

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 : Réseaux et Système Intelligents

Thème

Reconnaissance des montants littéraires arabes manuscrits

Préparés par
Fatima Zohra BOUREGAA

Encadré par
Mr. El Mamoun MAMOUNI

Année Universitaire 2016/2017



Remerciement

Nous remercions tout d'abord dieu le tout puissant de nous avoir donné le courage et la patience qui a éclairé notre chemin pour achever ce travail.

Nous remercions très vivement notre encadreur de mémoire « Mr. MAMOUNI Elmamoun » pour nous avoir suivis et soutenus dans ce parcours, pour nous avoir fait partager son expérience et ses connaissances.

Nous remercions Mr. KOHILI Mohammed de son soutien et son aide, et pour nous avoir conseillé et encouragé.

Nous remercions aussi Mr. DEMRI Mohammed pour ses conseils, son aide et son encouragement.

Nous remercions Mr. CHERAGUI Mohammed Amine » de son soutien et son aide. et pour nous avoir conseillé et encouragé.

Nous remercions docteur « OMARI Mohammed » de son soutien et son aide. et pour nous avoir conseillé et encouragé.

Nous respects et nous gratitudes vont également aux membres du jury qui nous ont fait l'honneur de juger ce travail et qui par leur disponibilité, leurs observations et leurs rapports nous ont permis d'enrichir notre travail.

Nous remercions également de tout notre cœur tous les enseignants qui ont contribué à notre apprentissage depuis notre jeune âge à ce jour.

Nous remercions toutes les personnes qui ont participé de manière directe ou indirecte à la concrétisation de ce travail.

Mme. Bouregaa Fatima Zohra





Dédicace

*Je dédie ce modeste travail à : A mes parents .Aucun
hommage ne pourrait être à la hauteur de l'amour Dont
ils ne cessent de me combler. Que dieu leur procure
bonne santé et longue vie.*

A mon cher mari. et bien sûr A mon fils Qussai

A mes frères abdelmalek, abdelalli. et

mes sœurs Khadija et Faiza et

tous les membre de la famille

« Bouregaa et Hamdi »

*A mes amies Yamina et Maissa, Et à tous ceux qui ont contribué de
près ou de loin pour que ce travail soit concrétisé, je*

vous dis merci

Mme. Bouregaa Fatima Zohra.



Résumé

Le travail présenté dans ce mémoire s'intègre dans le cadre général de la reconnaissance automatique de l'écriture arabe, principalement la reconnaissance automatique hors ligne de Montant littéraux arabe manuscrit. Nous avons présenté une méthode efficace pour la segmentation des mots arabes. Cette méthode est basée sur les probabilités de distance entre les mots et sous mots. Avant de décider la classe d'appartenance du montant littéral en entrée, il est nécessaire d'effectuer un certain nombre de traitements : binarisation, normalisation. Afin de caractériser nos images du montant littéral arabes, nous avons opté la matrice de distribution pour l'extraction des caractéristiques. Pour la classification des montants littéraux le classifieur machines à vecteurs de support (SVM : Support Vectors Machines) a été sélectionné. Dans ce cadre, nous avons utilisé une base de donnée standard CNEMPARMI composé de plus de 7000 images écrits par scripteurs différents. Cette base a été utilisée pour évaluer et tester les résultats de nos travaux.

Mots-clés

Reconnaissance automatique de l'écriture, Montant littéraux, SVM, Segmentation, apprentissage, classification,...

Table des matières

Remerciement	I
Dédicace	II
Résumé	III
Table des matières	IV
Liste de figures	VIII
Liste des tableaux	IX
Introduction générale.....	X
Chapitre I Reconnaissance d'écriture manuscrite.....	1
1.1. Introduction :	2
1.2. Différents aspects de reconnaissance de l'écriture :	2
1.2.1. La reconnaissance hors ligne:	2
1.2.2. La reconnaissance en ligne :	2
1.3. Caractéristiques de l'écriture arabe	3
1.4. Application	8
1.4.1. Hors ligne	8
1.4.1.1. Lecture des adresses postales	8
1.4.1.2. Domaine bancaire	8
1.4.1.3. Formulaires et bordereaux	9
1.4.2. En ligne	9
1.4.2.1. Ordinateurs sans clavier.....	9
1.5. Les phases d'un système de reconnaissance d'écriture:.....	9
1.5.1. Acquisition d'image.....	9
1.5.1.1. Acquisition hors ligne.....	10
1.5.1.2. Acquisition en ligne.....	10
1.5.2. Prétraitement et Normalisation	11
1.5.2.1. Seuillage	11
1.5.2.1.1. Seuillage global	11
1.5.2.1.2. Seuillage adaptatif	12
1.5.2.2. Binarisation.....	12
1.5.2.3. Transformation par érosion	12
1.5.2.4. Transformation par dilatation	13

1.5.2.5.	Squelette	14
1.5.2.6.	Normalisations.....	14
1.5.2.6.1.	Correction de l'inclinaison des lignes	14
1.5.2.6.2.	Correction de l'inclinaison des lettres	15
1.5.2.6.3.	La ligne de base	15
1.5.3.	Segmentation.....	16
1.5.3.1.	Segmentation de la page	16
1.5.3.2.	Segmentation d'un bloc de texte en lignes	17
1.5.3.3.	Segmentation des lignes en mots.....	17
1.5.3.4.	Segmentation des mots	18
1.5.4.1.	Approches de segmentation de document :	19
1.5.4.1.1.	Approche analytique explicite :	19
1.5.4.1.2.	Approche analytique implicite	19
1.5.4.1.3.	Approche analytique mixte	19
1.5.4.1.4.	Approche globale :	19
1.5.5.	Extraction des caractéristiques.....	20
1.5.5.1.	Les primitives structurelles et topologiques :	20
1.5.5.2.	Les primitives structurelles :	21
1.5.5.3.	Les primitives statistiques :	21
1.5.5.4.	Les primitives globales :	22
1.5.5.5.	Les primitives locales :	22
1.5.6.	Phase de classification	22
1.5.6.1.	Apprentissage	22
1.5.6.1.1.	Apprentissage supervisé	22
1.5.6.1.2.	Apprentissage non supervisé	23
1.5.6.1.3.	Reconnaissance et décision :	23
1.5.7.	Phase de post-traitement	23
1.6.	Méthodes de reconnaissances.....	23
1.6.1.	Méthodes globales	23
1.6.2.	Méthodes analytiques.....	24
1.7.	Conclusion :	24
	Chapitre II Machines à vecteurs de support.....	25
2.1.	Introduction :	26

2.2.	Principe de fonctionnement général :	26
2.2.1.	Linéarité et non linéarité :	27
2.2.1.1.	Cas linéairement séparable :	28
2.2.1.2.	Marge maximale de l'hyperplan :	29
2.2.1.3.	Cas non linéairement :	30
2.3.	Implémentation multi classe :	35
2.3.1.	Un contre tous (OAA : One Against All) :	35
2.3.2.	Un contre un (OAO : One Against One) :	37
2.4.	Les domaines d'applications des SVM :	39
2.5.	Les avantages et les inconvénients des SVM :	40
2.5.1.	Avantages :	40
2.5.2.	Inconvénients :	40
2.6.	Conclusion :	40
	Chapitre III Expérimentations et Résultats	41
3.1.	Introduction :	42
3.2.	Ressources matérielles et logicielles :	42
3.2.1.	Ressources matérielles	42
3.2.2.	Ressources logicielles	42
3.2.2.1.	Bref présentation de C++ Builder Embarcadero :	42
3.2.2.2.	Environnement de développement	43
3.3.	Le but de notre application :	44
3.4.	Les phases de notre system :	44
3.4.1.	L'acquisition des images :	44
3.4.2.	Prétraitement	45
3.4.3.	La Segmentation	46
3.4.4.	L'extraction des primitives :	49
3.4.4.1.	Construction de la matrice de distribution :	49
3.4.4.2.	Corpus :	50
3.5.	Expérimentations et Résultats	51
3.5.1.	Résultats de la segmentation	51
3.5.2.	Résultats de reconnaissance des montants littéraux :	52
3.5.3.	Résultats de reconnaissance des montants littéraux après la segmentation :	56
3.6.	Conclusion	56

Conclusion générale	57
Bibliographie	58

Liste des figures

Figure 1. 1 : Exemple de groupes de caractères ayant le même corps mais avec un nombre et / ou emplacement de diacritiques différents.	5
Figure 1. 2 : Des mots arabes avec leurs composantes connexes	6
Figure 1. 3 : Exemples de caractères avec boucles	7
Figure 1. 4 : Différents styles et fontes pour l'écriture arabe.	8
Figure 1. 5 : (a) Ecriture hors ligne (ensemble de pixels), (b) Ecriture en ligne (ensemble de coordonnées).	10
Figure 1. 6: Communication écriture homme-machine	11
Figure 1. 7 : Communication écriture homme-machine.	12
Figure 1. 8 : Image traitée par érosion.....	13
Figure 1. 9 : Image traitée par dilatation.	13
Figure 1. 10 : Correction de l'inclinaison des lignes à l'aide d'histogrammes de projection horizontale. Les pics les plus prononcés correspondent à la rotation correcte	14
Figure 1. 11 : Equation pour la rotation d'une image par angle θ	15
Figure 1. 12 : Inclinaison de l'écriture, et histogrammes de projection verticale correspondants.....	15
Figure 1. 13 : Histogramme de projection horizontale et bande de base.	16
Figure 1. 14 : Détection des différentes zones d'une page de document.	16
Figure 1. 15 : Segmentation de texte en lignes.	17
Figure 1. 16 : Segmentation de ligne en mots.	18
Figure 1. 17: Segmentation de mot en caractères.....	18
Figure 2. 1: Séparation de deux ensembles de points par un Hyperplan H.....	26
Figure 2. 2: Hyperplan optimal, marge et vecteurs de support	27
Figure 2. 3: linéarité et non linéarité	28
Figure 2. 4: linéarité et non linéarité	29
Figure 2. 5 : Données dans le cas non séparables.	31
Figure 2. 6: Transformation de l'espace de représentation et l'hyperplan séparateur dans le cas non linéairement séparables	32
Figure 2. 7: Fonction polynômiale	34
Figure 2. 8: Fonction RBF.....	34
Figure 2. 9 : Approche une-contre-reste avec des zones d'indécision	36

