". Le 1^{er} Séminaire National sur la Simulation Numérique dans les Sciences Appliquées, (SNSA'2016), Université de Guelma, 15 Décembre, 2016.