

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université Ahmed Draia - Adrar
Faculté des Sciences et de la Technologie
Département des Mathématiques et Informatique



Mémoire de fin d'étude, en vue de l'obtention du diplôme de Master en
informatique

Option : Système d'Information et Technologies Web

Thème

**Utilisation des caractéristiques statistiques et les
Moments de Zernike pour la Reconnaissance des Lettres
Arabes Manuscrits**

Préparés par

Tayebi Abderrazeq et Bomediana Touhami.

Soutenu le 06 juin 2018, devant le jury composé de:

Mr. KOHILI Mohammed
Mr. CHOUGUEUR Djilali
Mr. MAMOUNI El mamoun
Mr. OUAHAB Abdelwahab

Encadreur
Président
Examinateur
Examinateur

Année Universitaire 2017/2018

DEDICACE

A mes parents

A tout ma famille

A tout les Amies

TAYEBI ABDERRAZEQ

DEDICACE

Aux êtres qui me sont les plus chères au monde : mes parents.

A mes soeurs et mes frères

A toute ma famille

A tous mes amis (es)

A ceux qui me sont chères.

BOMEDIANA TOUHAMI

Remerciment

Avant tout, Nous remercions Dieu le très haut qui nous avons données le courage et la volonté de réaliser ce modeste travail. Nous remercions le seigneur tout puissant de nous avoir accordé volonté et patience dans l'accomplissement de ce travail à terme.

Nous tenons à exprimer nos vifs remerciements à

Mr. KOHILI Mohammed". "

Nous remercions « TOUS » les Messieurs et dames, les professeurs pour leurs précieux conseils.

Notre remerciements vont également aux membres du jury d'avoir accepté d'évaluer notre travail. Sans oublier de remercier nos amis.

Sans oublier bien sûr de remercier profondément tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Abderrazeq et Touhami

Résumé

Nous présentons dans ce mémoire une étude expérimentale qui a pour l'objectif de l'atteint de meilleur taux de reconnaissance des caractères arabes manuscrits en utilisant les caractéristique statistique et les moments de Zernike puis une combinaison entre les deux. Ce travail a nécessité la réalisation d'un système complet de reconnaissance des caractères arabes manuscrits hors-ligne, la génération d'une base de données de 4350 caractères pour des différentes positions (début, milieu, fin et isolé) et l'utilisation la méthode de classification SVM en raison de l'efficacité de cette méthode. Les résultats préliminaires obtenus sont très encourageants et prometteurs par rapport à la littérature et montrent que l'utilisation de l'hybridation peut être bénéfique à une meilleure reconnaissance des caractères arabe manuscrits par la méthode SVM.

Mots clés

Reconnaissance, caractères arabes manuscrits, caractéristique statistique, moments de Zernike Machines à vecteurs de support SVM.

Abstract

In this paper, we present an experimental study, which has for the objective of achieving better recognition rate of handwritten Arabic characters using statistical characteristics and moments of Zernike then a combination between the two. This work required the realization of a complete off-line handwritten Arabic character recognition system, the generation of a database of 4350 characters for different positions (beginning, middle, end and isolated) and the use of the SVM classification method due to the efficiency of this method. The preliminary results obtained are very encouraging and promising compared to the literature and show that the use of hybridization can be beneficial to better recognition of handwritten Arabic characters by the SVM method.

Keywords

Recognition, handwritten Arabic characters, statistical characteristics, and moments of Zernike, support vector machines SVM.

Sommaire

Dédicace I	I
Dédicace II	II
Remerciement	III
Résumé	IV
Sommaire	V
Liste Des Tableaux	IX
Liste Des Figures	X
Introduction Générale	01
Chapitre I : Reconnaissance de caractères arabe manuscrits	
I.1 Introduction	03
I.2 Caractéristiques de l'écriture Arabe	03
I.3 Différents aspects de reconnaissance de l'écriture	10
I.3.1 Reconnaissance en-ligne et hors-ligne	10
a) La reconnaissance En-Ligne (on-line)	10
b) La reconnaissance hors-ligne (off-line)	11
I.3.2 Reconnaissance globale ou analytique	12
a) Approche globale	12
b) Approche analytique	13
I.4 Organisation générale d'un système de reconnaissance	14
I.4.1 Phase d'acquisition	14
I.4.1.1 Les étapes d'acquisition	14
I.4.1.2 Notion de voisinage	14
I.4.2 Phase de prétraitement	15
I.4.2.1 Binarisation	15
I.4.2.2 Transformation par érosion	16
I.4.2.3 Transformation par dilatation	16
I.4.2.4 Ouverture morphologique	17
I.4.2.5 Fermeture morphologique	17
I.4.2.6 Squelettisation d'une image	17
I.4.2.7 Normalisation	18
I.4.3 Phase de segmentation	18
I.4.3.1 Les techniques de la segmentation	19

a-Segmentation implicite.....	19
b-Segmentation explicite	19
I.4.3.2 Etapes de segmentation.....	20
I.4.4 Phase d'extraction des caractéristiques.....	21
I.4.4.1 Caractéristiques structurelles.....	21
I.4.4.2 Les caractéristiques statistiques.....	22
I.4.4.3 Les transformations globales.....	22
I.4.4.4 Superposition des modèles (template matching) et corrélation.....	22
I.4.5 Phase de classification.....	22
I.4.5.1 L'apprentissage	22
I.4.5.2 Reconnaissance et décision.....	23
I.4.5.3 Décision Bayésienne.....	23
I.4.5.4 La méthode du plus proche voisin (KPPV).....	24
I.4.5.5 Les réseaux de neurones	24
I.4.6 Phase de post traitement	25
I.5 Conclusion.....	26

Chapitre II : Les techniques de reconnaissance de caractères arabes

II.1 Introduction	27
II.2 Méthodes d'extraction des caractéristiques	27
II.2.1 Zoning (densités).....	27
II.2.2 Les moments de Zernike.....	28
II.2.2.1 Formule des moments de Zernike.....	29
II.3 Méthodes de classification.....	31
II.3.1 Machines à vecteurs supports (SVM).....	31
II.3.2 Historique	31
II.3.3 Application des SVM à la classification binaire.....	32
II.3.3.1 Définition.....	32
II.3.3.2 Notion d'apprentissage.....	32
II.3.3.3 Principe de fonctionnement générale.....	33
II.3.4 Linéarité et non linéarité.....	35
II.3.4.1 Cas linéairement séparable.....	35
II.3.4.2 Cas non linéairement séparable.....	35
II.3.5 Fondements mathématique.....	36

II.3.5.1 Principe.....	36
II.3.5.2 Forme primale.....	37
II.3.5.3 Forme duale.....	38
II.3.6 Principe du noyau	39
II.3.6.1 Exemple.....	40
II.3.6.2 Condition de Mercer.....	40
II.3.6.3 Exemples de noyaux.....	41
II.3.7 Marges souples.....	41
II.3.8 SVM multi-classes.....	43
II.3.8.1 Approche Une-contre-Tous	44
II.3.8.2 Approche Une-contre-Une.....	44
II.3.8.3 Phase de décision.....	44
II.3.8.3.1 votant.....	44
II.3.8.3.2 Arbre de décision.....	44
II.3.9 Avantages et limites.....	45
II.3.9.1 les avantages.....	45
II.3.9.2 Les limites.....	45
II.4 Conclusion.....	46
 Chapitre III : Implémentation et résultats	
III.1 Introduction.....	47
III.2 Ressources matérielles et logicielles.....	47
III.2.1 Ressources matérielles.....	47
III.2.2 Ressources logicielles.....	47
III.3 Description de notre système de reconnaissance.....	48
III.3.1 Création des séparateurs (on ligne).....	48
III.3.1.1 Description de base de données utilisées.....	49
III.3.1.2 Prétraitement et Segmentation.....	49
III.3.1.2.1 Algorithme de segmentation.....	49
III.3.1.2.2 les résultats de segmentation.....	50
III.3.1.2.3 Les problèmes de segmentation.....	50
III.3.1.2.4 Evaluation des résultats de segmentation.....	51
III.3.1.3 Image de caractère.....	51
III.3.1.4 Elimination de bruit et extraction d'objet concerne.....	51

III.3.1.5 Normalisation des images.....	52
III.3.1.6 L'extraction des primitives.....	52
III.3.1.7 Création de corpus.....	53
III.3.1.8 L'apprentissage.....	54
III.3.1.9 Création des séparateurs.....	54
III.3.2 Reconnaissance (hors ligne).....	54
III.3.2.1 Classification.....	55
III.4 Résultats et discussion.....	55
III.4.1 En cas de zernik.....	56
III.4.2 En cas de zonage.....	59
III.4.3 En cas d'hybridation.....	60
III.4 Conclusion.....	63
Conclusion générale.....	64
Références.....	66

Liste des tableaux

Chapitre I	
Tableau 1.1 Alphabet arabe dans ses différentes formes	04
Tableau 1.2. Différents hampes et jambages dans les lettres arabes.	05
Tableau 1.3 Exemple de mots composés de la droite vers la gauche de 1, 2, 3 PAWs	05
Tableau 1.4 Exemple de caractères avec et sans madda	06
Tableau 1.5 Caractères susceptibles d'être ligaturés verticalement	06
Tableau 1.6. Lettres arabes ayant des points diacritiques	07
Tableau 1.7. Lettres arabes ayant des corps identiques.	09
Chapitre III	
Tableau 3.1: Les résultats de segmentation.	50
Tableau 3.2 : Exemple des classes utilisé	56
Tableau 3.3 : Taux de reconnaissance zernike	57
Tableau 3.4 : variante taux pour chaque classe	57
Tableau 3.5 : résultats de moment de zernik pour taille vecteur (n=18, m =18)	58
Tableau 3.6 : résultats de zonage	59
Tableau 3.7 : résultats de la variante zonage	60
Tableau 3.8 :résultats de concaténation zernik et zonage	61
Tableau 3.9 : résultat concaténation zonage et zernik	62

Liste des figures

Chapitre I	
Figure 1.1 Exemple d'écriture arabe montrant la ligne de base	05
Figure 1.2. Exemple de pseudo-mots constituant des mots arabes.	05
Figure 1.3. L'article défini "ال" dans deux mots arabes.	06
Figure 1.4. Deux types de textes arabes avec et sans signes de voyelles.	08
Figure 1.5. Exemples de caractères arabes écrits suivant différentes fontes.	09
Figure 1.6 Différentes phrases arabes dans différents modèles [08]	09
Figure I.7 : Différents systèmes, représentations et approches de reconnaissance.	13
Figure 1.8 : Schéma général du système de reconnaissance des caractères.	14
Figure 1.9 Voisinage à 4 et Voisinage à 8	15
Figure 1.10 Effets de certaines opérations de prétraitement.	15
Figure 1.11 Exemple de Binarisation adaptative [22]	16
Figure 1.12 Image traitée par érosion	16
Figure 1.13 Image traitée par dilatation	16
Figure 1.14 Image traitée par Ouverture	17
Figure 1.15 Image traitée par Fermeture	17
Figure 1.16 La squelettisation d'une Image [28]	17
Figure 1.17 Exemple d'histogramme horizontal et vertical d'une ligne de texte [28].	20
Figure 1.18 Exemple de segmentation [2]	21
Chapitre II	
Figure 2.1 Exemple de reconstructions à partir des descripteurs de Zernike	30
Figure 2.2 Séparation de deux ensembles de points par un Hyperplan H	33
Figure 2.3 Hyperplan optimal, marge et vecteurs de support	34
Figure 2.4 Maximisation de la marge	34
Figure 2.5 linéarité et non linéarité	35
Figure 2.6 Transformation de l'espace de représentation et l'hyperplan séparateur dans le cas non linéairement séparables.	36
Figure 2.7 Transformation linéaire des données en une séparation linéaire dans un nouvel espace.	40
Figure 2.8 Hyperplan de séparation optimale avec marge souple dans un cas non linéairement séparable.	42

Figure 2.9 :Graphe acyclique direct pour la classification de quatre classes en utilisant des classificateurs SVM binaires.	45
Chapitre III	
Figure 3.1 Schéma général de notre système de reconnaissance.	48
Figure 3.2: Exemple de ligatures verticales connectées ou non.	50
Figure 3.3: Exemple de liaison indésirable entre caractères.	51
Figure 3.4: Exemple de coupure indésirable	51
Figure 3.5 : Exemple d'élimination des objets bruits et extraire des objets concernes.	51
Figure 3.6 : Exemple de normalisation et centralisation des images.	52
Figure 3.7: Exemple de corpus	53
Figure 3.8 : Schéma général de classification.	54
Figure 3.9 : Taux de reconnaissance Zernike	57
Figure 3.10 : Taux de reconnaissance de zonage	60

Introduction

générale

Introduction

La reconnaissance automatique de l'écriture consiste à faire reconnaître par une machine, des mots ou des séquences de mots tracés par les êtres humains, soit de manière différée soit de manière interactive.

La reconnaissance automatique des mots manuscrits prend de l'ampleur durant les quatre dernières décennies, le nombre de documents que manipule toute entreprise étant phénoménale, le recours aux systèmes automatiques devient indispensable.

La reconnaissance des textes est utilisée dans plusieurs domaines où le texte est la base de travail, principalement en bureautique, pour des buts d'indexation et d'archivage automatique de documents, en publication assistée par ordinateur (PAO) pour faciliter la composition à partir d'une sélection de plusieurs documents, dans la poste pour le tri automatique du courrier, dans une banque pour faciliter la lecture des montants de chèques...etc.

Des systèmes concernant le manuscrit latin montrent de plus en plus des taux de reconnaissance remarquables, ce qui manque à l'écriture arabe.

La langue arabe n'a pas eu cette chance, contrairement au latin, elle reste encore au niveau de la recherche et de l'expérimentation, c'est-à-dire que le problème reste encore un pari ouvert pour les chercheurs. L'écriture arabe étant par nature cursive, elle pose de nombreux problèmes aux systèmes de reconnaissance automatique.

La cursivité de l'écriture arabe montre une complexité de la morphologie des caractères. Ce problème engendre une forte inertie à différents niveaux notamment :

- Le choix de primitives pertinentes décrivant la variabilité de la morphologie des caractères, sachant que certaines caractéristiques topologiques sont sensibles à la dégradation, notamment les points diacritiques et les boucles.
- La nécessité d'une modélisation robuste et une méthode d'apprentissage efficace pour prendre en considération toutes les variations morphologiques de l'écriture arabe.

Les primitives sont généralement définies par expérience ou par intuition. Plusieurs caractéristiques peuvent être extraites. La représentation des caractéristiques utilisée est une représentation vectorielle. La taille du vecteur peut être large si un grand nombre de caractéristiques est extrait. Il a été observé que certaines caractéristiques extraites sont non pertinentes ou redondantes au système de reconnaissance. Donc, il est nécessaire de définir des caractéristiques discriminantes lors du développement d'un système de reconnaissance.

Il existe plusieurs méthodes pour extraire les caractéristiques des caractères manuscrit, l'objectif de notre travail consiste à développer un système de reconnaissance des caractères

manuscripts arabes en utilisant les meilleurs caractéristique qui donne meilleurs taux de reconnaissance. Pour cela on a utilisée deux types, les caractéristiques statiques et les moments de Zernike puis une combinaison entre les deux caractéristiques. Ce travail a nécessité la réalisation d'un système complet de reconnaissance des caractères arabes manuscrits hors-ligne, la génération d'une base de données de caractères dans leurs différentes positions (début, milieu, fin et isolé) et pour la phase d'apprentissage c'est la méthode de classification SVM en raison de l'efficacité de cette méthode.

Notre mémoire est structuré en trois (03) chapitres :

Le premier chapitre de ce mémoire, présente le concept général de reconnaissance des caractères arabes manuscrits, comme l'historique de l'écriture arabe et en mettant le point sur les caractéristiques morphologique de l'écriture Arabe puis les différents aspects de reconnaissance de l'écriture, Enfin les différentes phases de processus de reconnaissance.

Le deuxième chapitre représente les techniques de reconnaissance, tel que les méthodes de l'extraction des caractéristiques en représentant deux méthode, enfin une description détailler sur la méthode de classification SVM.

Nous présentons dans le chapitre trois (03) une description détaillée de notre système de reconnaissance de caractères Arabes manuscrits. Ainsi que nous avons présenté les différents résultats d'expérimentations par l'utilisation de différentes caractéristiques qui donnent le bon taux de reconnaissance. Finalement, nous présentons une discussion et évaluation des résultats obtenus.

Chapitre I

Reconnaissance de
caractères arabes
manuscrits

I.1 Introduction

La reconnaissance de l'écriture manuscrite par ordinateur est un domaine très vaste, les travaux de recherches sur l'écriture arabe sont moins nombreux en comparaison avec d'autres types d'écriture (le latin, le japonais...). En plus la cursivité de l'écriture arabe montre une complexité de la morphologie des caractères.

Un système de reconnaissance de l'écriture doit idéalement, localiser, reconnaître et interpréter n'importe quel texte ou nombre écrit sur un support de qualité arbitrairement variable tel que des cartes, des formulaires, des agendas, des vieux manuscrits, etc.

Parmi les domaines d'application, on trouve le domaine postal pour la reconnaissance du code de l'adresse postale et la lecture automatique des chèques bancaires ; le domaine administratif pour la gestion électronique des flux de documents ; les bibliothèques numériques pour l'indexation de documents et la recherche d'informations ; la biométrie pour l'identification du scripteur...etc.

I.2 Caractéristiques de l'écriture Arabe

L'arabe est écrit par plus de cent millions de gens, dans plus de vingt pays différents.

L'écriture arabe a été développée à partir d'un type d'Araméen. La langue araméenne comporte moins de consonants que l'arabe, alors de nouvelles lettres ont été créées en ajoutant des points aux lettres déjà existantes. D'autres petites marques appelées diacritiques sont utilisées pour indiquer de courtes voyelles, mais elles ne sont généralement pas utilisées [01].

L'arabe se distingue des autres écritures à différents points de vue. L'une des différenciations les plus importantes provient de sa semi-cursivité aussi bien dans sa forme imprimée que manuscrite. Elle est cursive, c'est-à-dire que les lettres sont liées généralement entre elles. Il n'y a pas de différence entre les lettres manuscrites et les lettres imprimées ; les notions de lettre capitale et lettre minuscule n'existent pas.

L'arabe est une écriture consonantique qui utilise un alphabet de 28 lettres auquel il faut ajouter la Hamza « ء », qui est le plus souvent considérée comme signe complémentaire [02]. La hamza « ء » a une orthographe spéciale qui dépend de règles grammaticales, ce qui multiplie les formes nécessaires à sa représentation, puisqu'elle peut s'écrire seule ou sur le support de trois voyelles (alif, waw et ya) dont elle suit le code (Tableau 1.1).

De plus l’alphabet arabe comprend d’autres caractères additionnels tels que « ة » et « لا », de ce fait, certains auteurs considèrent que l’alphabet arabe comprend plutôt 31 lettres que 29. La considération du symbole « ~ » qui s’écrit uniquement sur le support du caractère « ل », fait apparaître d’autres graphismes. L’écriture arabe a ainsi plusieurs spécificités que nous citons ci-après (Tableaux 1.1) [03].

N°	Lettre isolé	Au début	Au milieu	A la fin	Prononciation
01	ا / آ / إ	ا / آ / إ	ا / آ / إ	ا / ل / إ	Alif
02	ب	ب	ب	ب	Ba
03	ت	ت	ت	ت	Ta
04	ث	ث	ث	ث	Tha
05	ج	ج	ج	ج	Jim
06	ح	ح	ح	ح	Ha
07	خ	خ	خ	خ	Kha
08	د	د	د	د	Dal
09	ذ	ذ	ذ	ذ	Dhal
10	ر	ر	ر	ر	Ra
11	ز	ز	ز	ز	Zay
12	س	س	س	س	Sin
13	ش	ش	ش	ش	Chin
14	ص	ص	ص	ص	Sad
15	ظ	ظ	ظ	ظ	Dhad
16	ط	ط	ط	ط	TTa
17	ظ	ظ	ظ	ظ	Dha
18	ع	ع	ع	ع	Ayn
19	غ	غ	غ	غ	Ghayn
20	ف	ف	ف	ف	Fa
21	ق	ق	ق	ق	Qaf
22	ك	ك	ك	ك	Kaf
23	ل	ل	ل	ل	Lam
24	م	م	م	م	Mim
25	ن	ن	ن	ن	Noun
26	ه	ه	ه	ه	He
27	و	و	و	و	Waw
28	ي	ي	ي	ي	Ya
Autre formes de caractères (complémentaires, combinaison)					
29	ء	ء / ء	ء	ء	Hamza
30	لا	لا	لا / لا	لا	Madda + alif
31	لا / لا	لا / لا	لا / لا	لا / لا	Lam alif + hamza
32	ة			ة	Ta marbouta
33	لا	لا	لا	لا / لا	Lam alif

Tableau 1.1 Alphabet arabe dans ses différentes formes

Un trait caractéristique de l'écriture arabe est la présence d'une ligne de base horizontale dite encore ligne de référence ou d'écriture. C'est le lieu des caractères (Figure 1.1).



Figure 1.1 Exemple d'écriture arabe montrant la ligne de base

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 introduit une superposition verticale entre la lettre qui comprend le descendant et la lettre suivante. Le tableau 1.2 donne les différentes lettres arabes possédant des hampes et des jambages.