Figure 2. 10: Résolution des cas d'indécision dans la méthode 1vsR.....	37
Figure 2. 11: Approche une-contre-une	39
Figure 3. 1 : L'interface de C++ builder Embarcadero.....	43
Figure 3. 2: Des échantillons de base de données standard (a) et montants des chèques (b)...	45
Figure 3. 3: L'interface d'Acquisition et prétraitement des images	46
Figure 3. 4: Interface de la segmentation	47
Figure 3. 5 : L'organigramme de l'extraction de seuil d0.....	48
Figure 3. 6: L'organigramme de la segmentation	49
Figure 3. 7: Algorithme de Construction de la matrice de distribution.....	50
Figure 3. 8: Interface de création des corpus	51
Figure 3. 9: Exemples des erreurs sur la segmentation	52
Figure 3. 10: Le taux erreur et prédiction de la classification à partir des différents noyaux ..	53
Figure 3. 11: Matrice de confusion obtenue par le système, évalué sur la base de montants littéraux arabe manuscrits.....	54
Figure 3. 12: Matrice de confusion obtenue par le système, évalué sur la base de montants littéraux arabe manuscrits (suite).	54

Liste des Tableaux

Tableau 1. 1 : Comparaison des caractéristiques de différentes langues	3
Tableau 1. 2 : (a) les caractères arabes et leurs différentes formes, (b) les caractères additionnels « ؤ » et « ٱ », (c) et (d) Hamza et Madda et les positions qu'elles occupent en association avec Alif, Waw et Ya.	5
Tableau 1. 3 : Les quatre formes des caractères "ain" et "ha" en fonction de leurs positions dans la chaîne de caractères.	6
Tableau 3. 1: Résultats général de la segmentation	52
Tableau 3. 2 : Le taux de reconnaissance obtenu avec le différent noyau	53
Tableau 3. 3 : Matrice de confusion pour certains montants	55
Tableau 3. 4: Résultats général de la classification.....	56

Introduction générale

La reconnaissance automatique de l'écriture est attachée à la reconnaissance de caractères manuscrits et imprimés, la reconnaissance de mots et de texte, la reconnaissance du scripteur et la reconnaissance de documents. Elle relève du domaine de la reconnaissance des formes qui s'intéresse aux formes de l'écriture en plus des autres formes telles que : signatures, objets, etc. Le but est de transformer un texte écrit en une représentation compréhensible par une machine et facilement reproductible par un système informatique.

La reconnaissance de l'écriture concerne plus précisément toutes les tâches en relation avec la lecture et la rétro-conversion de documents papiers tels que le traitement des formulaires, l'indexation et l'archivage pour le catalogage numérique, le tri automatique du courrier, la lecture de chèques, des interfaces sans clavier, analyse du geste écrit, interaction avec le stylo électronique, etc.

La reconnaissance est dite en-ligne lorsque les données dynamiques sont acquises pendant l'écriture. On pense ici à une tablette graphique ou un papier électronique où l'utilisateur écrit avec un stylo. Par contre, on dit qu'elle est hors-ligne lorsqu'il s'agit de reconnaître l'image d'un mot obtenue avec un scanner.

L'objectif de cette mémoire est de proposer un système de reconnaissance de l'écriture manuscrite arabe hors-ligne. Ce système s'appuie sur une méthode de segmentation globale et utilise les Machines à Vecteurs de Support (SVM) dans la phase de classification.

Le premier chapitre est un rappel de certaines notions générales d'écriture manuscrite. Ainsi que les étapes nécessaires pour la réalisation d'un système de reconnaissance de l'écrit, ainsi une étude de reconnaissance de l'écriture manuscrite arabe. On citant l'état de l'art de la segmentation des textes dans le cas général. Pour celui nous décrivons le processus de l'étape de la détection des objets dans une page, à la segmentation des blocs de texte en lignes puis en mot puis en caractères. Nous mettons l'accent sur les méthodes utilisées dans ce type de segmentation.

Le deuxième chapitre va mettre l'accent sur la méthode de classification choisie, par l'étude de la méthode de Machines à Vecteurs de Support (SVM).

Le troisième chapitre constitue notre contribution, il s'agit d'un algorithme permettant la segmentation des mots arabes manuscrits, suivie dès les tests et résultats obtenus.

Nous terminons le travail par une conclusion sur les résultats obtenus par la méthode utilisée, et des perspectives de ce travail.

Chapitre I

Reconnaissance d'écriture manuscrite

1.1. Introduction :

Ces dernières années, des progrès considérables ont été réalisés dans la mise en œuvre de systèmes pour la reconnaissance de l'écriture manuscrite, et cela grâce, d'une part aux nombreux travaux effectués dans ce domaine, et, d'autre part, à la production évaluée à bas prix des micro-ordinateurs et des systèmes d'acquisition (scanner, tablette à digitaliser... etc.).

La reconnaissance de l'écriture manuscrite par ordinateur est un domaine très vaste, Les travaux de recherches sur l'écriture arabe, bien qu'ils soient moins nombreux en comparaison avec d'autres types d'écriture (ex. le Latin et le Japonais).

La reconnaissance de l'écriture manuscrite intéresse de nombreux domaines d'applications tels que le tri automatique du courrier, le traitement automatique des dossiers administratifs, ou encore l'enregistrement des chèques bancaires. Quelques systèmes de reconnaissance de l'écriture manuscrite ont été réalisés. Cependant, ils sont spécifiques à un domaine précis et sont encore limités.

1.2. Différents aspects de reconnaissance de l'écriture :

1.2.1. La reconnaissance hors ligne:

Il s'agit de reconnaître des textes manuscrits à partir de documents écrits au préalable. L'image du texte écrit est numérisé à l'aide d'un scanner, les informations recueillies se présentent sous la forme d'une image discrète constituée d'un ensemble de pixels. L'écriture prend l'aspect d'un signal spatial bidimensionnel numérisé [1].

1.2.2. La reconnaissance en ligne :

Il s'agit de reconnaître l'écriture au fur et à mesure de son tracé. Le texte est saisi avec un stylo et une tablette à numériser, les informations recueillies sont constituées par une suite ordonnée de points (définis par leurs coordonnées) échantillonnés à cadence fixe. L'écriture prend l'aspect d'un couple de signaux temporels numérisés.

La reconnaissance hors ligne ne peut pas a priori s'appuyer sur l'information temporelle du tracé qui est perdue, mais elle peut tenir compte de l'épaisseur du tracé (les pleins et les déliés).

La reconnaissance en ligne peut disposer de l'information temporelle (vitesse, accélération, levés de stylo, retours en arrière, barres de t, points diacritiques), mais d'aucune information sur l'épaisseur du tracé si on ne dispose pas d'un signal de pression de la pointe du stylet sur le support [1].

1.3. Caractéristiques de l'écriture arabe

L'écriture arabe imprimée ou manuscrite possède des caractéristiques différentes d'autres langues en structure et en mode de liaison entre les caractères formant un mot, ce qui rend l'application directe des techniques de reconnaissance développées pour les caractères chinois ou latins, par exemple, une tâche délicate pour leur reconnaissance. La comparaison des diverses caractéristiques du manuscrits : arabe, latin, hindou est décrite dans le tableau suivant:





Caractéristiques	Arabe	Latin	Hindou
Justification		G à D	G à D
Cursivité		Non	Oui
Diacritiques		Non	Oui
Nombre de voyelles (vowels)	2	5	
Nombre de lettres	28	26	40
Formes des lettres		2	1

Tableau 1. 1 : Comparaison des caractéristiques de différentes langues [2].

L'écriture arabe est à la base de nombreuses autres langues telles que le Perse et certains dialectes d'Afrique, d'Inde, d'Indonésie, de Chine et de Turquie [3]. L'écriture arabe utilise 28 lettres (Tableau 1.2) auxquelles il faut ajouter « El hamza ء » qui est le plus souvent considéré comme un caractère à part entière [2]. El hamza 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 lettres "alif, waw, ya". De plus, l'alphabet arabe comprend d'autres caractères additionnels tels que « ؤ » et « ڤ ». La considération du symbole « ~ » qui s'écrit uniquement sur le support du caractère « ٲ », fait apparaître d'autres graphismes.

Caractère	Initiale	Milieu	Finale	Isolé
Alif			ﻝ	ﺍ
Ba	ﺏ	ﺏ ﺏ	ﺏ	ﺏ
Ta	ﺕ	ﺕ	ﺕ	ﺕ

Tha	ث	ث	ث	ث
Jim	ج	ج	ج	ج
Ha	ح	ح	ح	ح
Kha	خ	خ	خ	خ
Dal			د	د
Thé			ذ	ذ
Ra			ر	ر
Za			ز	ز
Sin	س	س	س	س
Chin	ش	ش	ش	ش
Sad	ص	ص	ص	ص
Da	ض	ض	ض	ض
Thad	ط	ط	ط	ط
Zad	ظ	ظ	ظ	ظ
Ain	ع	ع	ع	ع
Ghin	غ	غ	غ	غ
Fa	ف	ف	ف	ف
Qaf	ق	ق	ق	ق
Kaf	ك	ك	ك	ك
Lam	ل	ل	ل	ل
Mim	م	م	م	م
Nun	ن	ن	ن	ن
Ha	ه	ه	ه	ه
Waw	و	و	و	و
Ya	ي	ي	ي	ي
Alif + ~			آ	آ
			إ	إ

Alif + ء			أ	إ
Waw+ء			و	ؤ
Ya+ ء		ي	ى	ئ
Ta marbouta				ة
Lamalif			لا	لا
Lamalif +~			لا	لا
Lamalif + ء			لا	لا
			لا	لا

Tableau 1. 2 : (a) les caractères arabes et leurs différentes formes, (b) les caractères additionnels « ء » et « لا », (c) et (d) Hamza et Madda et les positions qu'elles occupent en association avec Alif, Waw et Ya.