	Hampes						
Lettres	ا	ل	ك	و	ي	لا	ط
	Jambage						
Lettres	و	و	م	و	و	و	
	و	س	ط	و	ع	م	

Tableau 1.2. Différents hampes et jambages dans les lettres arabes.

Certains caractères ne peuvent être rattachés à leur gauche, ainsi ils ne peuvent se trouver qu'en position isolée ou finale ; ce qui donnent quand ils existent, des mots composés d'une ou de plusieurs parties appelés généralement PAW (Peace of Arabic Word) ou encore pseudo-mot

[03]. Un PAW correspond donc à une chaîne d'un ou de plusieurs caractères (Tableau 1.3).

3 PAW/mot	2 PAW/mot	1 PAW/mot
القائمة	شمال	نحل

Tableau 1.3 Exemple de mots composés de la droite vers la gauche de 1, 2, 3 PAWs

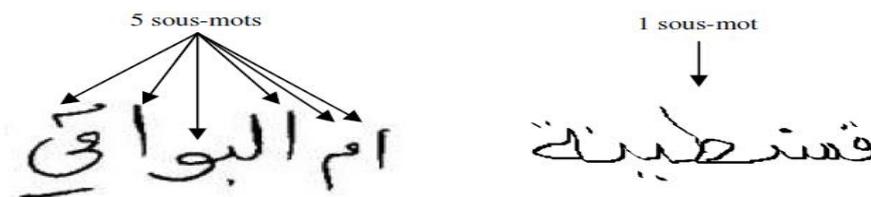


Figure 1.2. Exemple de pseudo-mots constituant des mots arabes.

Les caractères arabes ne possédant pas une taille fixe (hauteur et largeur), leur taille varie d'un caractère à un autre et d'une forme à une autre à l'intérieur d'un même caractère.

Pour des raisons de justification de texte et/ou d'esthétique, les ligatures horizontales peuvent être allongées en insérant entre les caractères d'une même chaîne une ou plusieurs elongations « madda » (ou tatwil), correspondant au symbole « - ». L'élongation se situe toujours à gauche du caractère coûtant. Si le trait d'allongement est associé à un caractère en position début ou finale, le caractère prend sa forme du milieu et voit sa chasse augmenter du nombre de « madda » insérées (Tableau 1.4) [04].

Au niveau du PAW, l'insertion de traits d'allongement affecte uniquement sa largeur, la morphologie reste la même [05]. Les éditeurs de texte tels que Word de Microsoft, insèrent dans les lignes de texte, le nombre approprié de « Madda », pour la justification gauche-droite d'un texte arabe.

Avec 6 maddas	Avec 3 maddas	Avec 1 madda	Sans madda
قـــــــــــــــــ	قـــــــــ	قـ	ق
تـــــــــــــــــ	تـــــــــ	تـ	ت

Tableau 1.4 Exemple de caractères avec et sans madda

L'écriture arabe est semi-cursive dans sa forme imprimée ainsi que manuscrite. Les caractères d'une même chaîne (ou pseudo-mots) sont ligaturés horizontalement et parfois verticalement (dans certaines fontes deux, trois et même quatre caractères peuvent être ligaturés verticalement), rendant difficile la segmentation en caractères.

{ قـ , جـ , حـ , خـ }	{ فـ , جـ , حـ , خـ }	{ لـ , جـ , حـ , خـ }	{ مـ , جـ , حـ , خـ }
-----------------------	-----------------------	-----------------------	-----------------------

Tableau 1.5 Caractères susceptibles d'être ligaturés verticalement

Ligatures obligatoires des lettres : لمحة { ل م ح ة }

Ligature esthétique entre les 2 premières lettres : لمحة

Ligature esthétique entre les 3 premières lettres : لمحة

L'écriture arabe nécessite un grand nombre de ligatures pour être composée de façon satisfaisante .On peut classifier les ligatures typographiques en trois types qui sont :

- Les ligatures linguistiques, ce sont celles qui sont obligatoires. Elles obéissent à des règles grammaticales. L'exemple le plus pertinent est la ligature arabe Lam-Alif.
- Les ligatures esthétiques, ce sont celles qui ne sont pas indispensables et que l'on peut remplacer par leurs composantes non liées.
- Les ligatures contextuelles, ce sont les variantes de position dans un processus de composition de texte.

Contrairement à l'écriture latine, les articles (le, la, les) font partie du mot auquel ils sont rattachés. La séquence "ال " (un pseudo mot qui contient la lettre "ا " isolée, suivi d'un autre pseudo-mot qui commence par la lettre "ل") correspond nécessairement au début d'un mot [06]. La figure 1.3 indique le commencement des deux mots arabes "الرحمن" et "الرحيم" par cette séquence.



Figure 1.3. L'article défini "ال " dans deux mots arabes.

De plus, la forme d'un caractère diffère selon sa position dans les pseudo-mots et même dans certains cas ; selon le contexte phonétique. En outre, plus de la moitié des caractères arabes incluent dans leur forme des points diacritiques. Ces points peuvent se situer au-dessus ou au-dessous du caractère, mais jamais en haut et en bas simultanément. Plusieurs caractères peuvent avoir le même corps mais un nombre et /ou une position de points diacritiques différents.

Dans l'alphabet arabe, 15 lettres parmi les 28 possèdent un ou plusieurs points. Le nombre maximal de points que peut avoir une lettre est de trois points au-dessus du caractère, ou deux points en dessous. Ces points permettent de différencier la prononciation des lettres arabes. Le tableau 1.6 présente les lettres ayant des points diacritiques ainsi que leurs nombres et positions.

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

Tableau 1.6. Lettres arabes ayant des points diacritiques

Les caractères arabes peuvent être voyellés. Les voyelles appelées aussi diacritiques dans certains documents et courtes voyelles dans d'autres, tels que [07], peuvent se placer au-dessus ou en dessous du caractère. Les voyelles sont d'une invention postérieure aux consonnes.

Dans l'arabe contemporain ordinaire, on écrit seulement les consonnes et les voyelles longues. Un même mot avec différentes voyelles courtes peut être compris comme verbe, nom ou adjectif. Par d'exemple « علم » peut signifier « drapeau : عِلْمٌ » ou « savoir : عِلْمٌ » ou encore «

Enseigner : عَلَّمَ », selon sa voyellation.

Il existe 8 signes de voyellation qui peuvent se placer au-dessus de la ligne d'écriture, tels que fatha (َ) dammah (ُ) , soukoun (ْ) et chaddah (ّ) qui doit être accompagnée de l'une des voyellations fatha, Dammah ou kasrah, et ceux qui peuvent se placer en dessous de la ligne d'écriture tels que Kasrah (ِ) . De plus trois «tanwin» peuvent être formés à partir d'un double fatha (ً) , d'un double dammah (ٌ) ou d'un double kasrah (ٍ) .

Selon la caractéristique précédente, on peut distinguer deux types de textes: les textes avec ou sans les signes de voyelles. Quelques textes arabes (le sacré Coran et les livres d'apprentissage de la lecture et de l'écriture pour les enfants) contiennent des signes de voyelles tandis que les autres tels que les journaux et les publications sont des textes sans signes. Le même texte sans et avec voyelles est présenté dans la figure 1.4.

<p>إن الحمد لله نحمده و نستعينه و نستغفره و نعوذ بالله من شرور أنفسنا و سيئات أعمالنا من يهده الله فلا مضل له ومن يضلل فلا هادي له و أشهد أن لا إله إلا الله وحده ل شريك له</p>	<p>إِنَّ الْحَمْدَ لِلَّهِ نَحْمَدُهُ وَنَسْتَعِينُهُ وَنَسْتَغْفِرُهُ وَنَعُوذُ بِاللَّهِ مِنْ شُرُورِ أَنْفُسِنَا وَسَيِّئَاتِ أَعْمَالِنَا مَنْ يَهْدِهِ اللَّهُ فَلَا مُضِلَّ لَهُ وَمَنْ يَضِلَّ فَلَا هَادِيَ لَهُ وَأَشْهَدُ أَنْ لَا إِلَهَ إِلَّا اللَّهُ وَحْدَهُ لَا شَرِيكَ لَهُ وَأَشْهَدُ أَنَّ مُحَمَّدًا عَبْدُهُ وَرَسُولُهُ</p>
---	---

Figure 1.4. Deux types de textes arabes avec et sans signes de voyelles.

L'écriture arabe contient beaucoup de polices et modèles d'écriture, ainsi il est difficile parfois de séparer un mot d'un autre, particulièrement quand les gens écrivent avec la calligraphie.

L'écriture arabe est connue pour sa richesse en fontes et styles. Il existe environ 450 fontes d'écritures différentes dont seulement quelques-unes sont couramment utilisées dans le monde arabo-musulman, nous citons à titre d'exemple : le Neskhi, Thoulthi, Roqa, Diwani,



Koufi, Farsi, etc. Le Neskhi demeure aujourd'hui la fonte la plus utilisée pour l'écriture imprimée. Chaque style arabe est régi par des lois particulières. D'un style à un autre, les proportions d'une même lettre et son dessin peuvent changer considérablement. Par exemple dans la fonte Roqa la lettre Alif est plus petite que la lettre Ain, mais dans la fonte Koufi c'est l'inverse. La figure 1. 5 présente la manière d'écriture des caractères arabes "ا", "ك" et "ي" dans différentes fontes.

Figure 1.5. Exemples de caractères arabes écrits suivant différentes fontes.



Figure 1.6 Différentes phrases arabes dans différents modèles [08]

Plusieurs groupes de caractères possèdent le même corps mais un nombre et/ou un emplacement de points diacritiques différents. Le tableau 1.7 affiche les caractères ayant le même corps. Comme il est indiqué dans le tableau 1.7, le caractère Alif- Maqsoora "ى", partage le même corps que le caractère "ي" mais sans points diacritiques, ce caractère apparait seulement à la fin du mot. Exemples : "مبتغى", "ليلى", etc.

د → { د ذ }	ر → { رة ژ زي }	س → { سه سد }
ص → { صد صذ }	ز → { رز }	ح → { جد حذ }
و → { فوق }	ط → { طظ }	ع → { عغ }
ع → { عغ }	ى → { ى ي }	ه → { هه }

Tableau 1.7. Lettres arabes ayant des corps identiques.

Par ailleurs, la cursivité de l'écriture arabe montre une complexité de la morphologie des caractères, les élongations des ligatures horizontales ainsi que les combinaisons verticales de certains caractères, constituent les problèmes majeurs liés au traitement de cette écriture surtout pour les pseudo-mots.

En effet, ces problèmes engendrent une forte inertie à différents niveaux notamment :

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

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

Face à ces problèmes, la nécessité d'une modélisation robuste une méthode d'apprentissage efficace pour prendre en considération toutes les variations morphologiques de l'écriture arabe.

I.3 Différents aspects de reconnaissance de l'écriture

Il n'existe pas de système universel d'OCR qui permet de reconnaître n'importe quel caractère dans n'importe quelle fonte. Tout dépend du type de données traitées et bien évidemment de l'application visée [09]. Il existe plusieurs modes de classification des systèmes OCR parmi lesquels on peut citer :

- Les systèmes qualifiés de « en-ligne » ou « hors-ligne » suivant le mode d'acquisition.
- Les approches globales ou analytiques selon que l'analyse s'opère sur la totalité du mot, ou par segmentation en caractères.
- Les approches statistiques, structurelles ou stochastiques relatives aux traits caractéristiques extraits des formes considérées.

I.3.1 Reconnaissance en-ligne et hors-ligne

Ce sont deux modes différents d'OCR, ayant chacun ses outils propres d'acquisition et ses algorithmes correspondants de reconnaissance.

a) La reconnaissance en-ligne (on-line) :

Ce mode de reconnaissance s'opère en temps réel (pendant l'écriture). Les symboles sont reconnus au fur et à mesure qu'ils sont écrits à la main. Ce mode est réservé généralement à l'écriture manuscrite. C'est une approche « signal » où la reconnaissance est effectuée sur des données à une dimension. L'écriture est représentée comme un ensemble de points dont les coordonnées sont fonction du temps [10], [11].

La reconnaissance en-ligne présente un avantage majeur c'est la possibilité de correction et de modification de l'écriture de manière interactive vu la réponse en continu du système [12].

L'acquisition de l'écrit est généralement assurée par une tablette graphique munie d'un stylo électronique.

b) La reconnaissance hors-ligne (offline) :

Démarre après l'acquisition. Elle convient aux documents imprimés et les manuscrits déjà rédigés. Ce mode peut être considéré comme le cas le plus général de la reconnaissance de l'écriture. Il se rapproche du mode de la reconnaissance visuelle. L'interprétation de l'information est indépendante de la source de génération [13].

La reconnaissance hors-ligne peut être classée en plusieurs types :

- ❖ **Reconnaissance de texte ou analyse de documents** : Dans le premier cas il s'agit de reconnaître un texte de structure limitée à quelques lignes ou mots. La recherche consiste en un simple repérage des mots dans les lignes, puis à un découpage de chaque mot en caractères [09]. Dans le second cas (analyse de document), il s'agit de données bien structurés dont la lecture nécessite la connaissance de la typographie et de la mise en page du document. Ici la démarche n'est plus un simple prétraitement, mais une démarche experte d'analyse de document il y'a localisation des régions, séparation des régions graphiques et photographique, étiquetage sémantique des zones textuelles à partir de modèles, détermination de l'ordre de lecture et de la structure du document [14].
- ❖ **Reconnaissance de l'imprimé ou du manuscrit** : Les approches diffèrent selon qu'il s'agisse de reconnaissance de caractères imprimés ou manuscrits. Les caractères imprimés sont dans le cas général alignés horizontalement et séparés verticalement, ce qui simplifie la phase de lecture [09]. La forme des caractères est définie par un style calligraphique (fonte) qui constitue un modèle pour l'identification. Dans le cas du manuscrit, les caractères sont souvent ligaturés et leur graphisme est inégalement

proportionné provenant de la variabilité intra et inter scripteurs. Cela nécessite généralement l'emploi de techniques de délimitation spécifiques et souvent des connaissances contextuelles pour guider la lecture [15].

Dans le cas de l'imprimé, la reconnaissance peut être monofonte, multifonte ou omnifonte.

Un système est dit monofonte s'il ne peut reconnaître qu'une seule fonte à la fois c'est à dire qu'il ne connaît de graphisme que d'une fonte unique. C'est le cas le plus simple de reconnaissance de caractères imprimés [16].

Un système est dit multifonte s'il est capable de reconnaître divers types de fontes parmi un ensemble de fontes préalablement apprises [09].

Et un système omnifonte est capable de reconnaître toute fonte, généralement sans apprentissage préalable. Cependant ceci est quasiment impossible car il existe des milliers de fontes dont certaines illisibles par l'homme (sauf bien sûr pour celui qui l'a conçue) et avec un logiciel de création de fonte n'importe qui peut concevoir des fontes à sa guise [16]. Anigbogu [16] a présenté une autre définition pour ce terme c'est l'expression « *polyfonte* » et a qualifié un système polyfonte de système capable de reconnaître un très grand nombre de fontes.

Dans le cas du manuscrit, la reconnaissance peut être mono-scripteur, multi-scripteur ou omni-scripteur. L'écriture manuscrite hors-ligne peut être classée en deux catégories d'écritures : écriture cursive et écriture semi-cursive.

- Un système est dit Mono-scripteur (propre au scripteur) : c'est le fait que le système ne peut reconnaître qu'une seule écriture. Tous ces éléments influent sur la forme des lettres (écriture penchée, bouclée, arrondie, linéaire, etc.) et bien sûr sur la forme des ligatures, compromettant parfois le repérage des limites entre lettres.
- Un système est dit Multi-scripteur (propre à l'écriture manuscrite) : c'est que le système peut identifier et reconnaître l'écriture pour un certain nombre de scripteurs.
- Un système est dit Omni-scripteur (propre à n'importe quelle écriture manuscrite) : c'est le fait de réduire l'information contenue dans l'image au minimum nécessaire pour modéliser précisément la structure des caractères [17].

I.3.2 Reconnaissance globale ou analytique

Deux approches s'opposent en reconnaissance des mots : globale et analytique.

- a) **Approche globale** : considère le mot comme une seule entité et le décrit indépendamment des caractères qui le constituent. Cette approche présente l'avantage

de garder le caractère dans son contexte avoisinant, ce qui permet une modélisation plus efficace des variations de l'écriture et des dégradations qu'elle peut subir. Cependant cette méthode est pénalisante par la taille mémoire, le temps de calcul et la complexité du traitement qui croient linéairement avec la taille du lexique considéré, d'où une limitation du vocabulaire [11].

b) **Approche analytique** : contrairement à l'approche globale, le mot est segmenté en caractères ou en fragments morphologiques significatifs inférieurs aux caractères appelés graphèmes. La reconnaissance du mot consiste à reconnaître les entités segmentés puis tendre vers une reconnaissance du mot, ce qui constitue une tâche délicate pouvant générer différents types d'erreurs [10]. Un processus de reconnaissance selon cette approche est basé sur une alternance entre deux phases : la phase de segmentation et la phase d'identification des segments. Deux solutions sont alors possibles : la segmentation explicite (externe) ou la segmentation implicite (interne) [18]. Par ailleurs, les méthodes analytiques par opposition aux méthodes globales, présentent l'avantage de pouvoir se généraliser à la reconnaissance d'un vocabulaire sans limite à priori, car le nombre de caractères est naturellement fini. De plus l'extraction des primitives est plus aisée sur un caractère que sur une chaîne de caractères [19].

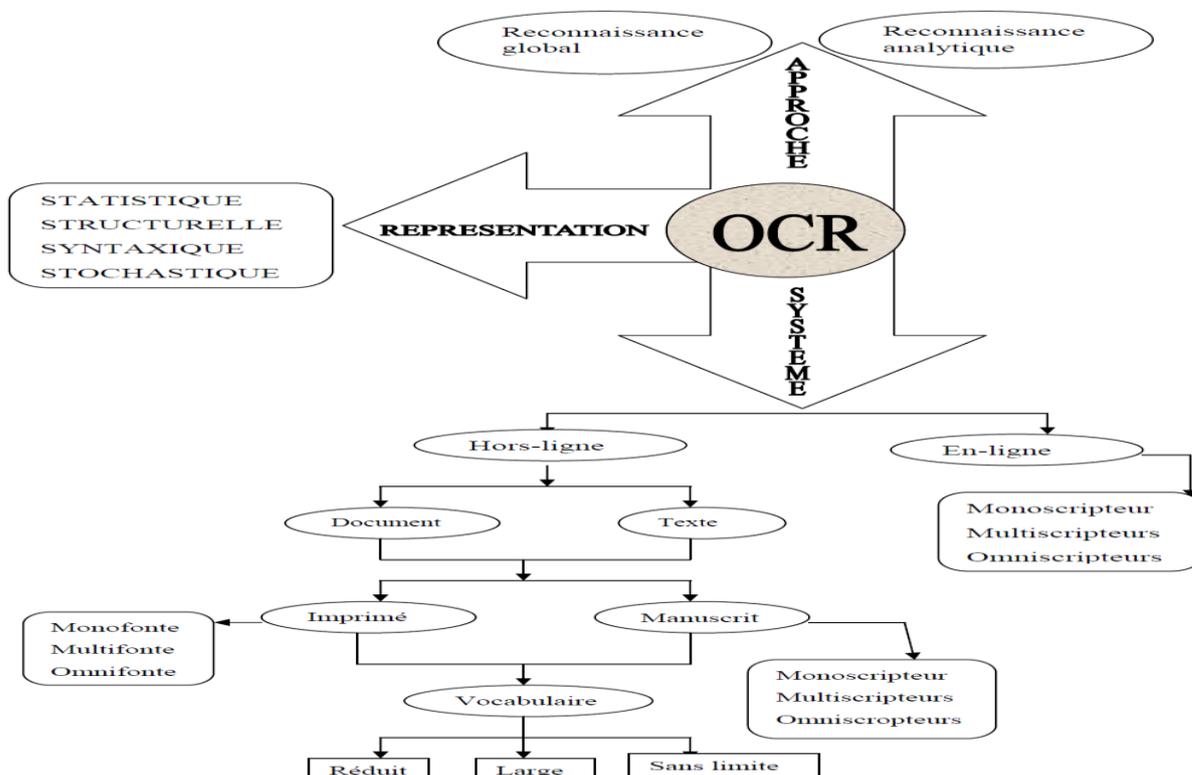


Figure I.7 : Différents systèmes, représentations et approches de reconnaissance .Tiré de [09]

I.4 Organisation générale d'un système de reconnaissance

Un système de reconnaissance fait appel généralement aux étapes suivant : acquisition, prétraitement, segmentation, extraction des caractéristiques, classification, suivi éventuellement d'une phase de post-traitement.

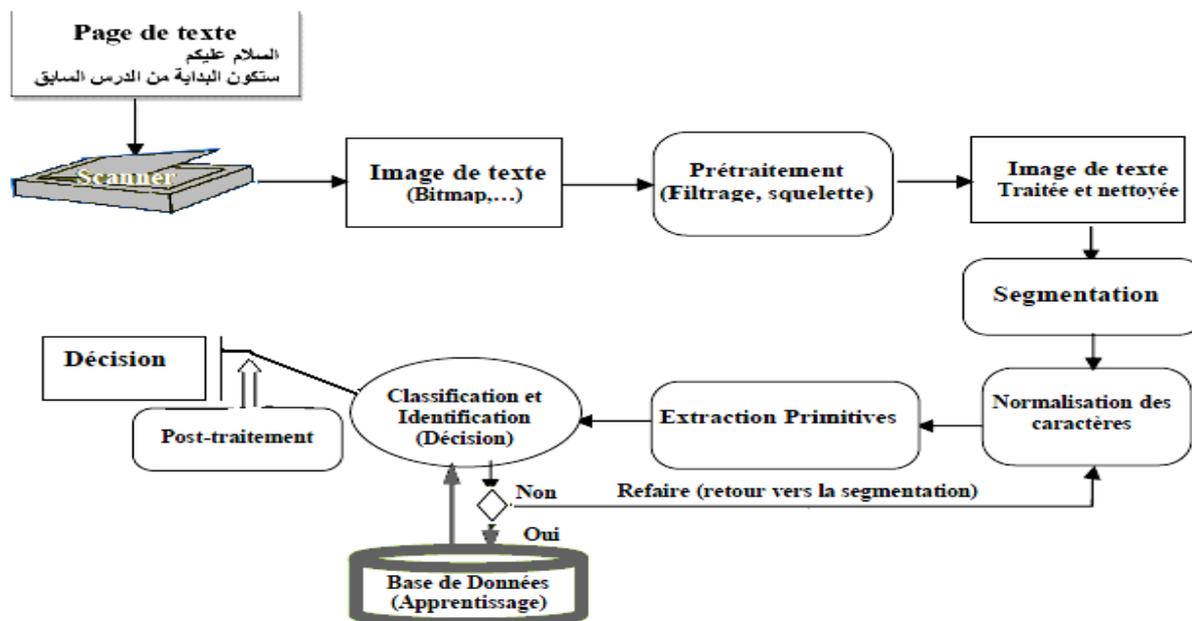


Figure 1.8 : Schéma général du système de reconnaissance des caractères.

I.4.1 Phase d'acquisition

Cette phase consiste à capter l'image d'un texte au moyen des capteurs physiques (scanner, caméra,...) et de la convertir en grandeurs numériques adaptées au système de traitement, avec un minimum de dégradation possible.

I.4.1.1 Etape d'acquisition

Elle consiste en deux phases :

- **Echantillonnage (numérisation)** d'une image est spatiale, par découpage en pixels.
- **Quantification (codage)** : est une valeur numérique donnée à l'intensité lumineuse, c'est un niveau de gris, appelé la dynamique de l'image.

Cette dynamique est donnée comme suit : 2^m , où m est le nombre de bits. Par exemple : le niveau de gris 256 est codé sur 8 bits, l'image couleur est codée sur 24 bits (1 octet pour chaque couleur(R, V, B)) [20].

I.4.1.2 Notion de voisinage

Dans le domaine de traitement des images on distingue deux types de voisinage :

- Voisinage à 4 : ensemble de pixels 'p' qui ont un coté en commun avec le pixel considéré.

- Voisinage à 8 : ensemble de pixels 'p' qui ont au moins un point de liaison avec le pixel considéré.

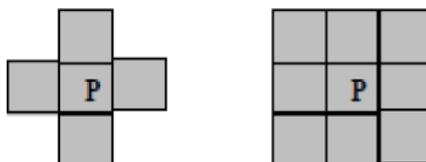


Figure 1.9 Voisinage à 4 et Voisinage à 8

I.4.2 Phase de prétraitements

Le prétraitement consiste à préparer les données issues du capteur à la phase suivante. Il s'agit essentiellement de réduire le bruit superposé aux données et essayer de ne garder que l'information significative de la forme représentée. Le bruit peut être dû aux conditions d'acquisition (éclairage, mise incorrecte du document, ...) ou encore à la qualité du document d'origine.

Parmi les opérations de prétraitement généralement utilisées on peut citer : la binarisation, la dilatation, l'érosion, la squelettisation et la normalisation (figure 1.10).

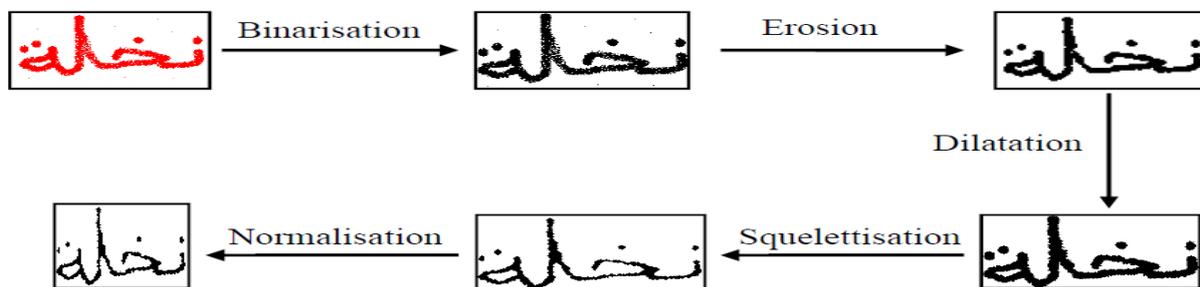


Figure 1.10 Effets de certaines opérations de prétraitement .

I.4.2.1 Binarisation

La binarisation c'est le passage d'une image en couleur, définie par plusieurs niveaux de gris en image bitonale (composée de deux valeurs 0 et 1) qui permet une classification entre le fond (image du support papier en blanc) et la forme (traits des gravures et des caractères en noir). Plusieurs techniques ont été développées dans le but de transformer une image à niveaux de gris ou en couleur, en image binaire. Toutes ces techniques sont basées sur le principe de seuillage comme le montre l'équation suivante :

$$I_b(x,y) = \begin{cases} 0 & \text{si } I_n(x,y) < T \\ 1 & \text{si } I_n(x,y) \geq T \end{cases}$$

$I_n(x, y)$ décrit l'intensité à "n" niveaux de gris à chaque point de l'image, $I_b(x, y)$ représente l'intensité à deux niveaux et T est le seuil de binarisation. Si $I_n(x,y)$ est supérieur à la valeur de seuil alors on attribue le point image correspondant à la valeur d'intensité maximale (le blanc). Dans le cas contraire, le point est considéré comme noir et on lui attribue la valeur d'intensité minimale [15].

Pour des images de niveaux de gris, on peut trouver dans [21] une liste des méthodes de binarisation, proposant des seuils adaptatifs (ex. s'adaptant à la différence de distribution des niveaux de gris).



Figure 1.11 Exemple de Binarisation adaptative [22]

I.4.2.2 Transformation par érosion

Si un pixel 'p' est noir ($p(x,y)=0$), et il y a au moins 3 pixels, ou 7 pixels dans les 4-voisins, ou les 8-voisins respectivement, qui sont blancs ($P_{voisin}(x,y)=1$), alors on affecte à ce pixel la couleur blanche ($p(x,y)=1$), c-à-d. on efface ce pixel, (C'est un lissage d'un ou deux pixels d'une forme connexe) [15].

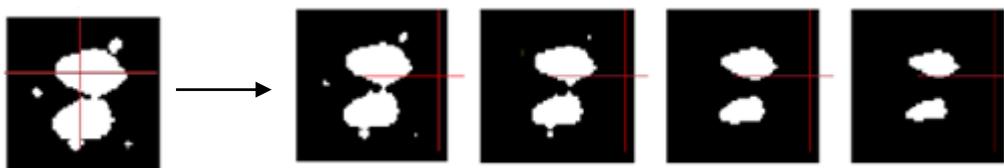


Figure 1.12 Image traitée par érosion

I.4.2.3 Transformation par dilatation

C'est l'inverse de la transformation par érosion, si un pixel 'p' est blanc ($p(x,y)=1$), et il y a au moins 3 pixels, ou 7 pixels dans les 4-voisins, ou les 8-voisins respectivement, qui sont noirs ($P_{voisin}(x,y)=0$), alors on affecte à ce pixel la couleur noir ($p(x,y)=0$).

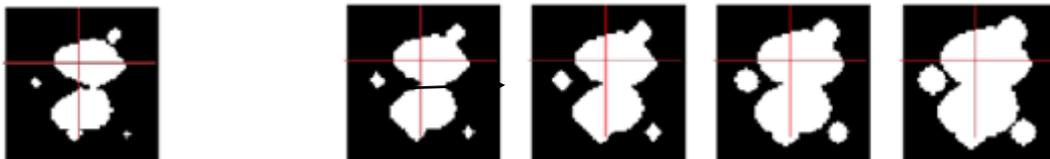


Figure 1.13 Image traitée par dilatation

I.4.2.4 Ouverture morphologique

C'est une combinaison d'opérations : érosion suivie d'une dilatation d'une image par le même élément structurant. Servant à adoucir les contours, simplifier les formes en lissant les bosses tout en conservant l'allure globale [15].



Figure 1.14 Image traitée par Ouverture

I.4.2.5 Fermeture morphologique

C'est le contraire de l'ouverture, elle consiste en une combinaison d'opérations : une dilatation suivie d'érosion d'une image par le même élément structurant. Elle permet de simplifier les formes, en comblant les creux.

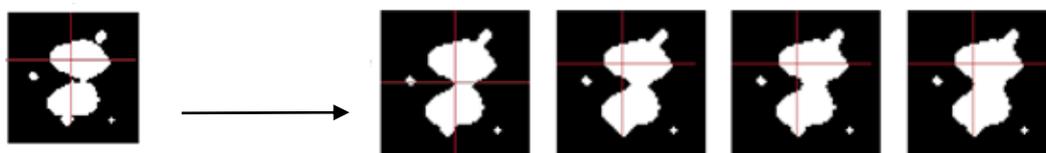


Figure 1.15 Image traitée par Fermeture

I.4.2.6 Squelettisation :

Le but de cette technique est de simplifier l'image du caractère en une image à « ligne » plus facile à traiter en la réduisant au tracé du caractère. Les algorithmes de squelettisation se basent sur des méthodes itératives. Le processus s'effectue par passes successives pour déterminer si un tel ou tel pixel est essentiel pour le garder ou non dans le tracé [23].

La squelettisation des caractères arabes peut induire en erreur : deux points diacritiques sont souvent confondus avec un seul [09].

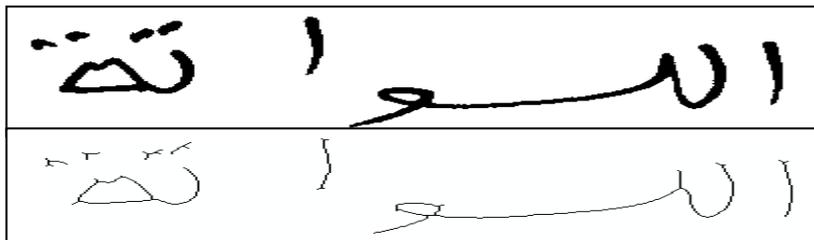


Figure 1.16 La squelettisation d'une Image [28]

I.4.2.7 Normalisation

L'étude de discernement nécessite l'élimination des conditions qui peuvent fausser les résultats, comme la différence de taille. Il faut donc aboutir à la normalisation de la taille des caractères. Après cette opération, les images de tous les caractères se retrouvent définies dans une matrice de même taille, pour faciliter les traitements ultérieurs. Cette opération introduit généralement de légères déformations sur les images. Cependant certains traits caractéristiques tels que la hampe dans des caractères (ا ل ط ظ par exemple) peuvent être éliminées à la suite de la normalisation, ce qui peut entraîner à des confusions entre certains caractères [24].

Une application doit être définie pour effectuer cette tâche. Cette application doit être réversible pour pouvoir aller d'une taille à une autre et revenir par la suite.

$G : \mathbb{R}^4 \longrightarrow$ ensemble des couleurs

$(x, y) \longrightarrow$ couleur (noir/blanc)

G est défini de la façon suivante :

$$\begin{cases} G(x,y) = \text{noir} & \text{si pixel } (\mu_1 x, \mu_2 y) = \text{noir} \\ G(x,y) = \text{blanc} & \text{sinon} \end{cases}$$

Tel que : $\mu_1 = (\text{largeur du cadre du caractère}) / (\text{largeur du cadre de la normalisation})$

$\mu_2 = (\text{hauteur du cadre du caractère}) / (\text{hauteur du cadre de la normalisation})$

I.4.3 Phase de segmentation

La segmentation est une opération appliquée à l'image qui consiste à subdiviser une scène réelle, en parties constituantes ou objets, en projetant une scène réelle sur un plan. Elle est la première opération à réaliser dans « la reconnaissance des formes ». Il faut donc, disposer

d'un certains nombres d'attributs, représentatifs des régions que l'on cherche à extraire, pour procéder à la classification individuelle des points [05].

I.4.3.1 Techniques de la segmentation

Il existe deux techniques permettant de mettre en œuvre la segmentation. La première, connue sous le nom de segmentation implicite, et la deuxième c'est la segmentation explicite.

a- Segmentation implicite

Les méthodes de segmentation implicite s'inspirent des approches utilisées dans le domaine de la parole, où le signal est divisé en intervalles de temps réguliers, et procèdent à une sur-segmentation importante de l'image du mot à pas fixe (un ou quelques pixels). Cela permet d'assurer un taux de présence important des points de liaison entre lettres considérées. La segmentation s'effectue pendant la reconnaissance qui assure son guide. Le système recherche dans l'image, des composantes ou des groupements de graphèmes₁ qui correspondent à ses classes de lettres [25]. Classiquement, il peut le faire de deux manières :

- soit par fenêtrage : le principe est d'utiliser une fenêtre mobile de largeur variable (qui n'est pas facile à déterminer) pour trouver des séquences de points de segmentation potentiels qui seront confirmés ou non par la reconnaissance de caractères. Elle nécessite deux étapes : la génération d'hypothèses de segmentation (séquences de points obtenus par le fenêtrage) ; la deuxième est le choix de la meilleure hypothèse de la reconnaissance (validation).
- soit par recherche de primitives : il s'agit de détecter les combinaisons de primitives qui donneront la meilleure reconnaissance.

b- Segmentation explicite

Cette approche, souvent appelée dissection, est antérieure à la reconnaissance et n'est pas remise en cause pendant la phase de reconnaissance. Les hypothèses des caractères sont déterminées à partir des informations de bas niveau présentes sur l'image. Ces hypothèses sont définitives, et doivent être d'une grande fiabilité car la moindre erreur de segmentation remet en cause la totalité des traitements ultérieurs.

Les approches de segmentation explicite, s'appuient sur une analyse morphologique du mot manuscrit pour localiser des points de segmentation potentiels. Elles sont particulièrement adaptées à l'analyse de la représentation bidimensionnelle et donc plus souvent utilisées dans les systèmes de reconnaissance hors-ligne de mots. Certaines méthodes de segmentation explicite sont basées sur une analyse par morphologiques mathématiques, exploitent les

concepts de régularité et singularité du tracé, analyse des contours supérieurs/inferieurs du mot. Les points de segmentation potentiels détectés sont confirmés à l'aide de diverses heuristiques [25].

I.4.3.2 Etapes de segmentation

- a) **Segmentation du texte en lignes** Les méthodes de traitement de l'arabe utilisent souvent la projection horizontale pour extraire les lignes. Cependant la présence des points diacritiques complique cette extraction et conduit parfois à la confusion des lignes [26]. Ce problème a lieu quand l'interligne est calculée par une simple moyenne des différentes interlignes. Pour remédier à ce problème, certains auteurs tels que [27] identifient d'abord les différentes lignes d'écriture, ensuite regroupent les blocs de texte d'après leur proximité par rapport aux lignes d'écriture déjà localisées.

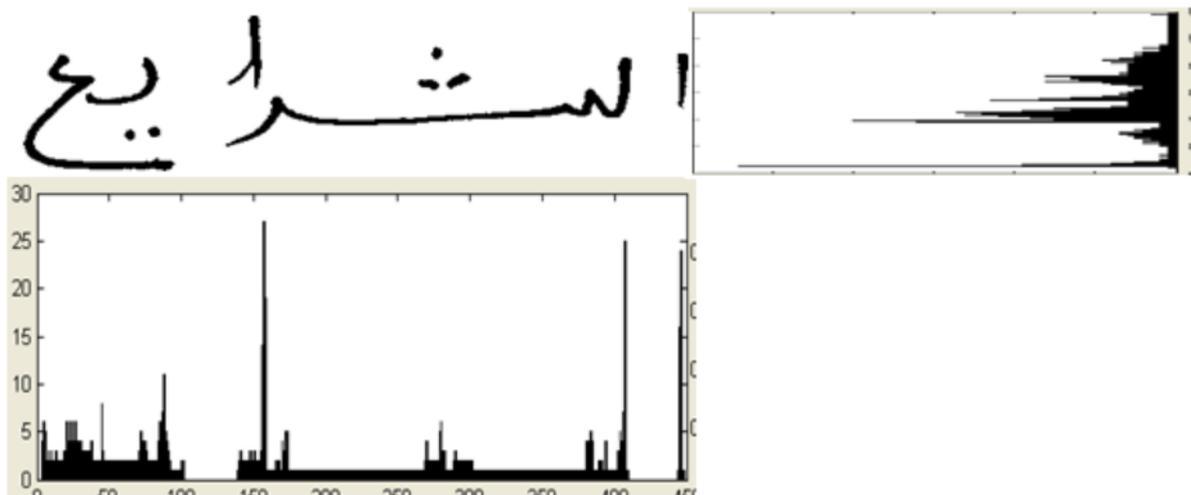


Figure 1.17 Exemple d'histogramme horizontal et vertical d'une ligne de texte [28].

- b) **Segmentation en Pseudo mot (PAW : Peace of Arabic Word)** Elle est réalisée en déterminant l'histogramme des projections verticales des différentes lignes de texte [04]. Cependant, cette méthode n'est pas efficace dans le cas où les PAWs se chevauchent verticalement. Dans ce cas, d'autres techniques sont utilisées telles que la détermination du contour [29], du squelette [25], ou encore des composantes connexes [30]. Le choix de la technique est souvent guidé par la méthode d'analyse [25],[04].
- c) **Segmentation de la ligne de texte en mots** En arabe OCR, la segmentation est souvent réservée à l'extraction des PAWs, le mot est plutôt considéré dans la phase de post-traitement (si elle est prévue) pour valider les résultats trouvés ou corriger les

erreurs de reconnaissance. Par ailleurs afin d'éviter le problème d'une segmentation erronée en mots, certains auteurs introduisent dans leurs systèmes, un seul mot à la fois [25],[04].

d) Segmentation du mot en caractères La segmentation en caractères (ou en graphèmes) constitue le problème le plus ardu lié à la reconnaissance de l'écriture arabe. Les difficultés à ce niveau sont du même type que celles affrontées lors de la reconnaissance du latin manuscrit (cursif), mais souvent plus complexes à cause de la diversité des formes du caractère arabe, de la courte liaison qui existe entre les caractères successifs, de l'allongement des ligatures horizontales et de la présence des ligatures verticales [04],[31].

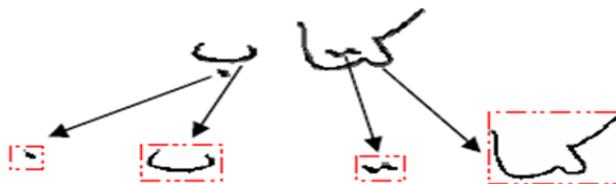


Figure 1.18 Exemple de segmentation [2]

I.4.4 Phase d'extraction des caractéristiques

C'est l'une des étapes les plus délicates et les plus importantes en OCR. Les types des caractéristiques peuvent être classés en quatre groupes principaux : caractéristiques structurelles, caractéristiques statistiques, transformations globales, et superposition des modèles et corrélation [32] [11].

I.4.4.1 Caractéristiques structurelles

Les caractéristiques structurelles décrivent une forme en termes de sa topologie et sa géométrie en donnant ses propriétés globales et locales. Parmi ces caractéristiques on peut citer :

- Les traits et les anses dans les différentes directions ainsi que leurs tailles.
- Les points terminaux.
- Les points d'intersections.
- Les boucles.
- Le nombre de points diacritiques et leur position par rapport à la ligne de base.
- Les voyellations et les zigzags (hamza).
- La hauteur et la largeur du caractère.

- La catégorie de la forme (partie primaire ou point diacritique, etc).

I.4.4.2 Caractéristiques statistiques

Les caractéristiques statistiques décrivent une forme en termes d'un ensemble de mesures extraites à partir de cette forme. Les caractéristiques utilisées pour la reconnaissance de textes arabes sont : le zonage (zoning), les caractéristiques de lieu géométrique (Loci) et les moments [32].

La méthode Loci est basée sur le calcul du nombre de segments blancs et de segments noirs le long d'une ligne verticale traversant la forme, ainsi que leurs longueurs [11].

I.4.4.3 Transformations globales

Elles sont naturellement basées sur une transformation globale de l'image. La transformation consiste à convertir la représentation en pixels en une représentation plus abstraite pour réduire la dimension des caractères, tout en conservant le maximum d'informations sur la forme à reconnaître. Par exemple : la transformée de Hough, la transformée de Fourier, et les moments de Zernike [25].