L'écriture arabe a ainsi plusieurs spécificités, nous citons :

- Un trait caractéristique de l'écriture arabe est la présence d'une ligne de base horizontale dite ligne de *référence* ou *d'écriture*. C'est le lieu de ligatures horizontales des caractères d'une même chaîne ;
- Les caractères arabes s'écrivent cursivement de la droite vers la gauche, aussi bien dans le cas de l'imprimé que du manuscrit ;
- L'arabe contient 28 caractères de base, dont 16 incluent dans leurs formes des points diacritiques qui peuvent être au nombre de 1, 2 ou 3. Ces points font la différence entre les caractères ayant un corps identique, ils peuvent être situés au-dessus ou au-dessous du corps du caractère de base ou au milieu (figure 1.1) :

(ظ ط) - (ج خ ح) - (ي ث ت ب)
(ه ة) (د ذ) (زر) (س ش)

Figure 1. 1 : Exemple de groupes de caractères ayant le même corps mais avec un nombre et / ou emplacement de diacritiques différents.

- La forme d'une lettre écrite dépend de son contexte, elle diffère selon que le caractère apparaît en position initiale, médiane ou isolée dans une chaîne de caractère (Tableau 1.3). A chaque lettre peut correspondre jusqu'à quatre formes différentes. Les formes correspondantes à un même caractère, souvent appelées "formes internes", présentent parfois de sensibles différences ; dans certain cas, il est même difficile d'en déduire

qu'il s'agit de la même lettre. Cependant le codage ASMO¹ attribue un seul code pour les différentes formes d'un même caractère, contrairement au latin où le code ASCII² prévoit deux codes différents pour la même lettre dans sa forme majuscule et minuscule.

Initial	Médiane	Finale	Isolée
علم	معلم	سمع	ورع
همس	مهد	لعبه	منتزه

Tableau 1. 3 : Les quatre formes des caractères "ain" et "ha" en fonction de leurs positions dans la chaîne de caractères.

- L'écriture arabe est par nature semi cursive, ce qui rend la phase de segmentation une opération cruciale dans un système OCR arabe. Cependant certains caractères de l'alphabet arabe ne peuvent pas être attachés à leurs successeurs dans le mot. De ce fait, la présence d'une de ces lettres peut diviser le mot en deux composantes connexes (une partie d'un mot comprenant un ou plusieurs caractères reliés). La figure suivante montre trois mots arabes : la premier (a) avec un seule composante connexe de neuf lettres, le deuxième (b) avec deux composantes connexes, une de trois lettres et l'autre avec une seule lettre, le troisième mot est composé de cinq composantes connexes ;

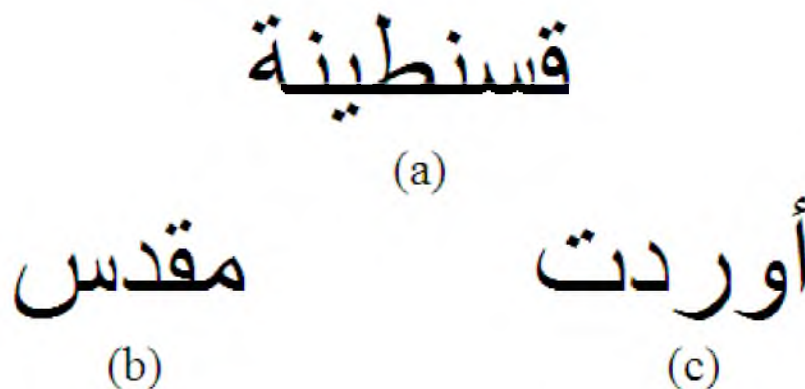


Figure 1. 2 : Des mots arabes avec leurs composantes connexes

- Les caractères arabes peuvent être voyellés. Les voyelles peuvent se placer au-dessus ou au-dessous du caractère. Dans l'arabe contemporain ordinaire, on écrit seulement les consonnes et les voyelles longues ; le lecteur doit déduire les voyelles courtes en utilisant le contexte. Les mêmes consonnes avec des voyelles courtes différentes

¹ ASMO :Arabic Standard Metrology Organization

² American Standard Code For Information Interchange

peuvent être comprises comme verbe, nom ou adjectif. A titre d'exemple, « علم » peut signifier « drapeau : عَلَم » ou « savoir : عِلْم » ou encore « enseigner : عَلَّمَ », selon sa voyellation. Il existe 8 signes qui peuvent se placer au-dessus de la ligne d'écriture, tels que Fathah (َ), Dhammah (ُ), Chaddah (ّ) et, Soukoun (◌), ou au-dessous tels que Kasrah (ِ). En plus « tanwin » qui peut être formé à partir d'un double fatha(ً), d'un double dhamma (ٌ), ou d'un double Kassra (ٍ),

- Les caractères arabes ne possèdent pas une taille fixe (hauteur et largeur). Leur taille varie d'un caractère à un autre et d'une forme à une autre pour un même caractère.
- Certains caractères arabes incluent une boucle qui peut avoir différentes formes (figure 1.3)

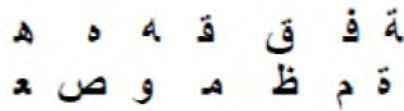


Figure 1. 3 : Exemples de caractères avec boucles

- Dans certaines fontes plusieurs caractères peuvent être écrits de façon combinée. Les combinaisons ou *ligature* verticales sont utilisées pour des raisons d'esthétique [3]. Elles peuvent être formées de deux, trois ou quatre caractères. On parle souvent de ligature de niveau n pour désigner le nombre « n » de caractères ligaturés ;
- L'écriture arabe est une écriture calligraphique, elle varie selon les milieux et les régions, d'une extrême simplicité formelle à la complexité exhaustive de l'arabesque.
- Il existe une centaine de styles dont seulement quelques-uns sont couramment utilisés dans le monde arabo-musulman, nous citons par exemple : Tholothi, Neskhi, Requeh, Dewani, Farci et Koufique. La Figure (1.4) suivante montre un exemple de différents styles graphiques de l'écriture arabe.



Figure 1. 4 : Différents styles et fontes pour l'écriture arabe.

1.4. Application

Grâce à la particularisation des problèmes, la reconnaissance de l'écriture a permis le développement d'applications particulières et efficaces concernant la reconnaissance de l'écriture hors ligne et en ligne.

1.4.1. Hors ligne

La reconnaissance de l'écriture hors ligne connaît un essor important dans les domaines associés au développement d'intérêts économiques.

1.4.1.1. Lecture des adresses postales

La lecture des codes postaux manuscrits associés à la lecture des noms des villes a permis d'étendre le développement des machines de tri automatique du courrier (lettres et objets plats).

1.4.1.2. Domaine bancaire

La reconnaissance des montants littéraux manuscrits est associée à la reconnaissance des montants numériques pour valider la lecture des chèques. La reconnaissance des chèques peut être associée à celle des coupons correspondants. Des machines capables de lire plusieurs milliers de chèques à l'heure sont déjà en service actuellement.

La reconnaissance des montants numériques a permis la création des guichets automatiques de remise de chèques bancaires. Dans ces automates, le client, identifié par sa carte bancaire, indique sur le clavier le montant du chèque. Si le montant coïncide avec le montant numérique reconnu, l'encaissement du chèque est immédiatement validé.

1.4.1.3. Formulaires et bordereaux

Il s'agit principalement des applications de la reconnaissance des caractères pré-casés manuscrits à la lecture des formulaires de sondage, les bordereaux de commande, les constats d'assurance.

1.4.2. En ligne

La reconnaissance de l'écriture en ligne connaît un essor important dans les domaines associés à une communication homme-machine naturelle, conviviale et ergonomique ainsi qu'à l'informatique nomade.

1.4.2.1. Ordinateurs sans clavier

La reconnaissance de l'écriture en ligne a pour ambition de remplacer le clavier et la souris de l'ordinateur par un stylo. Cette modification a pour but de rendre les ordinateurs plus conviviaux en permettant de les utiliser dans des situations très diversifiées (prises de notes, rédaction de commandes, de constats d'accidents, enseignement de l'écriture, etc.). Il s'agit, d'une part, de rendre l'informatique nomade.

La reconnaissance en ligne de l'écriture manuscrite présente également un intérêt fondamental lorsque le nombre de caractères à saisir et à reconnaître est très grand, c'est notamment le cas pour tous les textes en caractères orientaux : chinois, japonais, coréen...etc [1].

1.5. Les phases d'un système de reconnaissance d'écriture:

Un système de reconnaissance fait appel généralement aux étapes suivantes: Acquisition, prétraitements, segmentation, extraction des caractéristiques, classification, suivis éventuellement d'une phase de post-traitement.

1.5.1. Acquisition d'image

L'acquisition hors ligne et l'acquisition en ligne de sont deux modes différents d'un système de reconnaissance des mots, ayant chacun ses outils propres d'acquisition et ses algorithmes correspondants de reconnaissance.

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

1.5.1.1. Acquisition hors ligne

Dans le cas hors ligne, il s'agit de reconnaître des textes manuscrits à partir de documents écrits au préalable. L'image du texte écrit est numérisée à l'aide d'un scanner, les informations recueillies se présentent sous la forme d'une image discrète constituée d'un ensemble de pixels Figure 1.5 a.

1.5.1.2. Acquisition en ligne

Dans le cas en ligne, il s'agit de reconnaître l'écriture au fur et à mesure de son tracé. Le texte est saisi avec un stylo et une tablette à numériser, les informations recueillies sont constituées par une suite ordonnée de points (définis par leurs coordonnées) échantillonnés à cadence fixe Figure 1.5 b.