I.4.4.4 Superposition des modèles (template matching) et corrélation

La méthode de 'template matching' appliquée à une image binaire (en niveaux de gris ou squelettes), consiste à utiliser l'image de la forme comme vecteur de caractéristiques pour être comparé à un modèle (template) pixel par pixel dans la phase de reconnaissance, et une mesure de similarité est calculée [32].

I.4.5 Phase de classification

La classification dans un système OCR regroupe deux tâches : l'apprentissage et la reconnaissance (décision). A cette étape les caractéristiques de l'étape précédente sont utilisées pour identifier un caractère et l'attribuer à un modèle de référence [33].

I.4.5.1 Apprentissage

Il s'agit lors de cette étape d'apprendre au système les propriétés pertinentes du vocabulaire utilisé et de l'organiser en modèles de références. L'idéal serait d'apprendre au système autant d'échantillons que de formes d'écritures différentes, mais cela est impossible à cause de la grande variabilité de l'écriture qui conduirait à une explosion combinatoire de modèles de représentation. La tendance consiste alors à remplacer le nombre par une meilleure qualité des traits caractéristiques [04],[33].

L'apprentissage consiste en deux concepts différents ; l'entraînement et l'adaptation.

L'entraînement consiste à enseigner au système la description des caractères tandis que l'adaptation sert à améliorer les performances du système en profitant des expériences précédentes. Certains systèmes permettent à l'utilisateur d'identifier un caractère lorsqu'ils échouent à le reconnaître et ils utilisent l'entrée de l'utilisateur à chaque fois que le caractère est rencontré [34] [35].

Les processus d'apprentissage sont différents selon qu'il s'agisse de reconnaissance de caractères imprimés ou manuscrits ou de reconnaître des textes monospace ou multispaces.

D'une manière générale, on distingue deux types de techniques d'apprentissage : supervisé et non supervisé.

- L'apprentissage est dit supervisé s'il est guidé par un superviseur appelé professeur. Il est réalisé lors d'une étape préliminaire de reconnaissance en introduisant un grand nombre d'échantillons de référence. Le professeur indique dans ce cas le nom de chaque échantillon. Le choix des caractères de référence est fait à la main en fonction de l'application. Le nombre d'échantillons peut varier de quelques unités à quelques dizaines.
- L'apprentissage non supervisé ou sans professeur consiste à doter le système d'un mécanisme automatique qui s'appuie sur des règles précises de regroupement pour trouver les classes de référence avec une assistance minimale. Dans ce cas les échantillons sont introduits en un grand nombre par l'utilisateur sans indiquer leur classe [36].

I.4.5.2 Reconnaissance et décision

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

Dans ce qui suit, nous présentons une brève revue de littérature de quelques modèles discriminants parmi les plus connus pour la reconnaissance de caractères manuscrits.

I.4.5.3 Décision Bayésienne

Dans une approche par modélisation, on cherche un modèle de production $p(x|c_i)$, qui donne pour chaque classe de formes c_i , la distribution des données qui lui sont associées. Dans une approche par discrimination, on cherche à approximer la distribution $p(c_i|x)$, qui

représente la probabilité à posteriori des données étant donné la classe c_i . En pratique, cette information n'est pas toujours fournie [46].

La règle de décision bayésienne est une théorie clé en classification qui permet d'estimer la probabilité à posteriori à partir de la probabilité conditionnelle et d'émettre un vote d'appartenance de la forme traitée. Elle s'écrit :

$$P(c_i|x) = \frac{P(x|c_i)P(c_i)}{P(x)} \quad (1.1)$$

Pour k classes, la décision bayésienne cherche la classe c_i qui maximise la probabilité à posteriori. Elle s'écrit :

$$c(x) = \arg \max P(c_i|x) \quad (1.2)$$

Le terme $P(x)$ représente la probabilité a priori de la classe c_i . Il est particulièrement utile si les classes ne sont pas balancées dans l'échantillon de données considéré. La vraisemblance de l'observation $P(x)$, est une quantité constante qui peut être omise du processus de décision.

I.4.5.4 Méthode du plus proche voisin (KPPV)

L'algorithme KPPV affecte une forme inconnue à la classe de son plus proche voisin en la comparant aux formes stockées dans une classe de références nommée prototypes. Il renvoie les K formes les plus proches de la forme à reconnaître suivant un critère de similarité. Une stratégie de décision permet d'affecter des valeurs de confiance à chacune des classes en compétition et d'attribuer la classe la plus vraisemblable (selon la métrique choisie) à la forme inconnue [04].

Cette méthode présente l'avantage d'être facile à mettre en œuvre et fournit de bons résultats. Son principal inconvénient est lié à la faible vitesse de classification due au nombre important de distances à calculer.

I.4.5.5 Réseaux de neurones

Les réseaux de neurones ont connu un essor important grâce à l'algorithme de rétropropagation du gradient de l'erreur attribué à Werbos [37], Rumelhart [38] et Lecun [39].

Ce classifieur a trouvé application dans beaucoup de domaines tels que la reconnaissance de caractères, la reconnaissance de visages, la classification d'expressions de gènes, etc. Il est capable d'inférer n'importe quelle fonction de décision non linéaire moyennant une seule couche de neurones cachées et des fonctions d'activation sigmoïdales [40].

En OCR, les primitives extraites sur une image d'un caractère (ou de l'entité choisie) constituent les entrées du réseau. La sortie activée du réseau correspond au caractère reconnu. Le choix de l'architecture du réseau est un compromis entre la complexité des calculs et le taux de reconnaissance [36].

I.4.6 Phase de post traitement

L'objectif du post-traitement est l'amélioration du taux de reconnaissance des mots (par opposition au taux de reconnaissance du caractère). Cette phase est souvent implémentée comme un ensemble d'outils relatifs à la fréquence d'apparition des caractères dans une chaîne, aux lexiques et à d'autres informations contextuelles.

Comme la classification peut aboutir à plusieurs candidats possibles, le post-traitement a pour objet d'opérer une sélection de la solution en utilisant des niveaux d'informations plus élevés (syntaxiques, lexicales, sémantiques...) [04]. Le post-traitement se charge également de vérifier si la réponse est correcte (même si elle est unique) en se basant sur d'autres informations non disponibles au classifieur.

I.5 Conclusion

Nous avons présenté dans ce chapitre le concept de reconnaissance des caractères de façon générale, en mettant le point sur les caractéristiques de l'écriture manuscrite Arabe et les différents aspects d'un OCR ainsi que l'organisation générale d'un système de reconnaissance.

Nous avons vu que la cursivité de l'écriture arabe et les point diacritique montre une complexité de la morphologie des caractères. Face à ce problème, la nécessité d'une méthode d'extraction des caractéristiques plus efficace pour prendre en considération toutes les variations morphologiques de l'écriture arabe.

Pour cette raison il faut choisir une bonne méthode d'extraction de primitive de caractère et une bonne méthode de Classification, Tel est l'objectif du chapitre suivant.

Chapitre II

Les techniques de
reconnaissance de
caractères arabes
manuscrits

II.1 Introduction

Dans cette chapitre, nous présentons une évaluation de l'utilisation de diverses techniques de reconnaissance de manuscrite. Comme les techniques d'extraction de caractéristiques, nous bénéficions de Caractéristiques statistiques le zonage (zoning) et les moments de Zernike. Après la génération de vecteurs de caractéristiques, classificateur Support Vector Machines (SVM) sont utilisés pour l'étape de classification.

II.2 Méthodes d'extraction des caractéristiques

Les méthodes des extractions des caractéristiques représentent le cœur du système de reconnaissance, elle consiste à effectuer le traitement de manuscrit dans un autre espace de travail plus simple et qui assure une meilleure exploitation de données, et donc permettre l'utilisation, seulement, des informations utiles, discriminantes et non redondantes.

Au lieu d'alimenter directement le module de reconnaissance avec l'image du caractère, il est préférable de la paramétrer grâce à l'extraction de certaines de ses caractéristiques.

Cette étape de la reconnaissance consiste à extraire des caractéristiques permettant de décrire de façon non équivoque les formes appartenant à une même classe de caractères tout en les différenciant des autres classes.

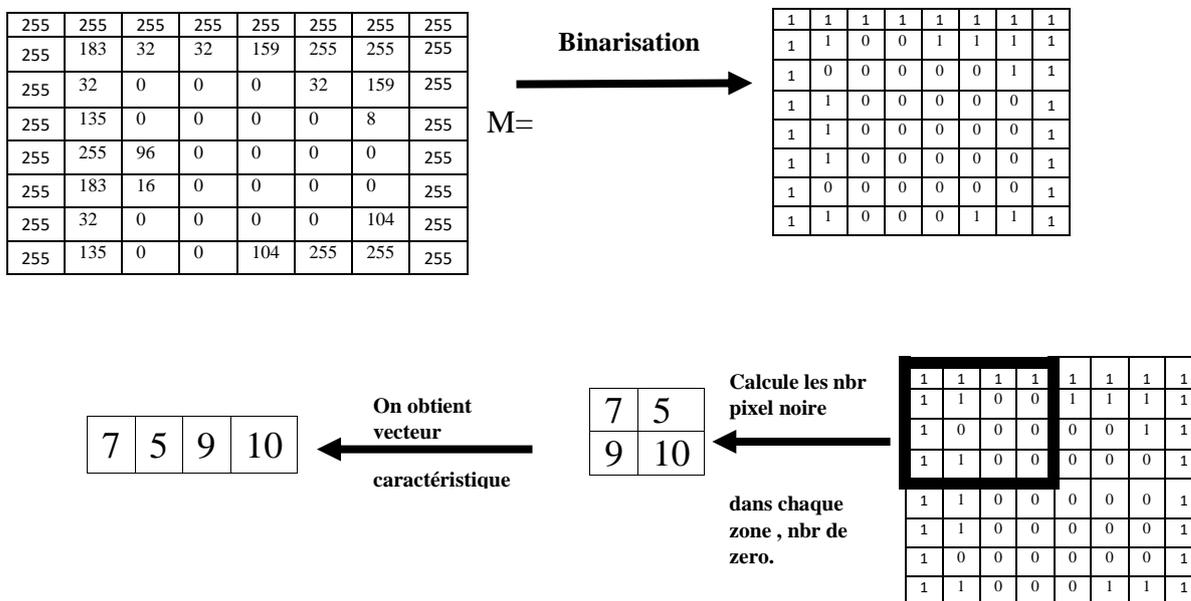
II.2.1 Zoning (densités)

C'est la densité de pixels noirs calculés dans différentes zones (zoning) de l'image de lettre. Pour obtenir ces mesures, on "découpe" horizontalement et verticalement le rectangle englobant le Lattre en zones de taille égale ; Le nombre de pixels noirs dans chaque zone forme alors les composantes du vecteur de caractéristiques. En découpant par exemple l'image en $\ll n \gg$ zones verticales (d'égale largeur) et $\ll m \gg$ zones horizontales (d'égale hauteur), on obtient un vecteur à $(n \times m)$ composantes.(notre cas 5×5) [47]

Les densités dans chaque zone devront être normalisées (en les divisant par exemple par la surface de la zone-moyenne) puisque les lettres ne sont pas tous de même taille.

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

Exemple : soit l'image m (8×8), on le diviser on 4 région (n=2, m=2), Chaque zone est composée de m (4×4).



II.2.2 Les moments de Zernike

Les moments géométriques fournissent un codage compact et facile à calculer, doté de bonnes propriétés d'invariance théorique. En fait, ce codage reste grossier si on conserve peu de termes et on n'a pas de méthodes permettant de déterminer le nombre idéal de moments pour un problème donné. Ce dernier inconvénient est surmonté par les moments de Zernike qui possèdent une propriété d'orthogonalité permettant la reconstruction (en général approchée) de la forme à partir des moments [48].

L'intérêt du calcul des moments de Zernike repose sur l'invariance par translation, par changement d'échelle et par rotation. Ils constituent un espace vectoriel dans lequel l'image de la forme est projetée. Cela permet de s'affranchir du problème de normalisation rencontré dans le cas d'autre moment, mais aussi d'aboutir à une description plus précise des formes, par utilisation d'ordres plus élevés, en maintenant les propriétés d'invariance.

Les polynômes de Zernike furent proposés en 1934 par Zernike. Dérivés de ces polynômes, les moments ont été utilisés par de nombreux auteurs en reconnaissance de caractères.

Plusieurs études montrent également la supériorité de ces descriptions par rapport à d'autres approches [49], car elles réduisent le bruit, les redondances et peuvent être reconstruite.

II.2.2.1 Formule des moments de Zernike

Les moments Zernike A_{nm} sont définis comme des polynômes complexes qui forment un ensemble orthogonal complet du disque (cercle) unité $(x^2 + y^2) \leq 1$. Ils correspondent à la projection de la forme $f(x, y)$ sur une base ZP de fonctions orthogonales $V_{nm}(x, y)$ et se défini par [49] :

$$ZP = \{V_{nm}(x, y) \mid (x^2 + y^2) \leq 1\} \tag{2.1}$$

Où le polynôme complexe V d'ordre n et de répétition m est défini tel que $n - |m|$ soit pair et $|m| \leq n$

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho)e^{jm\theta} \tag{2.2}$$

n : nombre entier positif ou nul.

m : nombre entier positif ou négatif, tel que $n - |m|$ pair et $|m| \leq n$.

ρ : longueur du vecteur à partir de l'origine au pixel (x, y) .

θ : angle entre le vecteur ρ et l'axe x dans le sens inverse de l'aiguille d'une montre.

$$V_{nm}(x, y) = R_{nm}(x, y)e^{jm \arctan(y/x)} \tag{2.3}$$

Le rapport polynomial est défini comme suit [49] :

$$R_{nm}(x, y) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} (x^2 + y^2)^{(n-2s)/2} \tag{2.4}$$

Soit $\rho=(x^2+y^2)^{1/2}$ la longueur du vecteur à partir de l'origine du pixel (x,y) et $\theta=\arctan(x/y)$ l'angle entre l'axe x et ce vecteur. $R_{nm}(\rho)$, la représentation en coordonnées polaires ($x=\rho \cos\theta, y=\rho \sin\theta$) de $R_{nm}(x,y)$ est un polynôme de degré n [49] :

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s} \tag{2.5}$$

Les moments de Zernike sont les projections de la fonction d'image f(x,y) dans la base orthogonale de fonction $V_{nm}(x,y)$. Le moment Zernike, d'ordre n et de répétition m, est un nombre complexe défini par :

$$A_{nm} = \frac{n+1}{\pi} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) [V_{nm}(x,y)]^* \tag{2.6}$$

où $x^2+y^2 \leq 1$ et * définit le complexe conjugué L'ordre des moments possède de grande influence sur la conservation de l'information angulaire. Plus l'ordre est élevé et plus les variations angulaires décrites sont fines. La Figure 2.1 en donne une illustration.

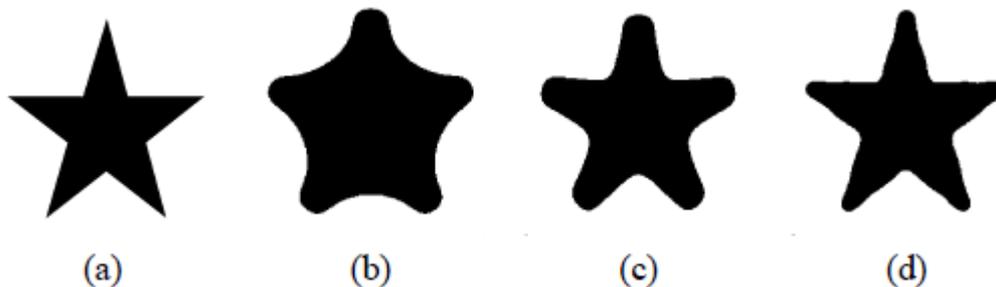


Figure 2.1 Exemple de reconstructions à partir des descripteurs de Zernike, (a) image d'origine, (b) reconstruction d'ordre 10, (c) reconstruction d'ordre 20, (d) reconstruction d'ordre 40 [32].

Les moments Zernike sont bien connus pour être invariants à la rotation. Une normalisation d'image est nécessaire pour rendre ces moments invariants à la translation et au facteur d'échelle [10].

II.3 Méthodes de classification

Les méthodes de classification ont pour but d'identifier les classes auxquelles appartiennent des objets à partir de certains paramètres descriptifs. Elles s'appliquent à un grand nombre d'activités humaines et conviennent en particulier au problème de la prise de décision automatisée. La procédure de classification sera extraite automatiquement à partir d'un ensemble d'exemples. Un exemple consiste en la description d'un cas avec la classification correspondante. Un système d'apprentissage doit alors, à partir de cet ensemble d'exemples, extraire une procédure de classification, il s'agit en effet d'extraire une règle générale à partir des données observées. La procédure générée devra classer correctement les exemples de l'échantillon et avoir un bon pouvoir prédictif pour classer correctement de nouvelles descriptions.

Les méthodes utilisées pour la classification sont nombreuses, citons : la méthode de Réseaux de Neurones, Séparateurs à Vastes Marges(SVM), la logique floue, ...etc.

Nous présentons dans la suite une étude détaillée sur la technique SVM. Cette méthode a montré leur efficacité dans de nombreux domaines d'applications tels que le traitement d'image, la catégorisation de textes, reconnaissance de caractère et le diagnostic médicale.

II.3.1 Machines à vecteurs supports (SVM)

Les méthodes à noyaux sont inspirées de la théorie de l'apprentissage statistique que Vladimir Vapnik. Les Machines à Vecteurs de Supports (SVM) [50] [51] [52] sont une famille des méthodes à noyaux, qui permettent de résoudre efficacement des problèmes de classification ou de régression.

Cette technique a connu un grand succès au cours des dernières années. Des outils comme la librairie LIBSVM [53] permettent de tester facilement un système à base de SVM, et ont contribué à son essor.

II.3.2 Historique

L'origine des machines à vecteurs de support (SVM) remonte à 1975 lorsque Vapnik et Chervonenkis proposèrent le principe du risque structurel et la dimension VC pour caractériser la capacité d'une machine d'apprentissage. A cette époque, ce principe n'a pas trouvé place et il n'existait pas encore un modèle de classification solidement appréhendé

pour être utilisable. Il a fallu attendre jusqu'à l'an 1982 pour que Vapnik propose un premier classificateur basé sur la minimisation du risque structurel baptisé SVM [50].

Ce modèle était toutefois linéaire et l'on ne connaissait pas encore le moyen d'induire des frontières de décision non linéaires. En 1992, Boser et al. proposent d'introduire des noyaux non-linéaires pour étendre le SVM au cas non-linéaire [54]. En 1995, Cortes et al. proposent une version régularisée du SVM qui tolère les erreurs d'apprentissage avec pénalités [55] [46]. Depuis, les SVMs (le pluriel est utilisé pour désigner les différentes variantes du SVM) n'ont cessé de susciter l'intérêt de plusieurs communautés de chercheurs de différents domaines d'expertise.

II.3.3 Application des SVM à la classification binaire

II.3.3.1 Définition

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

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

II.3.3.2 Notion d'apprentissage

L'apprentissage c'est l'acquisition de connaissances et compétences permettant la synthèse d'information [56]. Un algorithme d'apprentissage va permettre de passer d'un espace des exemples X à un espace dit des hypothèses H . L'algorithme SVM va explorer l'espace H pour obtenir le meilleur hyperplan séparateur. L'apprentissage permet, à partir d'un ensemble de paramètres en entrée, d'obtenir un ensemble de résultats en sortie. On va donc préparer un jeu de données à apprendre, constitué de couples paramètres/résultats. Le but recherché est d'apprendre une fonction permettant de prédire de nouveaux résultats pour de nouvelles entrées.

L'apprentissage est dit supervisé ou non supervisé. Les SVM se situent dans le groupe des algorithmes d'apprentissage supervisés puisque que l'on utilise une base d'exemples pour obtenir la règle de classification.

II.3.3.3 Principe de fonctionnement générale

Pour deux classes d'exemples données, le but des SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est linéaire appelé « hyperplan » [57]. Dans le schéma qui suit, on détermine un hyperplan qui sépare deux ensembles de points.

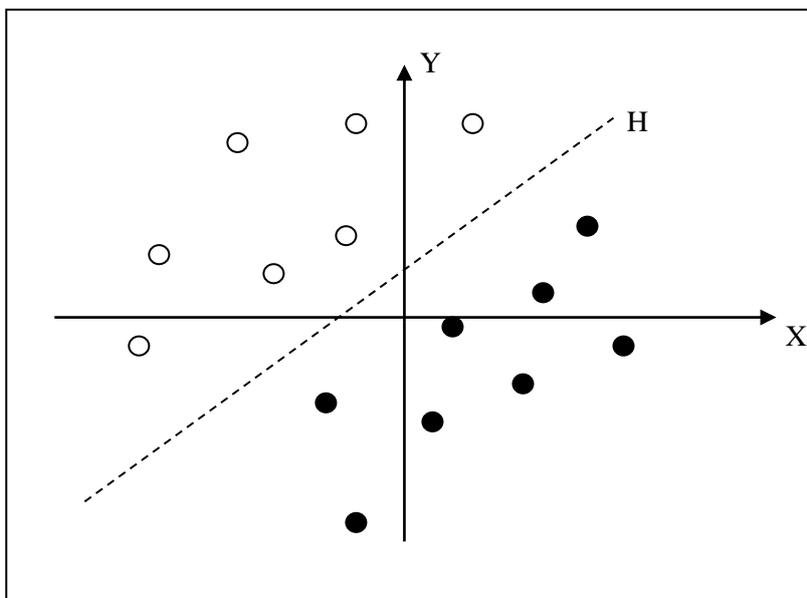


Figure 2.2 Séparation de deux ensembles de points par un Hyperplan H

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

Il est évident qu'il existe une multitude d'hyperplan valides mais la propriété remarquable des SVM est que hyperplan doit être optimal. Nous allons donc en plus chercher parmi les hyperplans valides, celui qui passe « au milieu » des points des deux classes d'exemples.