La reconnaissance hors ligne ne peut pas a priori s'appuyer sur l'information temporelle du tracé qui est perdue, mais elle peut tenir compte de l'épaisseur du tracé (les pleins et les déliés).

La reconnaissance en ligne peut disposer de l'information temporelle (vitesse, accélération, levés de stylo, retours en arrière, barres de t, points diacritiques), mais d'aucune information sur l'épaisseur du tracé si on ne dispose pas d'un signal de pression de la pointe du stylet sur le support.

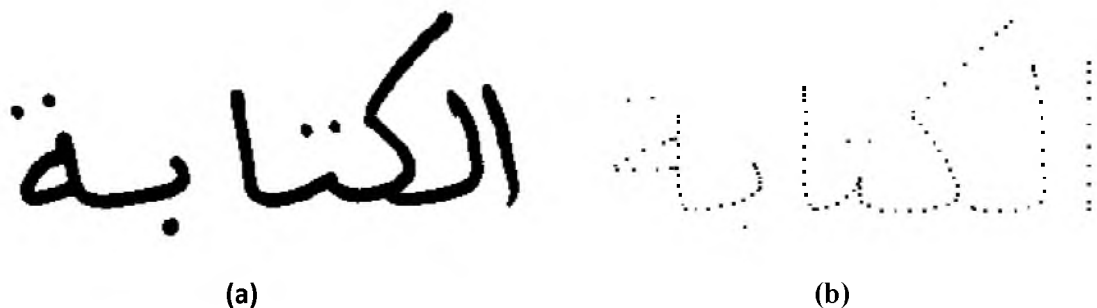


Figure 1. 5 : (a) Ecriture hors ligne (ensemble de pixels), (b) Ecriture en ligne (ensemble de coordonnées).

La figure 1.6 représente un exemple typique de la communication écriture homme-machine

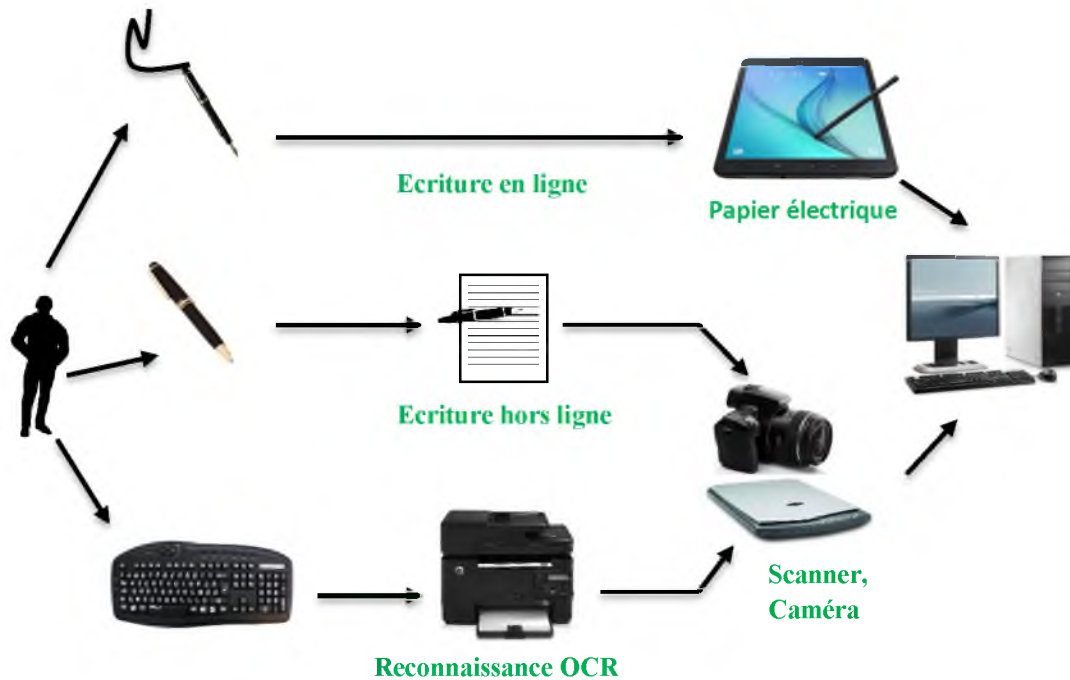


Figure 1. 6: Communication écriture homme-machine

1.5.2. Prétraitement et Normalisation

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.

1.5.2.1. Seuillage

On distingue en général deux approches : les approches à seuillage global, et les approches à seuillage adaptatif [5].

Pour une évaluation de différentes méthodes de Binarisation, on pourra consulter le travail [6].

1.5.2.1.1. Seuillage global

Le seuillage global consiste à prendre un seuil ajustable, mais identique pour toute l'image. Chaque pixel de l'image est comparé à ce seuil et prend la valeur blanc ou noir selon qu'il est supérieur ou inférieur. Cette classification ne dépend alors que du niveau de gris du pixel considéré.

Cette méthode convient pour les documents simples et de bonne qualité. Néanmoins, elle n'est plus applicable lorsque la qualité d'impression du texte n'est pas constante dans toute la page, des caractères peuvent être partiellement perdus. Des problèmes surviendront

également si le fond est bruité ou non homogène, dans ce cas des taches parasites peuvent apparaître.

Les histogrammes de la répartition des niveaux de gris des pixels de ces images contiennent deux pics nets : l'un pour la forme, l'autre pour le fond. Un seuil global peut aisément être sélectionné entre les deux pour classer les pixels selon qu'ils appartiennent aux formes ou au fond.

1.5.2.1.2. Seuillage adaptatif

Dans les documents pour lesquels l'intensité du fond et l'intensité de la forme peuvent varier au sein du document, un seuillage global est inadapté. Il devient nécessaire de choisir le seuil de Binarisation de manière locale. On calcule un seuil de Binarisation pour chaque pixel de l'image, en fonction de son voisinage.

1.5.2.2. 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 en niveaux de gris ou en couleur, a une image binaire. Toutes ces techniques sont basées sur le principe de seuillage comme le montre l'équation suivante :

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

Figure 1. 7 : Communication écriture homme-machine.

$I_n(x, y)$ 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 [4].

1.5.2.3. 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, 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) [4].

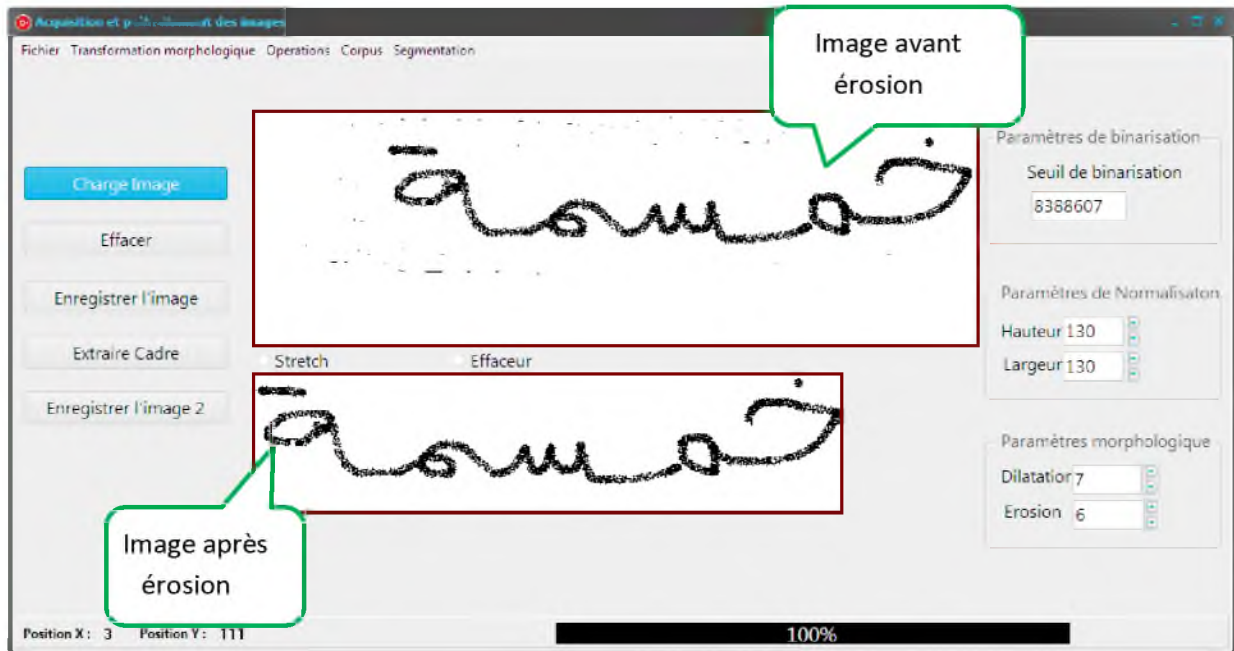


Figure 1. 8 : Image traitée par érosion.

1.5.2.4. 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 noir, alors on affecte à ce pixel la couleur noir ($p(x,y)=0$).

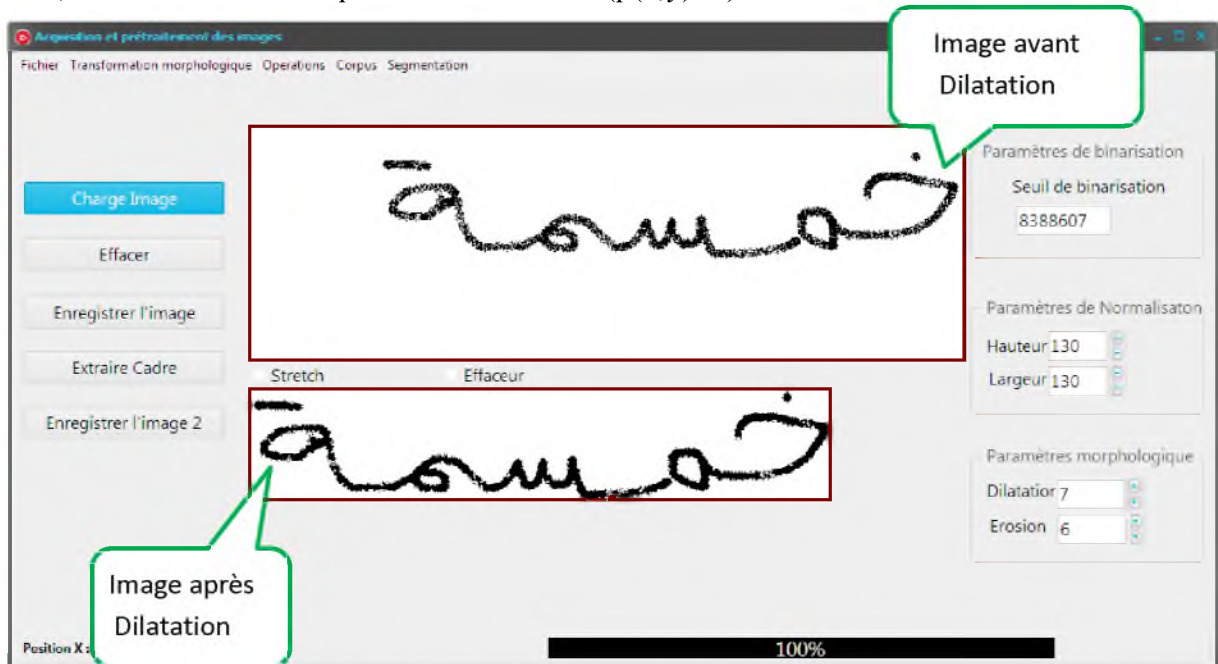


Figure 1. 9 : Image traitée par dilatation.

1.5.2.5. Squelette

La squelettisation est une opération qui permet de passer d'une image à sa représentation en "fil de fer". Le squelette a un pixel d'épaisseur. C'est une manière de représenter l'information indépendamment de l'épaisseur initiale de l'écriture.

Il permet d'extraire des caractéristiques importantes, comme les intersections et le nombre de tracés, leurs positions relatives. Il est également possible de renormaliser l'épaisseur de l'écriture à partir du squelette [9].

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

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

1.5.2.6. Normalisations

Les normalisations ont pour objectif de rendre l'écriture la plus indépendante possible du scripteur. Nous présenterons ici trois techniques de normalisation :

1.5.2.6.1. Correction de l'inclinaison des lignes

La correction de l'inclinaison des lignes de texte (également appelée correction de "skew"), consiste à redresser horizontalement les lignes d'écriture obliques.

Plusieurs méthodes sont disponibles. Les deux plus populaires sont la transformée de Hough (appliquée sur les centre de gravité des composantes connexes), et les histogrammes de projection.

La méthode de l'histogramme consiste à parcourir l'image selon des directions d proches de l'horizontale, et à compter le nombre de pixels noirs selon ces directions pour chaque ligne [5] (voir Figure 1.10).

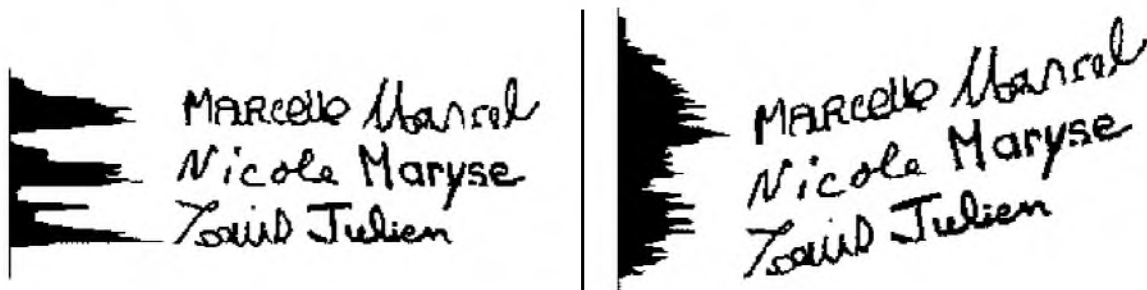


Figure 1. 10 : Correction de l'inclinaison des lignes à l'aide d'histogrammes de projection horizontale. Les pics les plus prononcés correspondent à la rotation correcte [10].

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

Pour corriger cette inclinaison, il suffit d'appliquer une rotation de l'image d'angle θ [5]:

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

Figure 1. 11 : Equation pour la rotation d'une image par angle θ

1.5.2.6.2. Correction de l'inclinaison des lettres

L'inclinaison des caractères est définie comme étant l'angle entre l'axe correspondant à la direction moyenne des caractères et l'axe vertical (l'angle α , figure 1.11). L'objectif de ce prétraitement est de transformer le mot de façon à ce que cet axe de direction principale devient vertical. Ceci permet une réduction considérablement de la variabilité de l'écriture et une amélioration de la qualité de la segmentation des mots en caractères [7] [8].

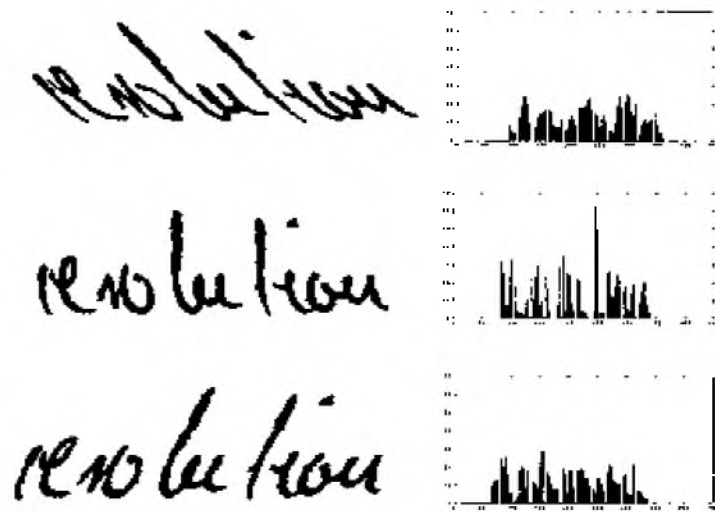


Figure 1. 12 : Inclinaison de l'écriture, et histogrammes de projection verticale correspondants.

1.5.2.6.3. La ligne de base

Les lignes de base encadrent la bande des minuscules et délimitent les zones contenant les ascendants et descendants. Ces lignes sont importantes en reconnaissance de l'écriture. Cette information est utilisée pour détecter ascendants et descendants, et également pour normaliser les primitives, les rendant ainsi moins dépendantes de la hauteur de l'écriture.

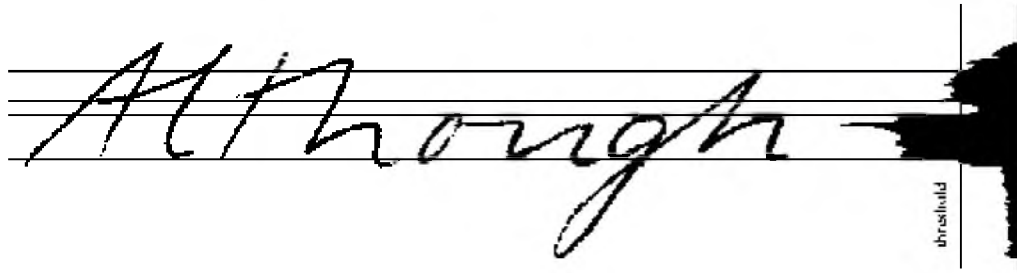


Figure 1.13 : Histogramme de projection horizontale et bande de base.

1.5.3. Segmentation

Dans cette phase les différentes parties logiques d'une image sont extraites. A partir d'une image acquise il y'a d'abord séparation des blocs de texte et des blocs graphiques, puis à partir d'un bloc de texte il y'a extraction des lignes, ensuite à partir de ces lignes sont extraits les mots puis les caractères ou graphèmes [11].

1.5.3.1. Segmentation de la page

Cette étape permet de localiser dans chaque page, les zones d'information conformément à leur apparence physique. Elle est associée généralement à l'étiquetage logique qui consiste à déterminer la nature du media représenté dans chaque zone (texte, graphique, photographie etc.).

Cette classification permet ensuite d'orienter la reconnaissance vers des systèmes spécialisés dans l'analyse de chaque type de media [12].

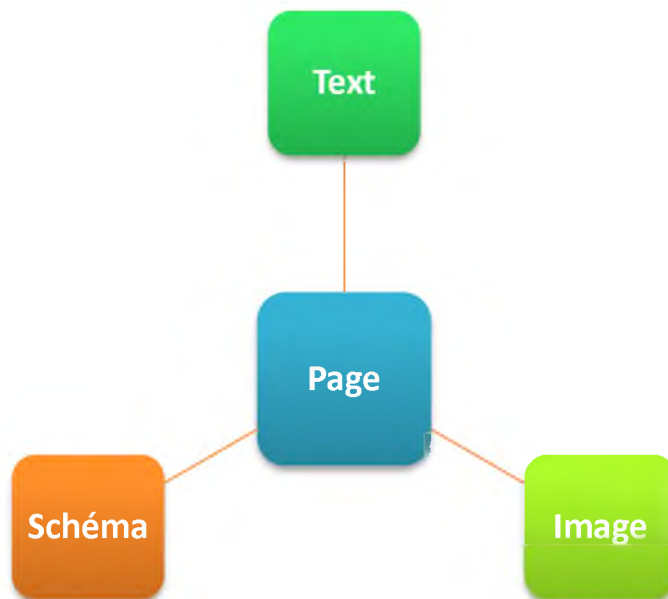


Figure 1.14 : Détection des différentes zones d'une page de document.