Cela revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale. On appelle cette distance « marge » entre l'hyperplan et les exemples.

L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge [58].

Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points.

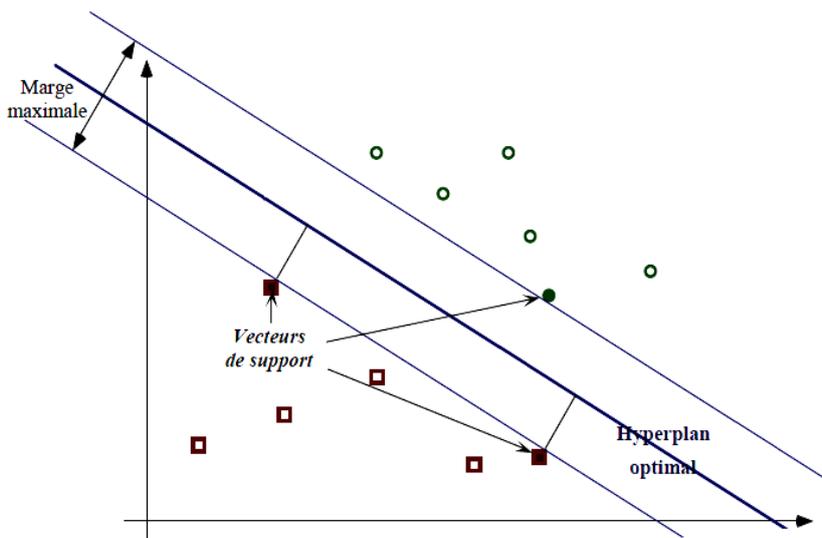


Figure 2.3 Hyperplan optimal, marge et vecteurs de support

Intuitivement, le fait d’avoir une marge plus large procure plus de sécurité lorsque l’on classe un nouvel exemple. De plus, si l’on trouve le classificateur qui se comporte le mieux vis-à-vis des données d’apprentissage, il est clair qu’il sera aussi celui qui permettra au mieux de classer les nouveaux exemples.

Dans le schéma qui suit, la partie droite nous montre qu’avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu’il tombe dans la marge. On constate sur la partie gauche qu’avec une plus petite marge, l’exemple se voit mal classé. [60], [57], [52]. La classification d’un nouvel exemple inconnu est donnée par sa position par rapport à l’hyperplan optimal. [57], [59]

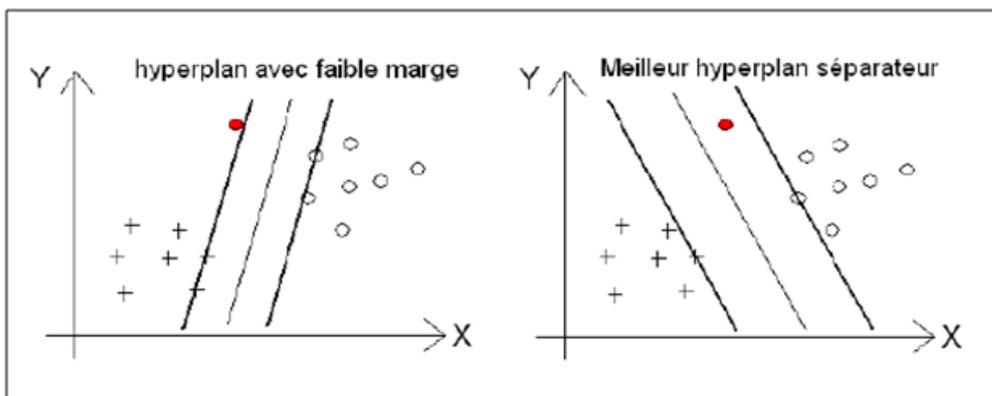


Figure 2.4 Maximation de la marge

II.3.4 Linéarité et non linéarité

Dans les SVM, on constate deux cas de séparabilité :

II.3.4.1 Cas linéairement séparable

Dans un problème linéairement séparable les SVM trouvent facilement un séparateur (classificateur linéaire) qui maximise la marge.

II.3.4.2 Cas non linéairement séparable

Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables [61] (Figure2.5).

Le principe consiste à projeter les données de l'espace d'entrée (appartenant à deux classes différentes) non linéairement séparables dans un espace de plus grande dimension appelé «espace de caractéristiques ou espace de redescription» de façon à ce que les données deviennent linéairement séparables.

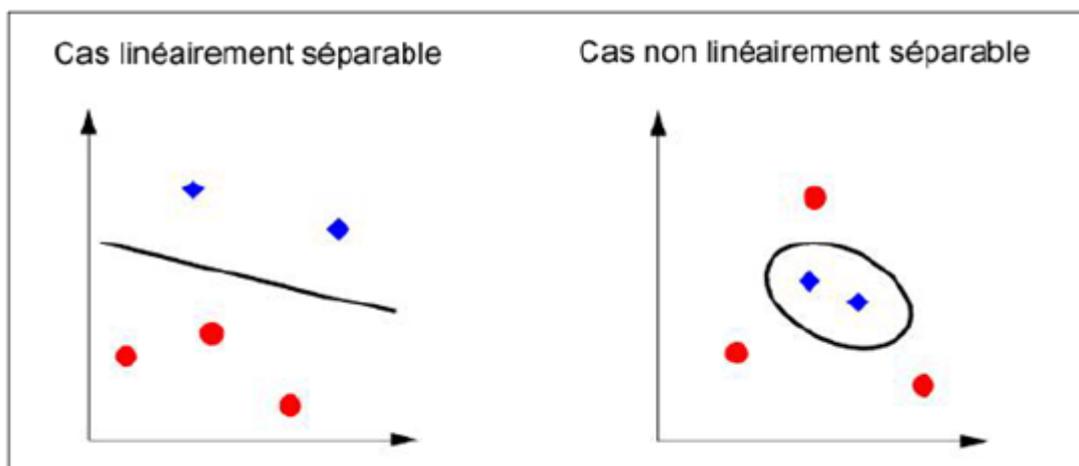


Figure 2.5 linéarité et non linéarité

On a donc une transformation d'un problème de séparation non linéaire dans un espace de représentation en un problème de séparation linéaire dans un espace de redescription de plus grande dimension. Cette transformation est réalisée via une fonction noyau (Figure 2.6).

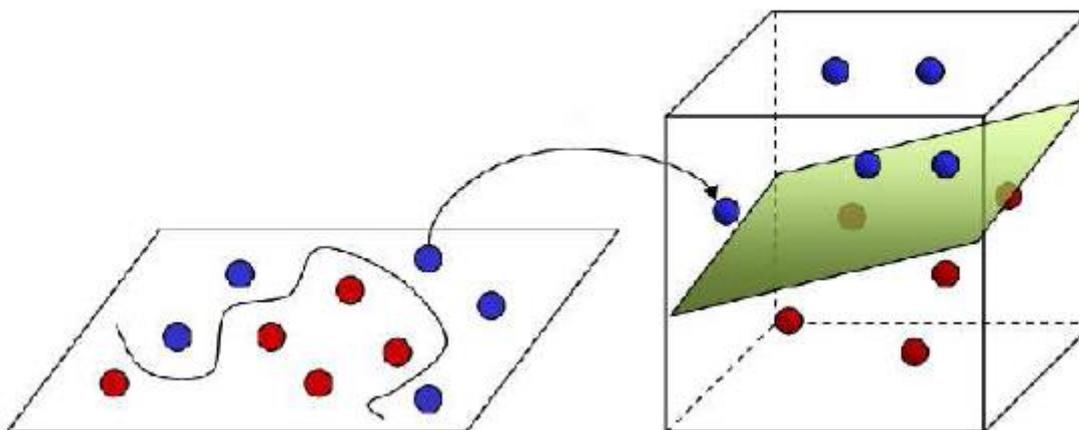


Figure 2.6 Transformation de l'espace de représentation et l'hyperplan séparateur dans le cas non linéairement séparables.

II.3.5 Fondements mathématiques

II.3.5.1 Principe

Les machines à vecteurs support forment une classe d'algorithmes d'apprentissage supervisés. Nous nous intéressons à une fonction notée f qui à toute entrée x fait correspondre une sortie $y = f(x)$. Le but est d'essayer d'apprendre f à partir d'un ensemble de couple (x_i, y_i) . Dans ce problème les machines à vecteurs support vont être utilisées pour classifier une nouvelle observation x en se limitant à deux classes $y \in \{-1, 1\}$.

Nous allons donc construire une fonction f qui à chaque valeur d'entrée dans un ensemble R^d va faire correspondre une valeur de sortie $y \in \{-1, 1\}$:

$$f : R^d \rightarrow \{-1, 1\}, \quad f(x) = y$$

Dans le cas linéaire, une fonction discriminante h est obtenue par combinaison linéaire d'un vecteur d'entrée $x = (x_1, \dots, x_d)$ et s'écrit :

$$h(x) = w \cdot x + b \tag{2.5}$$

La classe est donnée par le signe de $h(x)$: $f(x) = \text{sign}(h(x))$. Si $h(x) \geq 0$ alors x est de classe 1 sinon x est de classe -1. Le séparateur est alors un hyperplan d'équation : $w \cdot x + b = 0$.

Si (x_i, y_i) est un des p éléments de la base d'apprentissage notée A_p , on veut trouver le classifieur h tel que :

$$y_i(w \cdot x_i + b) \geq 0, i \in [1, p] \quad (2.6)$$

Dans le cas simple linéairement séparable il existe de nombreux hyperplans séparateurs. Selon la théorie de Vapnik [62], l'hyperplan optimal est celui qui maximise la marge. Cette dernière étant définie comme la distance entre un hyperplan et les points échantillons les plus proches. Ces points particuliers sont les vecteurs supports. La distance entre un point x quelconque et l'hyperplan est donnée par l'équation 2.7.

$$d(x) = \frac{w \cdot x + b}{\|w\|} \quad (2.7)$$

Donc maximiser la marge va revenir à minimiser $\|w\|$.

II.3.5.2 Forme primale

Les paramètres w et b étant définis à un coefficient multiplicatif près, on choisit de les normaliser pour que les échantillons les plus proches (x_s) vérifient l'égalité suivante :

$$y_s(w \cdot x_s + b) = 1$$

Donc quelque soit l'échantillon x_i on obtient :

$$y_i(w \cdot x_i + b) \geq 1 \quad (2.8)$$

La distance entre l'hyperplan et un point support est donc définie par $\frac{1}{\|w\|}$. La marge géométrique entre deux classes est égale à $\frac{2}{\|w\|}$. La forme primale (qui dépend seulement de w et b) des SVM est donc un problème de minimisation sous contrainte qui s'écrit :

$$\begin{cases} \min \left(\frac{1}{2} \|w\|^2 \right) \\ \forall (x_i, y_i) \in A_p, y_i(w \cdot x_i + b) \geq 1 \end{cases} \quad (2.9)$$

II.3.5.3 Forme duale

La formulation primale peut être transformée en formulation duale en utilisant les multiplicateurs de Lagrange. L'équation 2.9 s'écrit alors sous la forme suivante :

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^P \alpha_i (y_i (w \cdot x_i + b) - 1) \quad (2.10)$$

La formulation de Lagrange permet de trouver les extremums en annulant les dérivées partielles de la fonction $L(w, b, \alpha)$. Le lagrangien L doit être minimisé par rapport à w et b et maximisé par rapport à α . On résout ce nouveau problème en calculant les dérivées partielles :

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^P \alpha_i y_i x_i = 0 \quad (2.11)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^P \alpha_i y_i = 0 \quad (2.12)$$

En réinjectant les deux premières dérivées partielles 2.11 et 2.12 dans l'équation 2.10 nous obtenons :

$$L(\alpha) = \frac{1}{2} \sum_{i=1}^P \alpha_i y_i \sum_{j=1}^P \alpha_j y_j x_i \cdot x_j - \sum_{i=1}^P \alpha_i y_i \sum_{j=1}^P \alpha_j y_j x_i \cdot x_j - \sum_{i=1}^P \alpha_i y_i b + \sum_{i=1}^P \alpha_i$$

On en extrait la formulation duale (dépendant des α_i) suivante :

$$L(\alpha) = \sum_{i=1}^P \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (2.13)$$

On cherche donc à maximiser $L(\alpha)$ sous les contraintes $\alpha_i \geq 0$ et $\sum_i \alpha_i y_i = 0$. A l'optimal α^* les conditions de Karush Kuhn Tucker (conditions KKT) sont satisfaites et permettent d'écrire l'égalité suivante :

$$\alpha_i [y_i (w \cdot x_i + b) - 1] = 0, \forall i \in [1, P] \quad (2.14)$$

Cela nous donne $\alpha_i = 0$ ou $(w \cdot x_i + b) - 1 = 0$. Ces deux possibilités impliquent que seuls les α_i associés à des exemples situés sur la marge peuvent être non nuls. Autrement dit, ces exemples sur la marge constituent les vecteurs supports, qui seuls contribuent à définir l'hyperplan optimal.

Cette maximisation est un problème de programmation quadratique de dimension égale au nombre d'exemple. L'équation 2.11 nous donne la valeur optimale pour w noté $w^* : w = \sum_{i=1}^P \alpha_i^* y_i x_i$, avec α_i^* les coefficients de Lagrange optimaux. En utilisant l'équation de l'hyperplan 2.5 nous obtenons l'hyperplan de marge maximale :

$$h(x) = \sum_{i=1}^P \alpha_i^* y_i x \cdot x_i + b \quad (2.15)$$

II.3.6 Principe du noyau

Le cas linéairement séparable est peu intéressant, car les problèmes de classification sont souvent non linéaires. Pour le résoudre ce problème il suffit de projeter les données dans un espace de dimension supérieur appelé espace de redescription noté H. L'idée étant qu'en augmentant la dimensionnalité du problème on se retrouve dans le cas linéaire vu précédemment. Nous allons donc appliquer une transformation non linéaire $\Phi(\cdot)$ aux vecteurs d'entrée x_i tel que $x_i \in \mathbb{R}^d$ et $\Phi(x_i) \in \mathbb{R}^e$, ($e > d$). Ce changement va conduire à passer d'un produit scalaire dans l'espace d'origine $x_i \cdot x_j$ à un produit scalaire $\Phi(x_i) \cdot \Phi(x_j)$ dans l'espace de redescription (voir la figure 2.7).

Le principe est d'utiliser une fonction noyau notée K qui évite le calcul explicite du produit scalaire dans l'espace de redescription.

Nous avons alors l'égalité suivante :

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$$

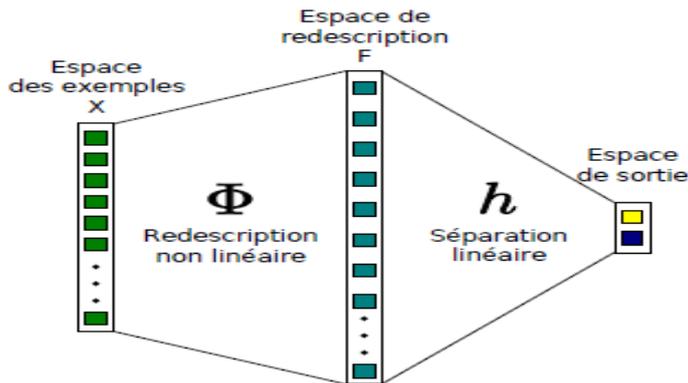


Figure 2.7 Transformation linéaire des données en une séparation linéaire dans un nouvel espace.

II.3.6.1 Exemple

Prenons le cas trivial où $x = (x_1, x_2)$ dans R^2 et $\Phi(x) = (x_1^2, \sqrt{2 x_1 x_2}, x_2^2)$ est explicite. Dans ce cas, H est de dimension 3 et le produit scalaire s’écrit :

$$\begin{aligned} \langle \Phi(x), \Phi(x') \rangle &= x_1^2 x_1'^2 + 2 x_1 x_2 x_1' x_2' + x_2^2 x_2'^2 \\ &= (x_1 x_1' + x_2 x_2')^2 \\ &= \langle x, x' \rangle^2 = k(x, x') \end{aligned}$$

Le calcul du produit scalaire dans H ne nécessite pas l’évaluation explicite de Φ . La fonction noyaux doit satisfaire la condition de Mercer présenté ci-dessous.

II.3.6.2 Condition de Mercer

Une fonction $k(.,.)$ symétrique est un noyau si, pour tous les x_i possibles, la matrice de terme général $k(x_i, x_j)$ est une matrice définie positive c’est-à-dire quelle définit une matrice de produit scalaire [36]. Dans ce cas, on montre qu’il existe un espace H et une fonction Φ tels que :

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

Malheureusement, cette condition théorique d’existence est difficile à vérifier et, de plus, elle ne donne aucune indication sur la construction de la fonction noyau ni sur la transformation Φ . La pratique consiste à combiner des noyaux simples pour en obtenir des plus complexes (multidimensionnels) associés à la situation rencontrée [36].

II.3.6.3 Exemples de noyaux

- Linéaire : $K(x, x') = x \cdot x'$
- Polynomial : $K(x, x') = (x \cdot x')^d$ ou $(c + x \cdot x')^d$
- Gaussien : $K(x, x') = \exp^{-\|x-x'\|^2/\sigma}$
- Laplacien : $K(x, x') = \exp^{-\|x-x'\|/\sigma}$
- Sigmoidale : $K(x, x') = \tanh(\alpha_0(x \cdot x') + \beta_0)$,

Les noyaux gaussiens sont des noyaux dits de type radial (fonction à base radiale abrégée RBF [63]), indiquant qu'ils dépendent de la distance entre deux exemples. L'hyperplan séparateur se réécrit avec la fonction noyau sous la forme suivante :

$$h(x) = \sum_{i=1}^P \alpha_i^* y_i k(x, x_i) + b \quad (2.16)$$

Beaucoup d'articles sont consacrés à la construction d'un noyau plus ou moins exotique et adapté à une problématique posée : reconnaissance de séquences, de caractères, l'analyse de textes... La grande flexibilité dans la définition des noyaux, permettant de définir une notion adaptée de similitude, confère beaucoup d'efficacité à cette approche à condition bien sûr de construire et tester le noyau le plus approprié.

II.3.7 Marges souples

Les machines à vecteurs support sont efficaces quand le problème est séparable. Nous avons vu que l'utilisation d'une méthode noyau permettait de traiter les cas non linéaires mais cela n'est pas utilisable dans le cas de données non séparables par exemple pour des données bruitées. En effet, on peut avoir des éléments à classer du mauvais côté de l'hyperplan comme le montre la Figure 2.7. Cortes et Vapnik en 1995 ont donc introduit le concept de marge souple [47]. Certains exemples d'apprentissage peuvent violer la contrainte 2.8 que l'on retrouve dans l'équation de la forme primale (2.9).

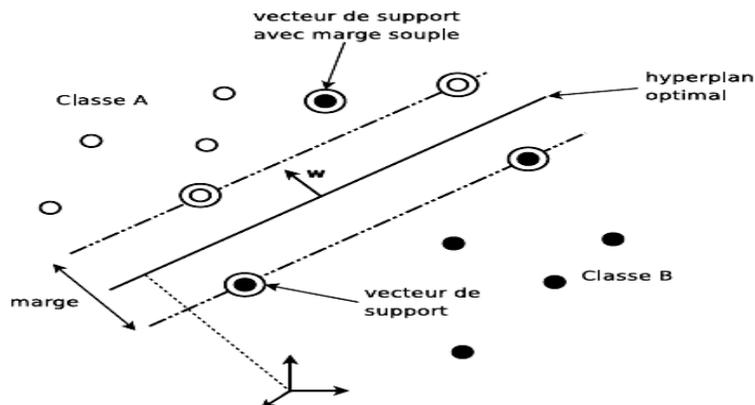


Figure 2.8 Hyperplan de séparation optimale avec marge souple dans un cas non linéairement séparable.