1.5.3.2. Segmentation d'un bloc de texte en lignes

Cette étape consiste à séparer les différentes lignes du texte pour en extraire les mots puis les caractères composants les mots. La plupart des études proposées dans ce domaine s'appuient sur une décomposition de l'image en composantes connexes [13].

D'autres par contre utilisent des techniques s'appuyant en grande partie sur les histogrammes de la projection horizontale [11]. Et certains auteurs optent pour des méthodes spécialisées telle que celle utilisée par Bennasri et Al dans [13] pour la segmentation en lignes de l'écriture arabe manuscrite.

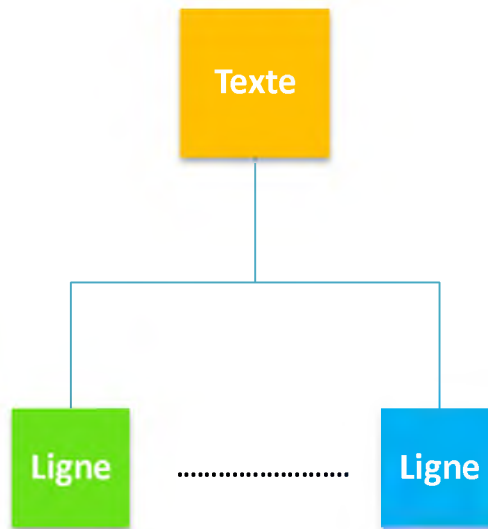


Figure 1. 15 : Segmentation de texte en lignes.

1.5.3.3. Segmentation des lignes en mots

La segmentation en mots est réalisée en déterminant l'histogramme des projections verticales des lignes pour détecter les espaces entre les mots et pouvoir les séparer. Cependant cette technique peut ne pas être efficace dans certains cas où les mots se chevauchent (cas par exemple de l'écriture arabe). Dans ce cas d'autres techniques sont utilisées telles que : le suivi du contour, détermination des squelettes ou la détermination des composantes connexe.

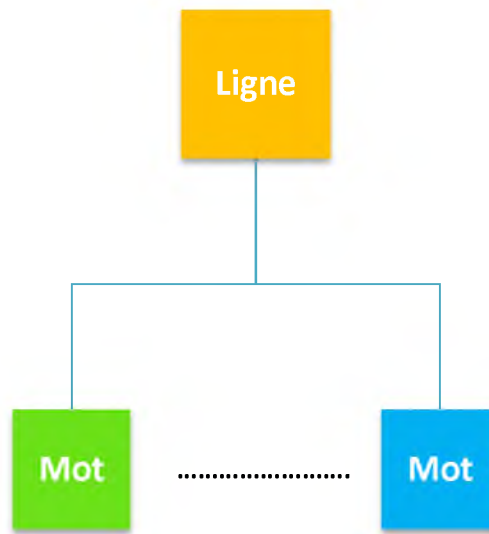


Figure 1. 16 : Segmentation de ligne en mots.

1.5.3.4. Segmentation des mots

La segmentation des caractères est une opération qui tente de décomposer une image de séquence de caractères (mot) en sous-images de symboles individuels. C'est l'un des processus de décision dans un système de reconnaissance optique de caractères. Son but est de décider si un motif isolé d'une image (caractère ou autre entité identifiable du mot) est correct ou non [14].

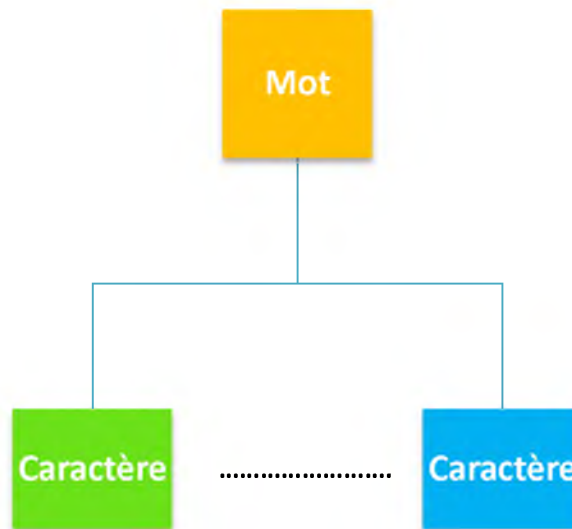


Figure 1. 17: Segmentation de mot en caractères.

1.5.4. Différentes approches et méthodes de segmentation :

La segmentation est une technique très importante dans le domaine du traitement d'image, en particulier les documents. La segmentation de documents dépend des caractéristiques de l'écriture et de l'approche de reconnaissance. Il existe différentes approches et différentes méthodes de segmentation de documents.

1.5.4.1. Approches de segmentation de document :

Il existe principalement trois stratégies de segmentation : L'approche analytique explicite, l'approche implicite et l'approche globale [51].

1.5.4.1.1. Approche analytique explicite :

Cette approche repose sur la segmentation et la reconnaissance d'éléments caractéristiques proches des lettres (qu'on appelle graphème). Il y a dans ce cas une pré-segmentation explicite qui s'appuie souvent sur des critères perceptifs tels que l'emplacement des ligatures entre les lettres. Les graphèmes sont reconnus individuellement. L'identification du mot est effectuée s'appuie sur l'information contextuelle [50].

1.5.4.1.2. Approche analytique implicite

L'approche analytique implicite évite certains résultats de la segmentation explicite. Cette fois, il n'y a pas de pré-segmentation du mot. La segmentation s'effectue pendant la reconnaissance. Le système cherche dans l'image les composantes ou les groupes de graphèmes qui correspondent à ses classes de lettres. Classiquement, il peut le faire de deux manières ; soit par fenêtrage, soit par recherche de primitives [50].

1.5.4.1.3. Approche analytique mixte :

La plupart des méthodes actuelles pratiquent un mélange des deux stratégies précédentes. D'abord, une phase de pré-segmentation est appliquée pour sur-segmenter l'image. Puis la meilleure combinaison de reconnaissance est choisie en considérant les combinaisons possibles de $1, 2, \dots, n$ graphèmes. Cette recherche est faite au moyen de méthodes d'optimisation telles que la programmation dynamique ou les modèles de Markov Cachés [51].

1.5.4.1.4. Approche global :

Dans l'approche globale, le mot est reconnu par sa forme générale. Il n'y a pas de segmentation (ni explicite, ni implicite). C'est pourquoi cette approche est censée être robuste au bruit ou aux imperfections du signal. Habituellement la reconnaissance dépend d'un lexique [51]. Puisque cette méthode nécessite un modèle pour chaque mot du lexique et que chaque modèle doit être appris, elle est plutôt utilisée pour des applications où le lexique est restreint comme c'est le cas dans le traitement automatique des chèques [41].

1.5.5. Extraction des caractéristiques

La phase d'extraction des primitives (encore appelées caractéristiques) est une étape très importante pour un système de reconnaissance de l'écriture. L'objectif de cette étape est la sélection des informations les plus pertinentes pour une tâche de classification donnée.

Cependant, la problématique de cette étape a pour origine le risque de perte d'informations significatives.

Dans la littérature, plusieurs méthodes d'extraction de primitives pertinentes ont été appliquées pour la reconnaissance de l'écriture. Aussi, les primitives utilisées sont généralement classées selon trois façons différentes [15].

Une première distinction est faite entre les primitives selon qu'elles sont extraites d'une image en niveaux de gris, d'une image binarisée, du contour ou du squelette de la forme résultant de la phase de prétraitement [16], [17]. En effet :

- Dans le cas d'une image en niveaux de gris, les primitives extraites sont : des primitives du type zoning, moments géométriques ou de Zernike, des primitives discrètes : largeur, hauteur, épaisseur moyenne des traits, etc ;
- Dans le cas d'une image binarisée, les primitives extraites sont : des composantes connexes [18], des projections, des histogrammes, des primitives du type zoning et des primitives discrètes ;
- Dans le cas du squelette, les primitives extraites sont : des descripteurs de Fourier, des descriptions de graphe ou des primitives discrètes (nombres de boucles, de croisements, rapport largeur sur hauteur, présence d'un point isolé, etc.) ;

Une deuxième distinction est effectuée entre les primitives topologiques, structurelles ou statistiques [16] ;

Une troisième distinction est effectuée entre les primitives globales et les primitives locales [23].

Dans la suite, nous présentons les principales caractéristiques structurelles et topologiques, les caractéristiques statistiques ainsi que les caractéristiques globales et locales.

1.5.5.1. Les primitives structurelles et topologiques :

L'auteur dans [24], [25], [26], [27], [28] décrivent les propriétés topologiques et géométriques de l'écriture. Il s'agit de compter dans l'image brute, ou à partir du squelette ou du contour de la forme :

- ✓ les segments de droite et leurs attributs (position, orientation, ...) ;
- ✓ la hauteur et la largeur du caractère ;

- ✓ les positions relatives entre segments ;
- ✓ les arcs, boucles et concavités, mesures de pentes et autres paramètres de courbures pour évaluer des orientations principales ;
- ✓ les points d'extrémités, de croisements et de jonctions des traits ;
- ✓ les angularités, points extremums et points terminaux ;
- ✓ le nombre de points diacritiques et leur position par rapport à la ligne de base de l'écriture ;
- ✓ autres paramètres tels que: la longueur et l'épaisseur des traits, le nombre de trous, les surfaces et les périmètres.

1.5.5.2. Les primitives structurelles :

Constituent des caractéristiques très informatives et discriminantes. Elles permettent de prendre des décisions rapides dans la reconnaissance de l'écriture avec une complexité de calcul modérée pendant l'extraction, en comparaison aux caractéristiques statistiques.