On introduit par conséquent des variables dites « ressorts » $\xi = \xi_1, \dots, \xi_P$ qui permettent d’assouplir la contrainte pour chaque exemple. Un paramètre supplémentaire de régularisation C est ajouté pour contrôler la pénalité associée aux exemples. La nouvelle forme primale décrite en 2.9 devient alors :

$$\begin{cases} \min(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^P \xi_i) \\ \forall (x_i, y_i) \in A_P, \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i \end{cases} \tag{2.17}$$

De même que pour la forme primale nous obtenons une nouvelle formulation duale qui est alors similaire à celle décrite précédemment. Si on utilise en plus la fonction noyau K dans la formulation duale 2.13 en appliquant la méthode des multiplicateurs de Lagrange on cherche alors à maximiser la nouvelle fonction $L(\alpha)$.

$$L(w, b, \xi, \alpha) = \frac{1}{2} w^2 + C \sum_{i=1}^P \xi_i - \sum_{i=1}^P \alpha_i [y_i(w \cdot x_i + b) - 1 + \xi_i]$$

En appliquant la même méthode on obtient $L(\alpha)$ à partir de l’expression du Lagrangien précédent.

$$L(\alpha) = \sum_{i=1}^P \alpha_i - \frac{1}{2} \sum_{i,j=1}^P \alpha_i \alpha_j y_i y_j K(x_i x_j) \tag{2.18}$$

$$\forall (x_i, y_i) \in A_P, 0 \leq \alpha_i \leq C \text{ et } \sum_i y_i \alpha_i = 0$$

Le seul changement est la contrainte supplémentaire sur les coefficients α_i , qui se traduit par une borne supérieure C . La solution de l'équation précédente 2.18 est de la forme :

$$h(x) = \sum_{i=1}^P \alpha_i^* y_i K(x, x_i) + b \quad (2.19)$$

Remarque :

Le calcul des α_i^* du Lagrangien vu précédemment est obtenu par programmation quadratique. C'est un problème bien connu pour lequel il existe de nombreux algorithmes d'optimisation. Ce type de problème a un optimum global unique, élément important pour éviter d'atteindre un optimum local [64]. Il s'ensuit un seul α^* qui maximise l'équation (2.18).

Une fois que l'on a obtenu les coefficients α^* , le coefficient b peut être obtenu en utilisant n'importe quel vecteur x_i avec l'égalité (2.14).

Des paragraphes précédents, on soulignera que la solution dépend uniquement d'un nombre restreint d'exemple de l'ensemble d'apprentissage et non pas de la dimension d de l'espace des exemples.

II.3.8 SVM multi-classes

A l'origine, les SVM ont été conçus essentiellement pour les problèmes à deux classes cependant plusieurs approches permettant d'étendre cet algorithme aux cas à N classes ont été proposées. Pour pouvoir traiter plus de deux classes, il convient d'apporter les modifications nécessaires.

Nous cherchons à modéliser une fonction $G : \Omega \rightarrow 1, \dots, m$, qui définit m partitions dans l'espace des caractéristiques. Connaissant G , les partitions des classes sont définies par $G^{-1}(k)$, où $k \in [1..m]$.

Pour un problème à deux classes, l'hyper-plan (w, b) du SVM délimite les deux partitions selon $signf(x)$ où $f(x) = w \cdot x + b$. Par ailleurs, bien que le SVM soit un classifieur binaire, il peut facilement être étendu pour décider de l'appartenance de données multi-classes. En particulier, on trouve deux schémas de classification :

II.3.8.1 Approche Un-contre-tous

L'approche la plus naturelle est d'utiliser cette méthode de discrimination binaire et d'apprendre N fonctions de décision $\{f_m\}$ $m = 1 \dots N$ permettant de faire la discrimination entre chaque classe de toutes les autres (chaque classe est opposé les autres),il faut donc poser N problèmes binaires. Le k ième classificateur sépare les données de la classe k de tout le reste des données d'apprentissage.

II.3.8.2 Approche Une-contre-Une

Une-contre-Une. Cette méthode requiert l'apprentissage de $\frac{1}{2} m (m - 1)$ classifieurs pour tous les couples de classes possibles. Durant le test, la méthode requiert la combinaison de toutes les sorties de classifieurs pour qu'une décision soit émise.

II.3.8.3 Phase de décision

II.3.8.3.1 votant

La fonctions de décisions (classifieurs) sont apprises et chacune d'entre elles effectue un vote pour L'affectation d'un nouveau point x . La classe de ce point x à le plus grand nombre de votes devient ensuite la classe majoritaire [65], [59].

II.3.8.3.2 Arbre de décision

Dans un arbre de décision, chaque nœud de l'arbre fait la décision si une certaine donnée appartient à un certain ensemble de classe. La succession des nœuds fait réduire le nombre de classes jusqu'à ce qu'on arrive à une seule classe [68] La figure (2.9) montre un exemple de graphe acyclique direct, qui a été utilisé avec succès dans les SVM.

Chaque noeud de l'arbre contient un classificateur destiné à distinguer entre deux classes seulement parmi les quatre. A chaque niveau, on élimine une classe jusqu'à ce qu'une classe reste.

Le choix d'une stratégie se fait par rapport au problème traité car il n'y a aucune étude comparative permettant de favoriser l'une par rapport à l'autre. La troisième méthode construit le minimum de classificateur SVM, mais elle n'est pas évidente à mettre en oeuvre car pour certains problèmes, il est difficile de trouver les dichotomies. Si on considère l'effet de l'apprentissage, la première stratégie est préférable puisqu'elle ne nécessite que le calcul de k SVM binaires.

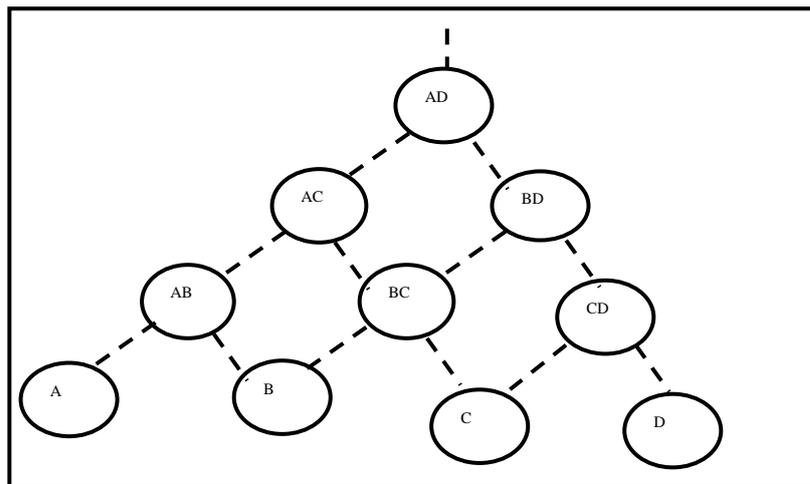


Figure 2.9 Graphe acyclique direct pour la classification de quatre classes en utilisant des classificateurs SVM binaires.

II.3.9 Avantages et limites

II.3.9.1 les avantages

- Les SVM possèdent des fondements mathématiques solides.
- Les exemples de test sont comparés juste avec les supports vecteur et non pas avec tout les exemples d'apprentissage.
- Décision rapide. La classification d'un nouvel exemple consiste à voir le signe de la fonction de décision $f(x)$.

II.3.9.2 Les limites

- Classification binaire d'où la nécessité d'utiliser l'approche un-contre-un.
- Grande quantité d'exemples en entrées implique un calcul matriciel important.
- Temps de calcul élevé lors d'une régularisation des paramètres de la fonction noyau.
- Elles ne servent pas à calculer des probabilités d'appartenance à la classe [69].
- L'utilisation des fonctions noyaux ne permet pas de sélectionner des caractéristiques importantes pour la classification [69].

II.4 Conclusion

Dans ce chapitre, nous avons vu deux concepts des méthodes d'extraction des caractéristiques utilisées dans la reconnaissance de caractère arabe manuscrit ainsi que le concept supervisé avec SVM ont présentées.

La méthode de SVM, a été détaillée en commençant par la classification binaire de classes dans le cas de données linéairement séparables et comment trouver un hyperplan optimal qui les sépare. Le cas du non séparabilité linéaire des données est évoqué, ainsi que le moyen de contourner la problématique en mappant les données d'apprentissage avec une fonction noyau dans un espace de dimension supérieur, afin qu'elles deviennent linéairement séparables.

L'atout de SVM, est que l'hyperplan optimal n'est caractérisé que par un certain nombre de points appelés les vecteurs de supports, parmi tous les points d'apprentissage, ce qui lui permet de classifier un grand nombre de données. La base théorique solide, sur la qu'elle repose SVM, est un autre point fort, ce qui fait d'elle, la méthode la plus utilisée pour les taches de classification et prédiction.

Comme toutes les méthodes d'apprentissage, SVM nécessite un temps de calcul, qui peut croire en fonction de la taille des données et du noyau utilisé. Néanmoins le temps d'apprentissage reste minime par rapport à d'autres méthodes telles que : les réseaux de neurones.

Chapitre III

Implémentation et
résultats

III.1 Introduction

Pour valider l'étude théorique de notre système il est nécessaire de passer maintenant à la partie pratique, et donc à l'implémentation d'un système de reconnaissance

Nous présentons dans ce chapitre une description détaillée de notre système de reconnaissance de caractères Arabe manuscrits hors-ligne, avec l'évaluation de performance de chaque phase de processus de reconnaissance. Ainsi que nous présentons les différents résultats d'expérimentations.

III.2 Ressources matérielles et logicielles

III.2.1 Ressources matérielles

Nous avons utilisé dans ce travail un PC portable (Intel Core I3-4005U CPU 1.70 Ghz, avec 4 Go de RAM), utilisé pour réaliser notre système de reconnaissance. et fait un test sur notre système.

III.2.2 Ressources logicielles

Environnement de développement

On a utilisé Windows 10 professionnel 64 bits comme système d'exploitation.

Nous avons développé notre système à l'aide de **Matlab** Qui est un langage de programmation de quatrième génération émulé par un environnement de développement du même nom, il est utilisé à des fins de calcul numérique. Développé par la société The MathWorks, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs [72]. Destiné aux ingénieurs, aux techniciens et aux scientifiques, c'est un outil très utilisé, dans les universités comme dans le monde industriel, qui intègre des centaines de fonctions mathématiques et d'analyse numérique (calcul matriciel le MAT de Matlab, traitement de signal, traitement d'images, visualisations graphiques, etc.) [27].

III.3 Description de notre système de reconnaissance

Notre système passe par deux étapes :

III.3.1 Création des séparateurs (on ligne)

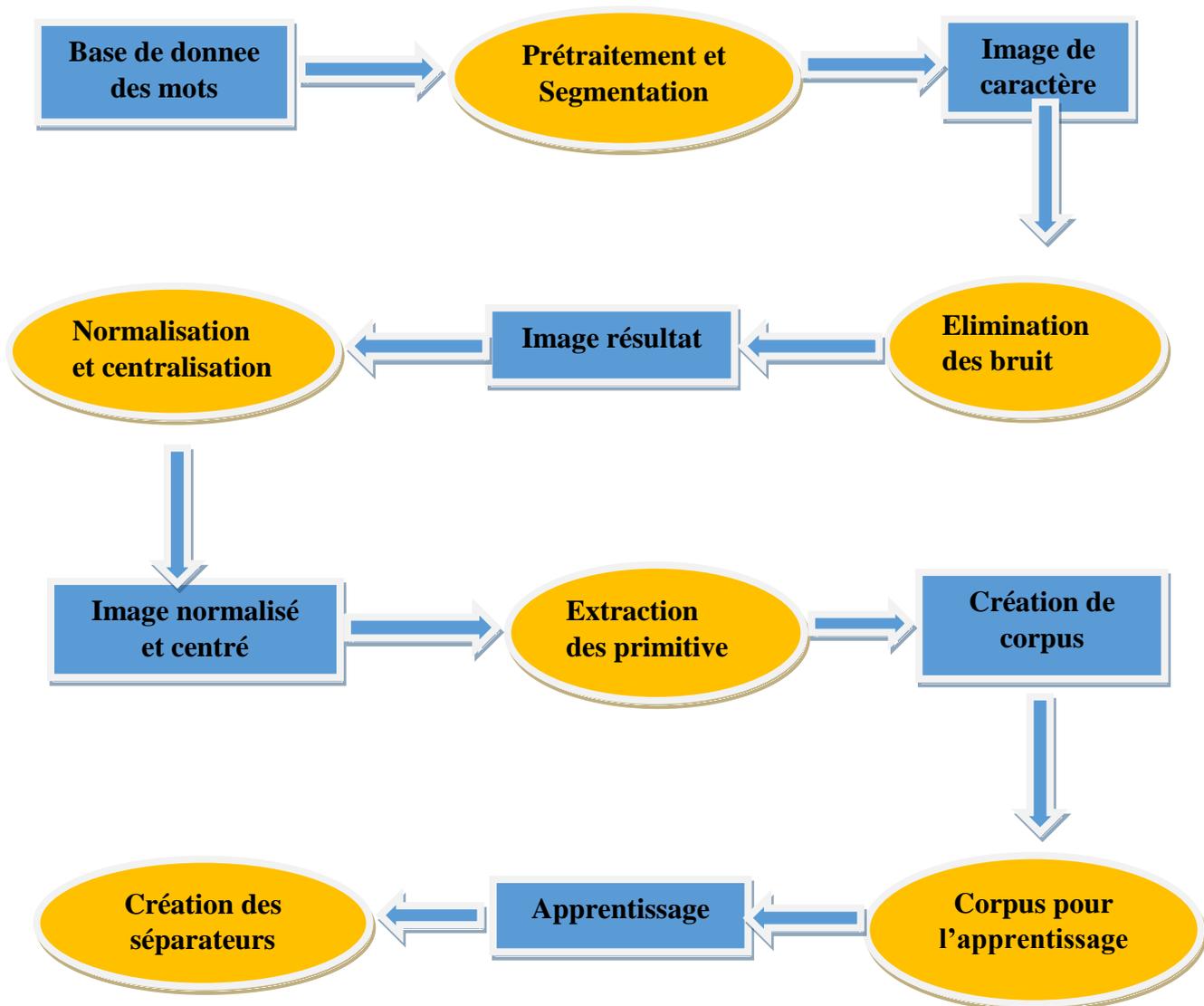


Figure 3.1 Schéma général de notre système de reconnaissance.

III.3.1.1 Description de base de données utilisées

M. Pechwitz et al, introduisent la base IFN/ENIT en 2002 .Il s'agit d'une base de données d'images de noms de villes tunisiennes. Outre la séquence de lettres, sont également annotées la forme que prend chacune des lettres au sein du mot, la présence des signes diacritiques secondaires, et une approximation de la ligne de base [70].

III.3.1.2 Prétraitement et Segmentation

Dans cette étape nous somme appuyer sur des résultats précédent [71]

Vous utilisons deux types de filtrages:"sobel" et "médian".

III.3.1.2.1 Algorithme de segmentation

Algorithme de segmentation:

```

Sélectionner une image ;
A=Filtrer par médian mayaneur de l'image ;
B=Filtrer par " sobel"de( A) ;
B=Filter par "médian moyaneur" de (B) ;
V_h=Calculer l'histogramme verticale de (B) ;
Calculer le seuil ;
Possible de segment=0 ;
j=1 ;
Pour i de 1 à taille de (V_h) faire
    Si (V_h <seuil) et (possible de segment=1) alors
        Région (j, 2)=i ;
        Possible de segment=0 ;
        j=j+1 ;
    Sinon Si (V_h(i)>=seuil) et(possible de segment=0) alors
        Region(j,1)=i ;
        Possible de segment=1 ;
    Finsi.
Finsi.
Fin pour.
Pour jj de 1 à taille de region faire
    C=A(region(1,1),region(1,2)) ;
    C=filtre par "sobel"de (C) ;
    C=filtre par "median moyaneur" de(C) ;
    H_h=calculer l'histogramme horizontale ;
    Clculer la seuil2 ;
    Pour i de 1 à taille de (H_h) faire
        Si ((H_h)<seuil2) alors
            Si (H_h(i+1)>seuil2)alors ségmenter ;
        Finsi.
    Finsi.
Fin pour.
Fin pour.

```

III.3.1.2.2 les résultats de segmentation

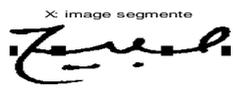
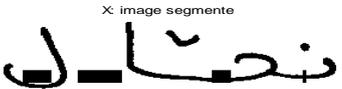
Pourcentage	Nombres des images	Exemple	
		Code image	Image
Inferieur à 25%	45 (7.90%)	ae19_025	
Entre 25 et 50%	115 (20.21%)	ai14_029	
Entre 50 et 75%	246 (43.23%)	aq34_055	
Entre 75 et 100%	163 (28.54%)	ae70_040	

Tableau 3.1: Les résultats de segmentation.

III.3.1.2.3 Les problèmes de segmentation

- Ligatures verticales connectées ou non.



Figure 3.2: Exemple de ligatures verticales connectées ou non.

- ☒ Liaison indésirable entre caractères.

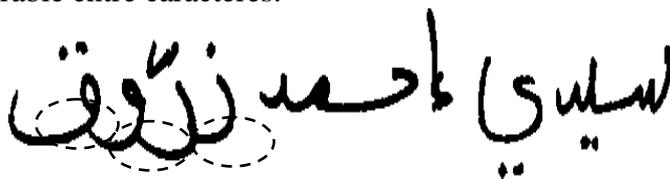


Figure 3.3: Exemple de liaison indésirable entre caractères.

- ☒ Coupure indésirable



Figure 3.4: Exemple de coupure indésirable.

III.3.1.2.4 Evaluation des résultats de segmentation

Malgré les efforts et les travaux intensifs réalisés dans le domaine de la segmentation de mots manuscrits, aucune méthode ne trouve pas des problèmes. Mais au fur et à mesure les auteurs essayent d’améliorer les résultats.

III.3.1.3 Image de caractère

Après la segmentation nous obtenons des objets avec des bruits, l’objectif de l’étape suivant est de faire un traitement sur ces objets pour produire des images claires.

III.3.1.4 Elimination de bruit et extraction d’objet concerne

Dans cette étape nous allons faire une élimination des éléments bruis en fonction de seuil prédéfini puis extraire l’objet concerne. Tous les éléments des taille continué compose l’objet concerne.

Exemple :

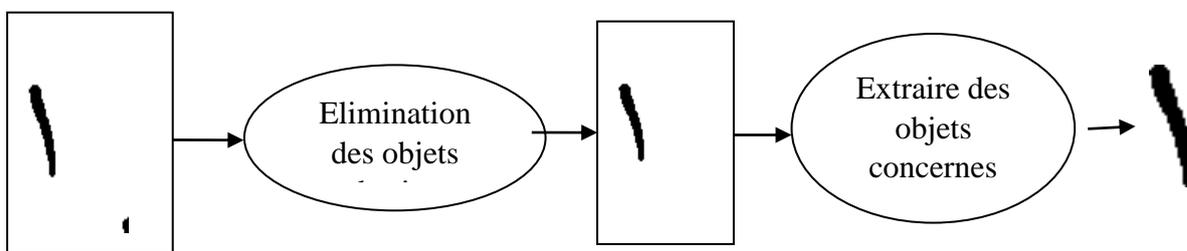


Figure 3.5 : Exemple d’élimination des objets bruits et extraire des objets concernes.

➤ Les problèmes d'élimination des objets bruits

Dans l'élimination d'objet bruit on peut avoir plusieurs cas indésirable par exemple le cas où les caractères contient des point diacritique.

III.3.1.5 Normalisation des images

Nous avons effectué cette opération pour éliminer les conditions qui peuvent fausser les résultats.

- **Dans le cas de moment de zernik**

Après élimination d'objet bruit et extraire d'objet concerne nous centrons cette objet et mettons tous les images dans même dimension. Parce que les moments de zernik sont invariants par apport à la translation et le changement d'échelle, donc les images de tous les caractères se retrouvent définies dans une matrice de même taille.

- **Dans le cas de zoning (densité)**

Après l'extraction de l'objet concerne nous effectuons le zonage expliqué dans (paragraphe II.2.1).en utilisant le grille ($n=5, m=5$), on obtient le même nombre de zones pour tous les images.

Exemple :

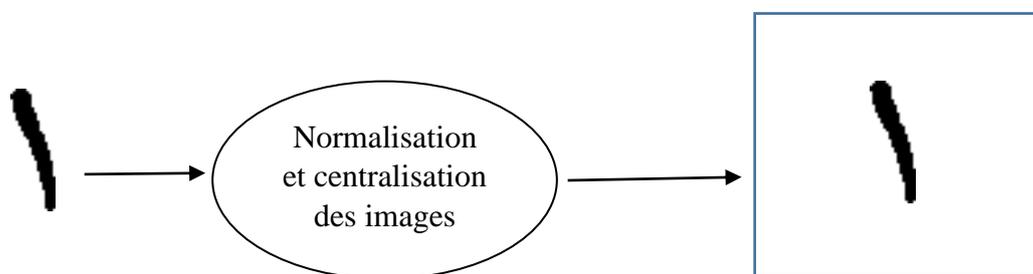


Figure 3.6 : Exemple de normalisation et centralisation des images.

III.3.1.6 L'extraction des primitives

L'extraction des caractéristiques souvent appelées primitives, consiste à représenter les données d'entrée (caractères) en un vecteur de primitives de dimension fixe.

C'est l'une des étapes les plus délicates et les plus importantes. La reconnaissance d'un caractère passe d'abord par l'analyse de sa forme et l'extraction de ces caractéristiques (primitives) qui seront exploités pour son identification.

Nous avons utilisé les moments de Zernik, et en faire le changement de l'ordre (n) et la répétition (m) on obtient le vecteur de caractéristique.

Ensuite nous avons utilisé une technique qui donne un vecteur de caractéristique binaire (0 si le pixel de l'image est noire et 1 sinon), la taille de vecteur de caractéristique (une image normaliser implique un vecteur de taille n×m).

Dans notre système, on a fait la concaténation entre le vecteur de zernik et le vecteur de zonage.

Exemple :

1	2	3	...	10	11	12	...	32	33	34	...	53	54	55
5.94	4.02	5.80	5.53	0.81	0.12	...	16	18	18	...	11	0	0

III.3.1.7 Création de corpus

Le corpus est un fichier (.mat) possède une structure particulière pour pouvoir être utilisée directement par l'application SVM multi-classe. Chaque ligne du corpus représente le vecteur de caractéristique d'une image, la première valeur de la ligne est le numéro de la classe, et le reste représente les caractéristiques primitives de l'image.

Nous présentons ci-dessous des lignes de corpus d'apprentissage pour la lettre «Alif_isolé » qui a le numéro de la classe 1.

Exemple :

1	9.1715	0.5401	9.1794	0.1768	1.0537	0.0192	9.1886	0.5047	0.0090	1.5355	0.0717	9.9207e-04	9.1843
1	10.0921	3.3484	9.5762	0.2259	5.8094	0.3245	8.7071	0.6346	0.0312	6.8309	1.1139	0.0403	7.8149
1	11.5424	0.6025	11.1861	0.0842	1.2014	0.0137	10.5159	0.2446	0.0017	1.8039	0.0519	6.3660e-04	9.6133
1	5.7467	0.9322	4.7672	0.7587	1.5957	0.1482	4.0550	1.9494	0.0811	2.1114	0.5003	0.0168	4.9466
1	6.8199	1.8852	6.3821	0.1995	3.6228	0.0361	5.5899	0.5698	0.0075	5.0874	0.1360	0.0012	4.6285
1	9.6246	1.2823	10.2868	0.4157	2.2926	0.1320	11.5161	1.1401	0.0210	2.8270	0.4783	0.0130	13.0886
1	0.2544	0.8467	0.2523	0.1801	1.5518	0.0412	0.5036	0.4977	0.0191	2.0223	0.1455	0.0027	0.9011
1	20.1481	2.3162	16.0660	1.9933	4.1415	0.3218	9.7778	4.9916	0.2048	5.0957	1.1289	0.0535	5.5213
1	14.3458	0.8129	13.1956	0.6606	1.5197	0.0621	11.1808	1.8459	0.0304	2.0691	0.2277	0.0046	8.8757
1	16.9530	3.7572	15.2718	1.1727	6.1141	0.4155	12.8227	3.0433	0.1094	6.2786	1.3952	0.0510	10.9408

Figure 3.7 : Exemple de corpus

III.3.1.8 L'apprentissage

L'apprentissage a pour but de généraliser l'information portée par les exemplaires de l'ensemble d'apprentissage pour faire émerger les représentations des classes. Ces représentations doivent permettre de s'affranchir de la variabilité des exemplaires d'une classe lors de la prise de décision. La validation de l'apprentissage est faite par les taux de reconnaissance obtenus sur un ensemble test constitué d'exemplaires différents de ceux de l'ensemble d'apprentissage.

III.3.1.9 Création des séparateurs

Dans cette phase nous avons utilisé l'approche une contre une pour crée des ensemble des séparateur. L'approche une contre une requiert l'apprentissage de $\frac{1}{2} m (m - 1)$ classifieurs pour tous les couples de classes possibles

III.3.2 Reconnaissance (hors ligne)

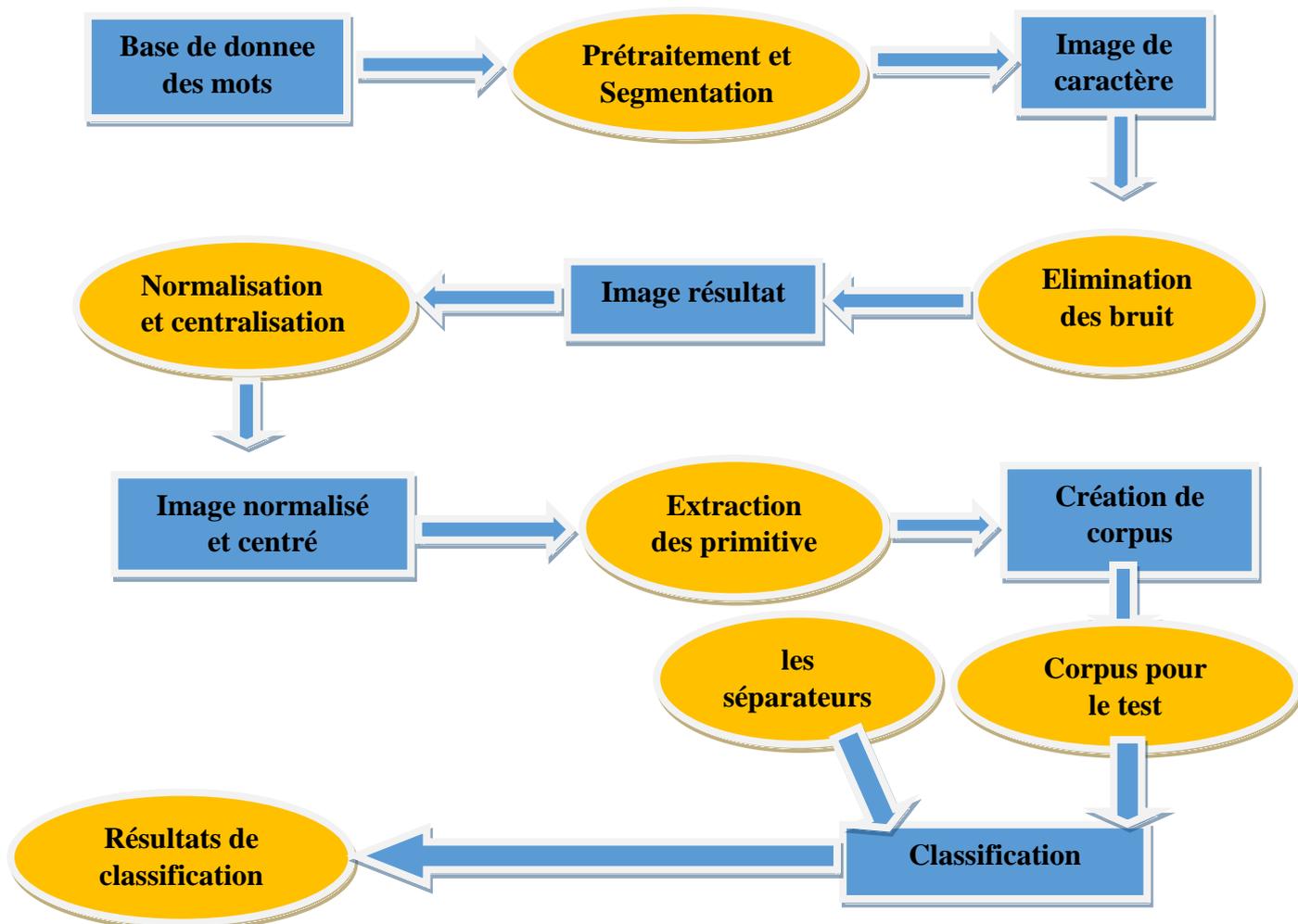


Figure 3.8 Schéma général de classification.

Dans la reconnaissance nous utilisons les mêmes étapes que dans la création des séparateurs, seulement dans l'étape de création de corpus nous utilisons une base de donne différente que la base de donner de l'apprentissage, à la fin nous sortons par un corpus de test.

III.3.2.1 Classification

Nous préparons le corpus de test. Alors nous comparons le corpus par les séparateurs et nous sortons par les résultats de classification.

III.4 Résultats et discussion

Nous avons utilisé une base donne de 87 classe, chaque classe contient plusieurs image parmi (60 - 20) images.

Classe	Exemple	Classe	Exemple	Classe	Exemple
Alif_isolé	ا	Mim_début	م	Kha_isolé	خ
Alif_médiale_finale	ا	Mim_finale	م	Kha_médiale	خ
Ayn_début	أ	Mim_isolé	م	lam_alif	ا
Ayn_finale	أ	Mim_médiale	م	lam_début	ا
Ayn_isolé	أ	Noun_début	ن	lam_finale	ا
Ayn_médiale	أ	Noun_finale	ن	lam_isolé	ا
Ba_debut	ب	Noun_isolé	ن	lam_médiale	ا
Ba_finale	ب	Noun_médiale	ن	Ya_début	ي
Ba_médiale	ب	Qaf_début	ق	Ya_finale	ي
Chin_debut	چ	Qaf_finale	ق	Ya_isolé	ي
Chin_finale	چ	Qaf_isolé	ق	Ya_médiale	ي
Chin_médiale	چ	Qaf_médiale	ق	Zâ_debut	ز

Dâd_début		Ra_finale_médiale		Zâ_médiale	
Dâd_finale		Ra_isolé		Zay_finale_médiale	
Dâd_médiale		Rayn_début		Zay_isolé	
Dal_isolé		Sâd_début		Jim_debut	
Dal_médiale_finale		Sâd_finale		Jim_finale	
Dhal_médiale_finale		Sâd_médiale		Jim_isolé	
Fa_début		Sin_début		Jim_médiale	
Fa_médiale		Sin_finale		Kaf_début	
Ha_debut		Sin_isolé		Ta_médiale	
Ha_finale		Sin_médiale		Tâ_médiale	
Ha_isolé		ta_almarbouta		Tha_debut	
Ha_isolé		Ta_debut		Tha_finale	
Hamza_nabira		Tâ_debut		HHa_médiale	
HHa_début		Ta_finale		Tâ_isolé	
HHa_finale		Tâ_finale			

Tableau 3.2 : Exemple des classes utilisé

Pour tester notre système nous avons créé une matrice de confusion pour chaque cas et nous arrivons les résultats suivant :

III.4.1 En cas de zernik

Nous avons formé un vecteur de caractéristique pour chaque image à partir des paramètres de zernik (n,m),à chaque fois que nous changent n et m , nous obtenons une valeur de A_{nm} et ainsi de suite jusqu'à ce que composant un vecteur de taille k ,après nous testons tous les image par chaque séparateur. Nous avons créé dix (10) vecteurs pour chaque image par des tailles différentes pour les différentes valeurs de n et m , les valeurs des n et m vérifier les condition $n-|m|$ pair et $|m| \leq n$.

Nous avons obtenons les résultats suivants :

Taille de vecteur	10	20	30	40	50	60	70	80	90	100
Taux de reconnaissance (%)	41.53	53.26	56.27	63.08	71.11	72.3	75.27	78.21	80.45	80.82
	3	6	7	8	1	3	7	1	5	2

Tableau 3.3 : Taux de reconnaissance zernike

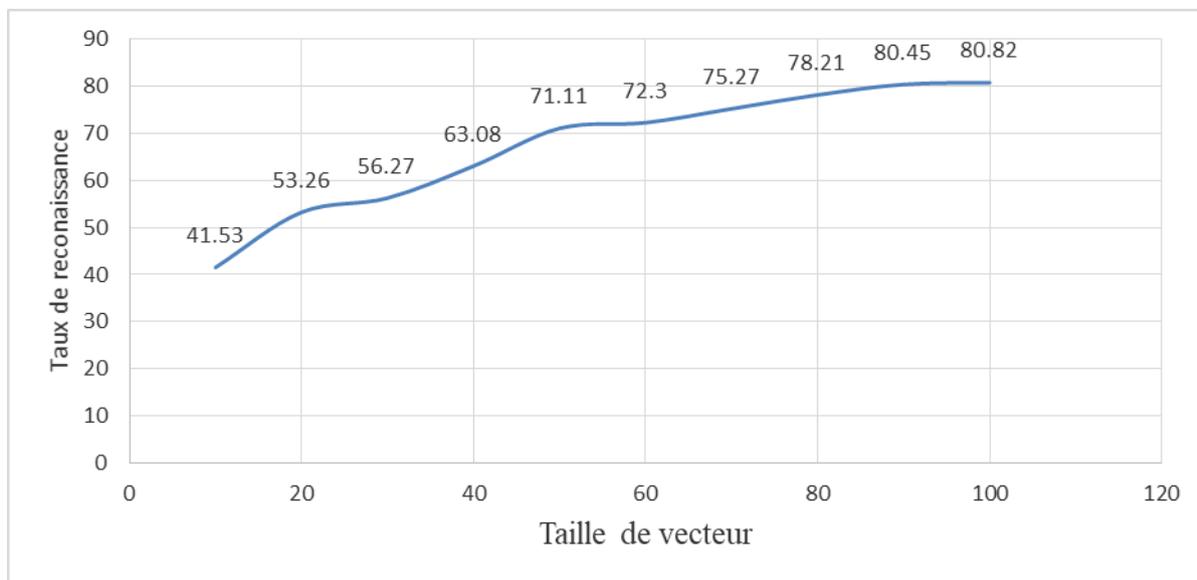


Figure 3.9 : Taux de reconnaissance de zernike

Nous observons qu’à chaque augmentation de la taille de vecteur le taux de reconnaissance augmenter. Dans la taille 100 n=18 et m=18. Donc on peut disons que à chaque augmentation de l’ordre n et la répétition m le taux sera augmenté.

Taille de vecteur	100	90	80	70
Classe 1	95.08	96.72	93.44	93.44
Classe 2	87.67	84.93	83.56	76.71
Classe 23	83.33	100.00	100.00	83.33
Classe 28	91.67	83.33	50.00	75.00
Classe 40	84.62	61.54	38.46	30.77
Classe 47	0.00	10.00	0.00	20.00
Classe 55	74.07	77.78	81.48	85.19
Classe 74	30.77	15.39	0.00	0.00
Classe 75	40.00	70.00	90.00	50.00
Classe 77	40.00	58.00	20.00	30.00
Classe 79	0.00	0.00	0.00	60.00
Classe 82	61.11	33.33	22.22	5.56
Classe 83	65.22	39.13	47.83	43.48
Classe 84	20.00	60.00	40.00	40.00
Classe 86	54.55	54.55	72.73	54.55
Classe 87	26.00	24.00	24.00	0.00

Tableau 3.4 : variante taux pour chaque classe

Mais le taux pour chaque classe est varié, une fois augmenté, une fois démunie et peut être devient zéro. Par exemple dans la classe 1 nous observons que le taux dans la taille 100 (n=18, m=18) est plus élevé sur la taille 80 (n=16, m=14) et la taille 70 (n=15, m=11) mais moins que la taille 90 (n=17 , m=17),et dans la classe 75 (n=16, m=4) et la classe 79 (n=16, m=12) le taux est en diminuant. Et dans la plupart des classes, nous trouvons le taux qu’il augmente.

Enfin nous prenons le vecteur qui donne meilleur résultats, c’est le cas de vecteur de taille 100 (n=18, m=18).

N° classe	Taux de reconnaissance	N° classe	Taux de reconnaissance	N° classe	Taux de reconnaissance
classe 1	95.08	classe 30	66.67	classe 59	80.00
classe 2	87.67	classe 31	100.00	classe 60	100.00
classe 3	90.00	classe 32	100.00	classe 61	75.00
classe 4	100.00	classe 33	100.00	classe 62	0.00
classe 5	100.00	classe 34	100.00	classe 63	100.00
classe 6	100.00	classe 35	100.00	classe 64	66.67
classe 7	80.00	classe 36	100.00	classe 65	100.00
classe 8	100.00	classe 37	100.00	classe 66	100.00
classe 9	100.00	classe 38	70.00	classe 67	33.33
classe 10	100.00	classe 39	100.00	classe 68	100.00
classe 11	100.00	classe 40	84.62	classe 69	100.00
classe 12	90.00	classe 41	100.00	classe 70	100.00
classe 13	90.00	classe 42	88.89	classe 71	100.00
classe 14	100.00	classe 43	64.82	classe 72	56.25
classe 15	100.00	classe 44	88.89	classe 73	34.00
classe 16	100.00	classe 45	62.50	classe 74	30.77
classe 17	100.00	classe 46	90.00	classe 75	40.00
classe 18	100.00	classe 47	0.00	classe 76	79.59
classe 19	100.00	classe 48	72.73	classe 77	40.00
classe 20	60.00	classe 49	100.00	classe 78	100.00
classe 21	91.67	classe 50	100.00	classe 79	0.00
classe 22	80.00	classe 51	33.33	classe 80	100.00
classe 23	83.33	classe 52	43.64	classe 81	100.00
classe 24	100.00	classe 53	41.18	classe 82	61.11
classe 25	100.00	classe 54	100.00	classe 83	65.22
classe 26	100.00	classe 55	74.07	classe 84	20.00
classe 27	100.00	classe 56	81.82	classe 85	100.00
classe 28	91.67	classe 57	100.00	classe 86	54.55
classe 29	100.00	classe 58	66.67	classe 87	26.00
Taux de reconnaissance			80.82 %		

Tableau 3.5 : résultats de moment de zernik pour taille vecteur 100 (n=18, m =18)

Nous remarquons dans les résultats que à chaque fois que la valeur de m et n grand le résultat est grand et le taux de reconnaissance c'est mieux que le précédent. Enfin cette méthode donne des résultats mieux s'il m et n est grand et la taille de vecteur grand.

III.4.2 En cas de zonage

Dans cette méthode Nous produisons un vecteur de taille 25 pour chaque image à partir d'algorithme de zonage et nous testons chaque image par tous séparateur, nous obtenons les résultats suivante :

N° classe	Taux de reconnaissance	N° classe	Taux de reconnaissance	N° classe	Taux de reconnaissance
classe 1	100	classe 30	100	classe 59	100
classe 2	95.89	classe 31	100	classe 60	100
classe 3	100	classe 32	100	classe 61	100
classe 4	100	classe 33	100	classe 62	56.25
classe 5	90	classe 34	90	classe 63	100
classe 6	100	classe 35	100	classe 64	100
classe 7	100	classe 36	100	classe 65	100
classe 8	100	classe 37	100	classe 66	100
classe 9	100	classe 38	100	classe 67	66.66
classe 10	100	classe 39	100	classe 68	100
classe 11	100	classe 40	100	classe 69	100
classe 12	100	classe 41	100	classe 70	100
classe 13	80	classe 42	100	classe 71	100
classe 14	100	classe 43	70.37	classe 72	87.5
classe 15	90	classe 44	77.77	classe 73	56
classe 16	100	classe 45	100	classe 74	84.61
classe 17	100	classe 46	100	classe 75	100
classe 18	100	classe 47	100	classe 76	24.49
classe 19	100	classe 48	72.72	classe 77	90
classe 20	90	classe 49	100	classe 78	100
classe 21	83.33	classe 50	100	classe 79	60
classe 22	100	classe 51	41.66	classe 80	100
classe 23	100	classe 52	94.54	classe 81	100
classe 24	100	classe 53	88.23	classe 82	66.66
classe 25	100	classe 54	100	classe 83	56.52
classe 26	100	classe 55	100	classe 84	60
classe 27	100	classe 56	100	classe 85	100
classe 28	100	classe 57	100	classe 86	100
classe 29	100	classe 58	91.66	classe 87	56
Taux de reconnaissance			92.19		

Tableau 3.6 : résultats de zonage

Nous avons fait les même étapes, mais nous changeons pour chaque étapes les valeurs des n et m nous obtenons par chaque cas un vecteur de $n \times m$, voir les résultats :

Taille de vecteur $n \times m$ n et m	16 $n,m=4$	25 $n,m=5$	36 $n,m=6$	49 $n,m=7$	64 $n,m=8$	81 $n,m=9$	100 $n,m=10$
Taux de reconnaissance (%)	92.19	92.19	92.19	92.19	92.19	92.19	92.19

Tableau 3.7 : résultats de la variante zonage

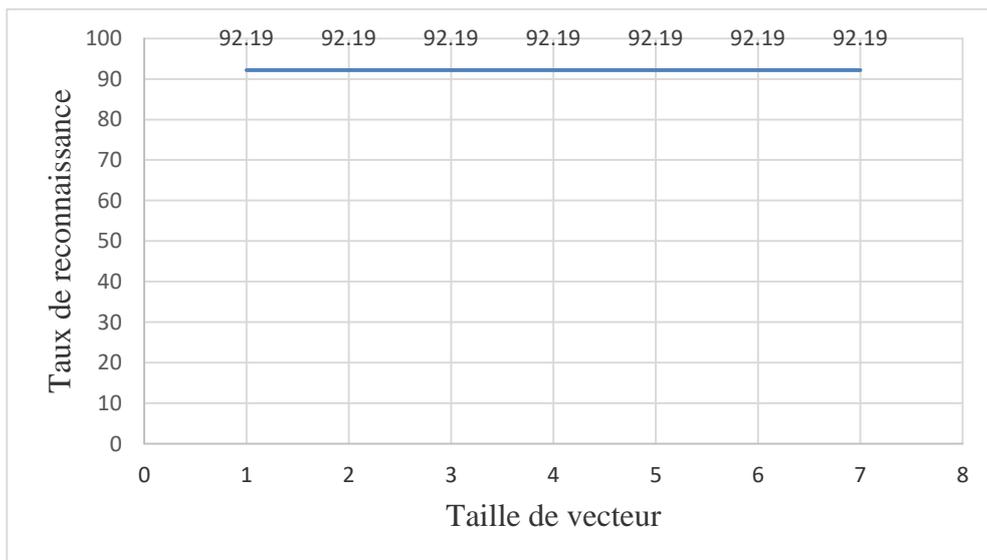


Figure 3.10 : Taux de reconnaissance de zonage

Nous observons par chaque taille de vecteur que nous obtenons les mêmes résultats pour le taux en générale et pour chaque classe, contrairement à la méthode de zernik qu’était changeait. Donc n et m n’est pas faire des effets sur la méthode de zonage.