1.5.5.3. Les primitives statistiques :

Décrivent une forme en un ensemble de mesures statistiques extraites à partir de cette forme [29], [31]. La raison principale de l'utilisation des primitives statistiques est de donner des informations locales sur le contenu de l'écriture. Il s'agit par exemple de :

- ✓ le profil de projection des densités de pixels noir/blanc qui représente le nombre de pixels sur chaque ligne ou chaque colonne de l'image ;
- ✓ l'histogramme directionnel permettant de compter le nombre de pixels sur une ligne dans une direction quelconque de l'image ;
- ✓ l'histogramme des transitions qui permet de retenir le nombre des transitions 0-1 et 1-0 entre pixels ;
- ✓ les moyennes locales de pixels de l'image situés à l'intérieur d'un masque rectangulaire (Zoning) ;
- ✓ les directions des contours dans une fenêtre locale ;
- ✓ les moments invariants qui sont des caractéristiques intéressantes, car elles sont invariantes en translation, taille et rotation. Ce sont des mesures statistiques de la distribution des pixels autour du centre de gravité du caractère. Ils ont été initialement appliqués à la reconnaissance de l'écriture latine et arabe, puis récemment ont été appliqués aux caractères [32].

Avec ce type de caractéristiques, aucune interprétation directe n'est faite sur le contenu de l'écriture. Ainsi, ces caractéristiques se trouvent moins fortes que les primitives structurelles.

Du fait que ces caractéristiques portent peu d'information sur l'écriture, un système de reconnaissance doit les gérer efficacement [30].

1.5.5.4. Les primitives globales :

Cherchent à représenter au mieux la forme générale d'un caractère et sont donc calculées sur des images relativement grandes. Citons par exemple la transformée de Fourier qui détecte les lignes dans les images [33].

1.5.5.5. Les primitives locales :

Sont calculées lors d'un parcours des pixels de l'image avec un pas d'analyse qui dépend de la modélisation, du type de primitive et de la taille de l'image.

Parmi ces primitives, citons par exemple la transformée de Fourier fenêtrée et les dérivées qui extraient une information sur la direction des traits et les moments invariants [34].

1.5.6. Phase de classification

La classification regroupe les deux tâches d'apprentissage et de décision qui jouent des rôles assez proches dans les systèmes de reconnaissance des écritures manuscrite. Le résultat de l'apprentissage est, soit la réorganisation ou le renforcement des modèles existants en tenant compte de l'apport de la nouvelle forme, soit la création d'un nouveau modèle représentant la forme entrée. Le résultat de la décision est un « avis » sur l'appartenance ou non de la forme aux modèles de l'apprentissage. Nous allons revenir dans la suite sur la fonction de ces tâches en développant quelques approches classiques qui les réalisent [35].

1.5.6.1. Apprentissage

Dans le cas d'apprentissage il s'agit en fait de fournir au système un ensemble de formes qui sont déjà connues (on connaît la classe de chacune d'elles). C'est cet ensemble d'apprentissage qui va permettre de « régler » le système de reconnaissance de façon à ce qu'il soit capable de reconnaître ultérieurement des formes de classes inconnues.

Il existe deux types d'apprentissage, supervisé et non supervisé [35].

1.5.6.1.1. Apprentissage supervisé

L'apprentissage est dit supervisé si les différentes familles des formes sont connues a priori et si la tâche d'apprentissage est guidée par un superviseur ou professeur, c'est-à-dire le concepteur, indique, pour chaque forme échantillon rentrée, le nom de la famille qui la contient. La tâche d'apprentissage tente de conserver ses liens de parenté en répartissant les familles dans des classes séparées entre elles [36].

1.5.6.1.2. Apprentissage non supervisé

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].

1.5.6.1.3. Reconnaissance et décision :

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

La reconnaissance peut conduire à un succès si la réponse est unique (un seul modèle répond à la description de la forme du caractère). Elle peut conduire à une confusion si la réponse est multiple (plusieurs modèles correspondent à la description). Enfin elle peut conduire à un rejet de la forme si aucun modèle ne correspond à sa description. Dans les deux premiers cas, la décision peut être accompagnée d'une mesure de vraisemblance, appelée aussi score ou taux de reconnaissance [36].

1.5.7. 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 mot). Cette phase est souvent implémentée comme un ensemble d'outils relatifs à la fréquence d'apparition des mots dans une chaîne, aux lexiques et à d'autres informations contextuelles.

Comme la classification peut aboutir à plusieurs candidats possibles, le posttraitement a pour objet d'opérer une sélection de la solution en utilisant des niveaux d'informations plus élevés (syntaxiques, lexicale, sémantiques...). 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.

1.6. Méthodes de reconnaissances

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

1.6.1. Méthodes globales

Les méthodes globales de reconnaissance des mots sont a priori séduisantes puisqu'elles ne rencontrent pas les difficultés liées aux ambiguïtés provenant de la segmentation. Toutefois, elles ne peuvent se concevoir que dans le cas d'un vocabulaire limité, ou dans le cas de vocabulaires de taille plus grande mais limitée dynamiquement.

Elles reposent sur une caractérisation globale du mot [1].

1.6.2. Méthodes analytiques

Contrairement aux méthodes de reconnaissance globales, les méthodes analytiques cherchent à identifier les graphèmes issus de la segmentation (fragments de lettres, des lettres ou des regroupements de lettres) pour reconstituer les mots. Elles présentent l'intérêt de pouvoir se généraliser à la reconnaissance d'un vocabulaire étendu ou limité dynamiquement, à partir d'une phase d'apprentissage des classes de graphèmes [1].

1.7. Conclusion :

L'écriture arabe est une écriture semi-cursive qui présente l'avantage d'une liaison favorisant la fluidité et la rapidité du geste.

Mais une des premières difficultés est la variabilité d'écriture entre scripteurs. La difficulté surmontée pour le traitement du caractère est sa forme et sa position dans le mot.

La reconnaissance de l'écriture arabe manuscrite est en soi une tâche qui est fondamentalement plus complexe que la reconnaissance de l'écriture latine manuscrite cursive à cause des spécificités de l'écriture arabe qui présente un certain nombre de difficultés supplémentaires.

Chapitre II



Machines à vecteurs de support

2.1. Introduction :

Parmi les méthodes à noyaux, inspirées de la théorie statistique de l'apprentissage de Vladimir Vapnik, les Machines à Vecteurs de Support (SVM) constituent la forme la plus connue. SVM est une méthode de classification binaire par apprentissage supervisé, elle fut introduite par Vapnik en 1995. Cette méthode est donc une alternative récente pour la classification.

Dans ce chapitre, nous abordons les aspects théoriques des machines à vecteurs de support.

Nous décrivons principale générale de l'utilisation des machines à vecteurs de support pour la classification. Nous y trouvons deux descriptions distinctes : la première traite le cas de données séparables alors qu'une version modifiée permet de considérer des données non séparables. L'extension au cas non linéaire est décrite par la suite en présentant différents algorithmes d'apprentissage du SVM sur de grands ensembles de données.

Enfin, nous abordons les deux stratégies de combinaison pour la classification de données multi classes.

2.2. Principe de fonctionnement général :

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 » [42]. Dans le schéma qui suit, on détermine un hyperplan qui sépare deux ensembles de points.

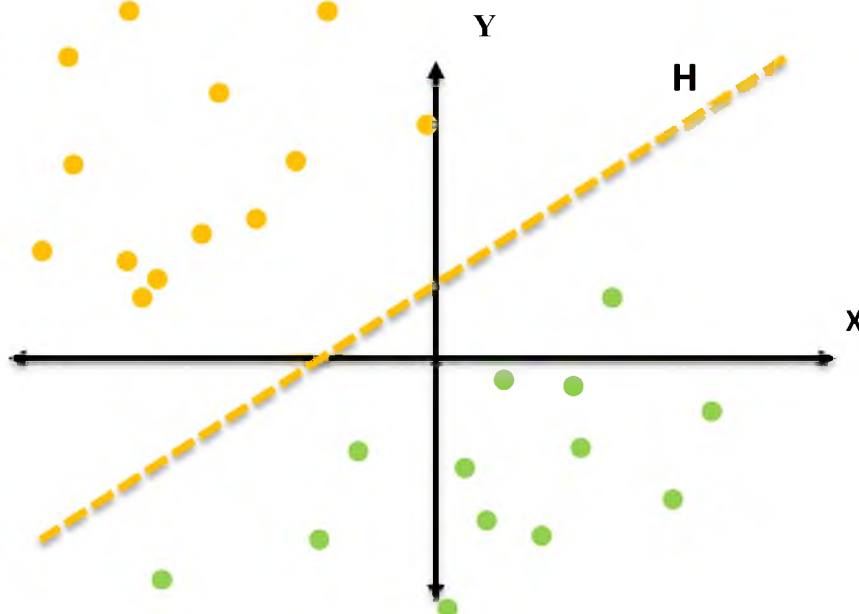


Figure 2. 1: Séparation de deux ensembles de points par un Hyperplan H

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 [43].