A travers les résultats des zonages (Tableau 3.5) nous trouvons la plupart des résultats ci-dessus de 50%, il y a deux classe moins de 50% (classe 51 et classe 76), nous disons que tous les classes sont reconnues, parmi ces classes sont perte de donner avec ces voisin, d’après ci-dessus nous observons que cette méthode contient un peu d’erreur.

III.4.3 En cas d’hybridation

Dans l’hybridation nous avons fait une concaténation entre le vecteur de moment zernik et le vecteur de zonage, nous obtenons un vecteur de taille 125, nous testons notre système par la matrice de confusion, nous venons par les résultats suivant :

N° classe	Taux de reconnaissance	N° classe	Taux de reconnaissance	N° classe	Taux de reconnaissance
classe 1	100	classe 30	100	classe 59	70
classe 2	98.63	classe 31	100	classe 60	100
classe 3	100	classe 32	100	classe 61	100
classe 4	100	classe 33	100	classe 62	75
classe 5	100	classe 34	100	classe 63	100
classe 6	100	classe 35	100	classe 64	100
classe 7	100	classe 36	100	classe 65	100
classe 8	100	classe 37	100	classe 66	100
classe 9	100	classe 38	100	classe 67	33.33
classe 10	100	classe 39	100	classe 68	100
classe 11	100	classe 40	100	classe 69	100
classe 12	100	classe 41	100	classe 70	100
classe 13	100	classe 42	100	classe 71	100
classe 14	100	classe 43	100	classe 72	100
classe 15	90	classe 44	100	classe 73	88
classe 16	100	classe 45	75	classe 74	69.23
classe 17	100	classe 46	100	classe 75	80
classe 18	100	classe 47	100	classe 76	91.83
classe 19	100	classe 48	81.81	classe 77	42
classe 20	100	classe 49	100	classe 78	100
classe 21	100	classe 50	100	classe 79	0
classe 22	100	classe 51	100	classe 80	50
classe 23	100	classe 52	100	classe 81	100
classe 24	100	classe 53	100	classe 82	72.22
classe 25	100	classe 54	100	classe 83	91.30
classe 26	100	classe 55	59.25	classe 84	100
classe 27	100	classe 56	100	classe 85	100
classe 28	91.66	classe 57	100	classe 86	72.72
classe 29	100	classe 58	100	classe 87	32
Taux de reconnaissance			92.69		

Tableau 3.8 : résultats de concaténation zernike et zonnig.

Dans la concaténation nous obtenons des résultats supérieurs à 80% parmi tous les classes seulement les cinq (5) classes sont moins de 60%, nous arrivons que la plupart des classes sont reconnus et le taux de reconnaissance se mieux que la premier méthode et le deuxième.

En comparaison, nous trouvons dans le cas de zonage la classe 83 le taux est 56.52% et dans l'hybridation est 91.30%, elle était augmenté par 20 % presque et la même chose dans la classe 82, 87, 76,73...etc. Mais dans la classe 86 elle était diminuée.

En cas de concaténation entre zonage et zernik nous obtenons les résultats suivants :

N° classe	Taux de reconnaissance	N° classe	Taux de reconnaissance	N° classe	Taux de reconnaissance
classe 1	100	classe 30	100	classe 59	100
classe 2	98.63	classe 31	100	classe 60	100
classe 3	100	classe 32	100	classe 61	100
classe 4	100	classe 33	100	classe 62	87.5
classe 5	100	classe 34	100	classe 63	100
classe 6	100	classe 35	80	classe 64	100
classe 7	100	classe 36	100	classe 65	100
classe 8	100	classe 37	100	classe 66	100
classe 9	100	classe 38	100	classe 67	33.333
classe 10	100	classe 39	100	classe 68	100
classe 11	100	classe 40	100	classe 69	100
classe 12	100	classe 41	100	classe 70	100
classe 13	100	classe 42	100	classe 71	100
classe 14	100	classe 43	100	classe 72	87.5
classe 15	100	classe 44	33.333	classe 73	86
classe 16	100	classe 45	100	classe 74	30.769
classe 17	100	classe 46	90	classe 75	100
classe 18	100	classe 47	60	classe 76	100
classe 19	100	classe 48	90.909	classe 77	100
classe 20	100	classe 49	100	classe 78	100
classe 21	75	classe 50	100	classe 79	20
classe 22	100	classe 51	58.333	classe 80	100
classe 23	100	classe 52	94.545	classe 81	100
classe 24	100	classe 53	58.824	classe 82	100
classe 25	100	classe 54	100	classe 83	86.957
classe 26	100	classe 55	100	classe 84	100
classe 27	100	classe 56	81.818	classe 85	100
classe 28	100	classe 57	100	classe 86	100
classe 29	100	classe 58	100	classe 87	36
Taux de reconnaissance			92.98		

Tableau 3.9 : résultat concaténation zonage et zernik

Nous observons que le taux est augmenté par 0.29 %, donc c'est mieux quand faire une concaténation entre zonage et zernik.

Dans le cas de reconnaissance nous avons créé nouveau base de donner pour le test contient 640 images, différente a la base d'apprentissage et nous faisons un test par chaque cas sur notre système.

Nous trouvons dans les plupart classe, le taux était fait amélioration dans la méthode d'hybridation contrairement à chaque méthode, par exemple la classe 17(dal_medial) 60% dans méthode zernike,100% dans l'hybridation, la classe 13(dad_debut)0% dans Zernike et zonage, 28.57% dans l'hybridation, ainsi de suite les classes ,2 (Alif_médiale_finale), 3 (Ayn_début), 5 (Ayn_isolé), 8 (Ba_finale) et 23 (Ha_finale)...etc.

Mais nous trouvons des classes mal taux de reconnaissance à cause les problèmes suivant :

- Segmentation des mots mal segmente.
- chevauchement de caractère dans élimination de bruit.
- Peut-être le bruit est grand para port le caractère.
- La taille de la base donne asymétrique.

III.5 Conclusion

Nous avons présenté dans ce chapitre une description détaillée de notre système de reconnaissance de caractères Arabes manuscrits, avec l'évaluation de performance de chaque phase de processus de reconnaissance. Nous avons présenté aussi les résultats d'expérimentations par l'utilisation de deux méthodes d'extraction des caractéristiques (méthode statistique et moment de Zernike).

Conclusion

générale

Conclusion générale

La reconnaissance de l'écriture manuscrite est un axe de recherche récent et qui est prometteur pour des travaux futurs en vue d'augmenter le taux de reconnaissance, la fiabilité, et d'optimiser les performances de nos machines et l'intégrité des systèmes de reconnaissance. L'objectif de ce mémoire est de proposer un système de reconnaissance de l'écriture manuscrite arabe hors-ligne. Le travail réalisé porte essentiellement sur la reconnaissance des caractères segmenté du texte. La méthode utilisée dans ce travail a l'intérêt d'être robuste, et les résultats obtenus fournissent globalement une solution de qualité.

La sélection des caractéristiques est une problématique complexe et d'une importance cruciale pour les systèmes d'apprentissage, donc il faut extraire les caractéristiques les plus pertinentes de l'image pour ne garder qu'une quantité réduite d'informations. Plusieurs caractéristiques ont été utilisées. Chaque caractéristique possède des inconvénients et des avantages et à ce jour il n'existe pas de primitive fiable à 100 %. Pour remédier, il sera intéressant d'utiliser plusieurs primitives à la fois. Durant notre travail, deux primitives ont été utilisées, les moments invariants de Zernike et caractéristique statique (zonage).

Et d'après les résultats obtenus, on a constaté que les caractéristiques statiques (zonage) sont avantageuses. Mais avec une combinaison entre les deux caractéristiques on a le pouvoir d'augmenter le taux de reconnaissance alors L'analyse des résultats obtenus par la version améliorée du système montre une amélioration significative du taux de reconnaissance par rapport à la première alternative. D'une manière générale, le but de ce projet n'est pas suffisamment atteint du fait que le taux de reconnaissance soit inférieur à 90%. Mais ceci n'est pas fatal, il s'explique par le nombre réduit d'exemples de la base d'apprentissage.

Pour achever, même si la méthode SVM utilisé pour l'apprentissage a un fondement théorique, rien ne garantit qu'une SVM conçu pour un problème de classification ait en toutes situations les performances attendues. La qualité de la base de données ainsi que les primitives sélectionnées influent énormément sur le système de reconnaissance des caractères manuscrits.

Plusieurs améliorations et perspectives sont envisageables pour poursuite de nos travaux, en particulier :

L'utilisation des méthodes d'apprentissage, comme modèles de Markov cachés en combinaison avec d'autres classifieurs notamment les réseaux de neurones peut diminuer le taux d'erreur de notre système et par conséquent avoir un système fiable.

Développement des techniques de segmentation, et enrichie la base de données des caractères pour la phase d'apprentissage.

Référence

Référence

- [01] P. Burrow : « Arabic handwriting recognition ». Master of science thesis. School of Informatics, university of Edinburg, England, 2004.
- [02] B. Al-Badr, R.M. Haralick : « Segmentation-free word recognition with application to Arabic ». IEEE. Proc. 3rd International conference on document analysis and recognition (ICDAR'95), pp. 355-359, Montreal, Canada, 1995.
- [03] Mamouni el mamoun «Sélection expérimentale de modèles SVM multi-classes application à la reconnaissance des caractères arabes».
- [04] S. Snoussi maddouri, « Modèle perceptif neuronal à vision globale-locale pour la reconnaissance de mots manuscrits arabes », 2002.
- [05] Haitaamar Schahrazed, « Segmentation de textes en caractères pour la reconnaissance optique de l'écriture arabe ». Thèse de magister université Elhadj-Lakhdar Batna, 2007.
- [06] F. Menasri, N. Vincent, E. Augustin, et M. Cheriet, “Un système de reconnaissance de mots arabes manuscrits hors-ligne sans signes diacritiques”, Proceedings du Colloque International Francophone sur l'Ecrit et le Document (CIFED), Rouen-France, pp. 121-126, 2008.
- [07] Richard Alan Peters II, « Image processing (digital image and matlab, frequency filtering) », 2007.
- [08] Somaya A. S. Al-Ma'adeed, « Recognition of Off-line Handwritten Arabic Words », University of Nottingham for the degree of Doctor of Philosophy, June 2004.
- [09] N. Benamara : « Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée ». Thèse de doctorat, spécialité Génie Electrique, Université des sciences, des Techniques et de médecine de Tunis II, 1999.
- [10] E.Lecolinet, O. Barrett : « Cursive word recognition : Methods and strategies ». In NATO/ASI , Fundamentals in handwriting recognition, Bonas, France June 21-july 3, 1993.
- [11] B. Al-Badr , S.A. Mahmoud : « Survey and bibliography of Arabic optical text recognition ». Signal processing , vol. 41, pp. 49-77, 1995.
- [12] P.M. Lallican, C. Viarp-Gaudin, S. Knerr : « From off-line to on-line handwriting recognition ». Proc. 7th workshop on frontiers in handwriting recognition, pp. 303-312, Amsterdam 2000.
- [13] I.R. Tsang : «Pattern recognition and complex systems». Thèse de doctorat, université d'Anterwerpen, 2000.

- [14] J. Trenkle, A. Gillies, S.Schlosser : « An off-line Arabic recognition system for machine printed documents ». Proc. Of the symposium on document image understanding technology (SDIUT'97), pp. 155-161 1997.
- [15] Maged Mohamed Mahmoud Fahmy, S.Al Ali : « Automatic recognition of handwritten Arabic characters using their geometrical features ». Studies in informatics and control journal (SIC journal), vol. 10, No 2, 2001.
- [16] J. Anigbogu : « Reconnaissance de textes imprimés mutifontes à l'aide de modèles stochastiques et métriques ». thèse de doctorat, Université de Nancy I, 1992.
- [17] Riadh Bouslimi, « Système de reconnaissance hors-ligne des mots manuscrits arabe pour multi-scripteurs » mémoire de master soutenue le 21/10/2006.
- [18] R.G. Casey, E. Lecolinet : « Strategies in character segmentation : A survey ». IEEE. Proc. 3rd international conference on document Analysis and recognition (ICDAR'95), pp. 1028-1033, Montreal, Canada, 1995.
- [19] B. Al-Badr , R.M. Haralick : « Symbol recognition without prior segmentation ». Conference SPIE-EI 1994.
- [20] A. Belaïd, « Reconnaissance de l'Écriture et Analyse de Documents : Numérisation, Prétraitements », 2006.
- [21] O. D. Trier and T. Taxt. « Evaluation of binarization methods for document images, On Pattern Analysis and Machine Intelligence », vol. 11, n.12, pp. 312- 314, December 1995.
- [22] H. Emptoz, F. Lebourgeois, V. Eglin, Y. Leydier. « La reconnaissance dans les images numérisées : OCR et transcription, reconnaissance des structures fonctionnelles et des métadonnées », 2003.
- [23] T. Steinherz, E. Rivlin, N. Intrator : «Off-line cursive word recognition : a survey ». International journal on document analysis and recognition, 2(2), pp. 90-110, 1999.
- [24] A. Belaïd, « Analyse et reconnaissance des documents », <http://www.loria.fr/~abelaid/Teaching> (support de cours), 2006.
- [25] Dr Suici-Meslati, « Reconnaissance de formes et écriture arabe manuscrite », support de cours 2008.
- [26] Najoua Ben Amara et al, « Modélisation Pseudo Bidimensionnelle pour la Reconnaissance de Chaînes de Caractères Arabes Imprimés », 1999.
- [27] Yousef Al-Ohali, MohamedCheriet et Ching Suen, « Databases for recognition of handwritten Arabic cheques ». Pattern Recognition Society 36 111 – 121. 2003.

- [28] Mustapha Kadri, « Méthode de reconnaissance de l'écriture arabe manuscrite en utilisant les réseaux neuronaux » Mémoire de magister Université des Sciences et des Technologies Mohamed Boudiaf d'Oran, 2010.
- [29] Najoua Ben Amara et al, « Une méthode stochastique pour la reconnaissance de l'écriture arabe imprimée », 1996.
- [30] Philippe BOLON ET AL, « Analyse d'image, Traitements de bas niveau » (support de cours), 1998.
- [31] M. Mansour, M. Benkhadda & A. Benyettou, « Optimized Segmentation Techniques for Handwritten Arabic Word and Numbers Character Recognition », 2005.
- [32] S. Kermi : « Classifieur neuronal base connaissances, application à la reconnaissance des caractères arabes isolés manuscrits ». Thèse de magister, université Badji Mokhtar, Annaba, Algérie 1999.
- [33] Muhammad Sarfraz, « Computer -Aided intelligent recognition techniques and applications » (section : Offline Arabic Character Recognition) King Fahd University of Petroleum and Minerals, Kingdom of Saudi Arabia, 2005.
- [34] Trevor Hastie and Patrice Y. Simard, « Metrics and Models for Handwritten Character Recognition ». Statistical science Vol. 13 N° 1, 54 -65, 1998.
- [35] Sander Bohte, « Spiking neural networks », Thesis University Leiden.2003,
- [36] Antoine Cornuéjol et Laurent Miclet, Livre : « Apprentissage artificiel : concepts et algorithmes », Préface de Tom Mitchell. Édition EYROLLES deuxième tirage 2003.
- [37] P. J. Werbos. Beyond regression : « New Tools for Prediction and Analysis in the Behavioral Sciences ». PhD thesis, Masters Thesis, Harvard University, 1974.
- [38] D. E. Rumelhart et al. « Learning internal representations by error backpropagation ». In J. L. McClelland In D. E. Rumelhart and the PDP research group, editors, Parallel distributed processing: explorations in the microstructure of cognition, pages 318–362. 1986.
- [39] Y. LeCun et al. « Backpropagation applied to handwritten zip code recognition ». Neural Computation, 1(4):541–551, Winter 1989.
- [40] A. Kolmogorov. « Three approaches to the quantitative definition of information ». Problems Inform. Transmission, 1(1) :1–7, 1965.
- [41] S. Alma'adeed, C. Higgins, D. Ellima, « Recognition of Off-Line Handwritten Arabic Words Using Hidden Markov Model Approach », ICPR, University of Nottingham, 2002.
- [42] Aburas A. A., Rehiel S. M. A., « Off-line Omni-style Handwriting Arabic Character Recognition System Based on Wavelet Compression », International Islamic University Malaysia, Electrical and Computer Engineering, Malaysia, 2007.

- [43] Jannoud I. A., « Automatic Arabic Hand Written Text Recognition System ». Damascus University, Damascus, Syria and Al-zaytoonah University, Amman, Jordan, American Journal of Applied Sciences 4 (11): 857-864, 2007.
- [44] Sari T., Souici L., Sellami M., « Off-line Handwritten Arabic Character Segmentation Algorithm: ACSA », Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), Algeria, 2002.
- [45] Zaïz Faouzi et al « SVM pour la reconnaissance de caractères manuscrits arabes » Laboratoire LESIA, Département d'Informatique, Université Mohamed Khider Biskra, 2010.
- [46] Nedjem eddine Ayat, « sélection de modèle automatique des machines à vecteurs de support: application à la reconnaissance d'images de chiffres manuscrits » Thèse de doctorat montréal, le 20 janvier 2004.
- [47] O. D. Trier, A. K. Jain, T. Taxt, « Feature extraction methods for character recognition - a survey », Pattern Recognition, Vol. 29, 1996, pp. 641-662.
- [48] M. Milgram, « Reconnaissance des formes méthodes numériques et connexionnistes », Edition Armand colin, Paris, 1993.
- [49] D. Arrivault, « Apport des Graphes dans la Reconnaissance Non-Contrainte de Caractères Manuscrits Anciens », Thèse pour l'obtention du Grade de Docteur de l'université de Poitiers, 2006.
- [50] V. N. Vapnik. « Estimation Dependences Based on Empirical Data. Springer »-Verlag, New York, USA, 1982.
- [51] Christopher J. C. Burges. « A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery », 2(2) :121–167, 1998.
- [52] Chih C. Chang and Chih J. Lin. « LIBSVM: a library for support vector machines », 2001.
- [53] Nello Cristianini and John Shawe-Taylor. « An Introduction to Support Vector Machines and Other Kernel-based Learning Methods ». Cambridge University Press, March 2000.
- [54] V. Blanz, B. Scholkopf, H.H. Bulthoff, C. Burges, V. Vapnik, and T. Vetter. « Comparison of view-based object recognition algorithms using realistic 3d models ». In ICANN, pages 251–256, 1996.
- [55] C. Cortes and V. Vapnik. « Support-vector networks. Machine Learning », 20(3):273-297, 1995.
- [56] BISSON, G. (2009). « Intelligence artificielle ».
- [57] H.Mohamadally, B.Fomani « SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges » Versailles St Quentin, France 16 janvier 2006.

- [58] J.Milgram, « Contribution à l'intégration des machines à vecteurs de support au sein des systèmes de reconnaissance des formes : Application à la lecture automatique de l'écriture manuscrite ». MONTRÉAL, LE 29 JUIN 2007
- [59] D.Kharroubi, Etude de techniques de classement "Machines à vecteurs supports" pour la vérification automatique du locuteur, Kharroubi DJamal (2002)
- [60] J.Milgram, Contribution à l'intégration des machines à vecteurs de support au sein des systèmes de reconnaissance des formes : Application à la lecture automatique de l'écriture manuscrite. MONTRÉAL, LE 29 JUIN 2007
- [61] Steve R. Gunn, « Support Vector Machines for Classification and Regression » University of Southampton, 10 May 1998
- [62] VAPNIK, V. (1998). « The Nature of Statistical Learning Theory ». Springer-Verlag.
- [63] BOTTOU, L. et CHIH-JEN, L. « Support Vector Machine Solvers, in Large Scale Kernel Machines ». MIT Press. 2007.
- [64] Ludovic Mercier, « Les machines à vecteurs support pour la classification en imagerie hyperspectrale : implémentation et mise en oeuvre. » UE ENG111 - Epreuve TEST Travail d'Etude et de Synthèse Technique en Informatique, 11 février 2010.
- [65] V.Vapnik, the nature of statistical learning theory. Springer Verlag, new York, 1995.
- [66] B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In KDD'95, pages 252_257, 1995.
- [67] R. Rifkin and A. Klautau. In defense of one-vs-all classification. Journal of Machine Learning Research, 5 :101_141, 2004.
- [68] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification", In advances in Neural Information Processing Systems 12,2000..
- [69] Robert Hastie, Trevor & Tibshirani. Stanford statistical learning online course SVM lecture. "Example and comparison with logistic regression". Web Video, Nov 2013. Accessed 15 Sep 2014.
- [70] Farès Menasri, "Contributions à la reconnaissance de l'écriture arabe manuscrite",
- [71] Safia KHERRASSI et Khayra BARKA, " Réalisation d'un système de reconnaissance de l'écriture arabe manuscrite à base d'automates à états finis pondérés",

Bibliographie :

- [72] <https://fr.wikipedia.org/wiki/MATLAB>.