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

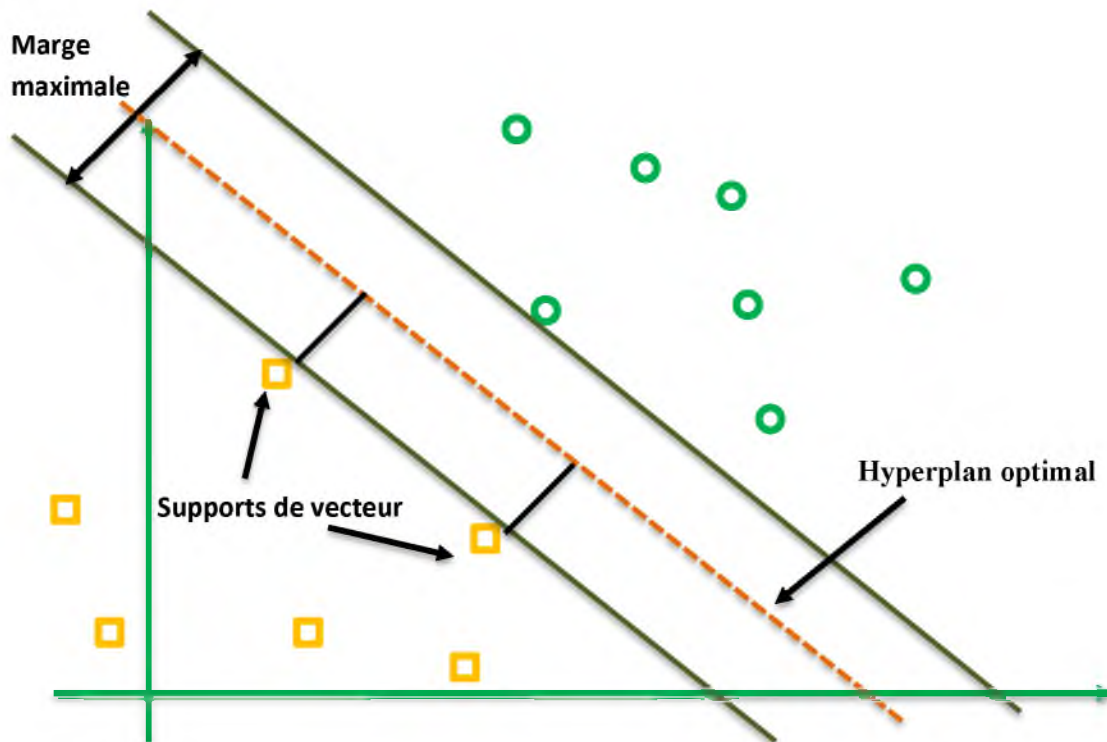


Figure 2. 2: Hyperplan optimal, marge et vecteurs de support

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.

2.2.1. Linéarité et non linéarité :

Dans les SVM, on constate deux cas de séparabilité :

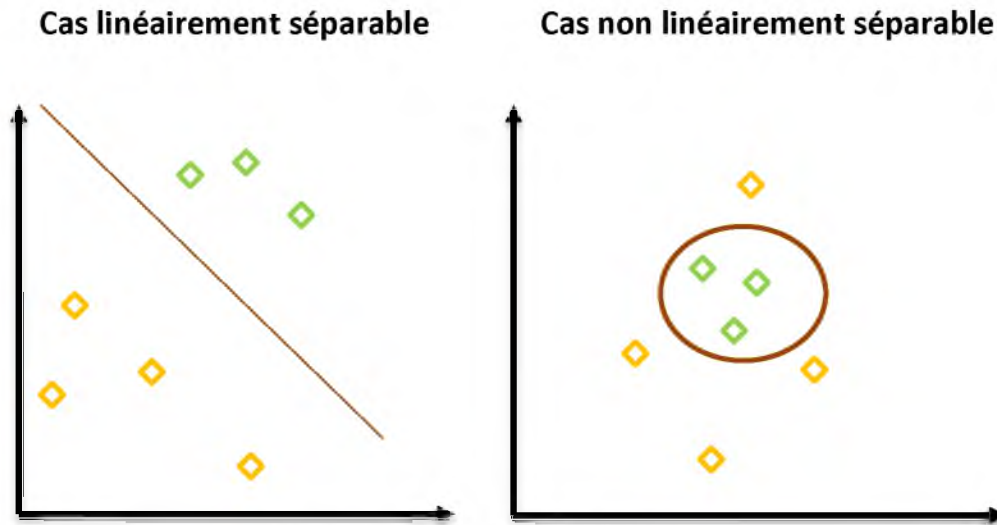


Figure 2. 3: linéarité et non linéarité

2.2.1.1. Cas linéairement séparable :

Un classifieur est dit linéaire lorsqu'il est possible d'exprimer sa fonction de décision par une fonction linéaire en. On peut exprimer une telle fonction par:

$$h(x) = \langle w|x \rangle + b = \sum_{i=1}^n w_i x_i + b \quad (Eq\ 2.1).$$

Où ($w \in \mathbb{R}^n$) est le vecteur de poids et le biais ($b \in \mathbb{R}^0$), alors que x est la variable du problème. X est l'espace d'entrée et qui correspond à \mathbb{R}^n , où n est le nombre de composantes des vecteurs contenant les données. Notons que l'opérateur $\langle \rangle$ désigne le produit scalaire usuel dans \mathbb{R}^n , w et b sont les paramètres à estimer de la fonction de décision $h(x)$.

Pour décider à quelle catégorie un exemple estimé x' appartient, il suffit de prendre le signe de la fonction de décision : $y = \text{sign}(h(x'))$. La fonction $\text{sign}()$ est appelée classifieur.

Géométriquement (voir figure 2.4), cela revient à considérer un hyperplan qui est le lieu des points x satisfaisant $\langle w|x \rangle + b = 0$. En orientant l'hyperplan, la règle de décision correspond à observer de quel côté de l'hyperplan se trouve l'exemple x' . On voit que le vecteur w définit la pente de l'hyperplan (w est perpendiculaire à l'hyperplan). Le terme b quant à lui permet de translater l'hyperplan parallèlement à lui-même [44].

L'objectif de la discrimination linéaire est de trouver la bonne fonction de décision $h(x)$. La classe de tous les hyperplans qui en découle sera notée H .

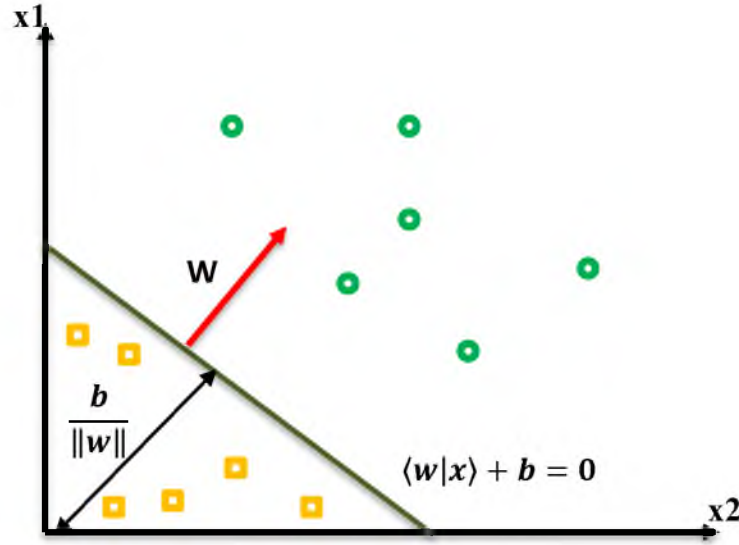


Figure 2. 4: linéarité et non linéarité [32]

2.2.1.2. Marge maximale de l'hyperplan :

La marge est la distance entre la frontière de séparation et les échantillons les plus proches. Dans les SVM, la frontière de séparation est choisie comme celle qui maximise la marge.

La marge géométrique représente la distance euclidienne prise perpendiculairement entre l'hyperplan et l'exemple x_i . En prenant un point quelconque x_p se trouvant sur l'hyperplan, la marge géométrique peut s'exprimer par :

$$\frac{b}{\|w\|} (x_i - x_p) \quad (Eq 2.2)$$

L'hyperplan à marge maximale est le modèle le plus utilisé dans les machines à vecteurs supports. L'estimation des paramètres (w^*, b^*) de l'hyperplan qui maximise la marge se fait en résolvant le problème d'optimisation suivante :

$$(w^*, b^*) = \underset{(w,b)}{\operatorname{argmax}} \{ \min(y_i(w x_i + b)), \|w\| = 1 \} \quad (Eq 2.3)$$

Dire que les deux classes de l'échantillon d'apprentissage S sont linéairement séparables est équivalent à dire qu'il existe des paramètres (w^*, b^*) tels que l'on a pour tous $(i=1,2,\dots, n)$:

$$w^* x_i + b^* > 0 \text{ si } y_i = 1 \quad (Eq 2.4)$$

$$w^* x_i + b^* < 0 \text{ si } y_i = -1 \quad (Eq 2.5)$$

Ce qui est équivalent à :

$$y_i(w^* x_i + b^*) > 0, \forall i = 1, 2, \dots, n \quad (Eq 2.6)$$

La définition consiste à dire qu'il doit exister un hyperplan laissant d'un côté toutes les données positives et de l'autre, toutes les données négatives. Dès lors, on peut définir deux plans se trouvant de part et d'autre de l'hyperplan et parallèles à celui-ci, sur lesquels reposent les exemples les plus proches.

Dans notre définition de l'hyperplan, il est possible que différentes équations correspondent au même plan géométrique :

$$\alpha(\langle \mathbf{w} | \mathbf{x} \rangle + \mathbf{b}^*) = 0 \quad (Eq\ 2.7)$$

α est une constante quelconque.

Il est donc possible de redimensionner de telle sorte que les deux plans parallèles aient respectivement pour équations :

$$(\mathbf{w}^* \mathbf{x}_i + \mathbf{b}^*) = 1 \quad (Eq\ 2.8)$$

$$(\mathbf{w}^* \mathbf{x}_i + \mathbf{b}^*) = -1 \quad (Eq\ 2.9)$$

Ces deux hyperplans sont appelés hyperplans canoniques. Ainsi la marge entre ces deux plans est égale à :

$$\gamma = \frac{b}{\|\mathbf{w}^*\|} \quad (Eq\ 2.10)$$

2.2.1.3. Cas non linéairement :

$$\xi_i = \max(0, \gamma - y_i)$$

$$\xi_i > \gamma \Leftrightarrow (\mathbf{x}_i, \mathbf{y}_i) \text{ est mal classé}$$

Soit \mathbf{x}_j^+ une observation de la classe positive

$$\begin{cases} \xi_i = 0 \Leftrightarrow Si\ \mathbf{x}_j^+ \in R_0 \\ 0 \leq \xi_i \leq \gamma \Leftrightarrow Si\ \mathbf{x}_j^+ \in R_1 \\ \xi_i \geq \gamma \Leftrightarrow Si\ \mathbf{x}_j^+ \in R_2 \end{cases} \quad (Eq\ 2.11)$$

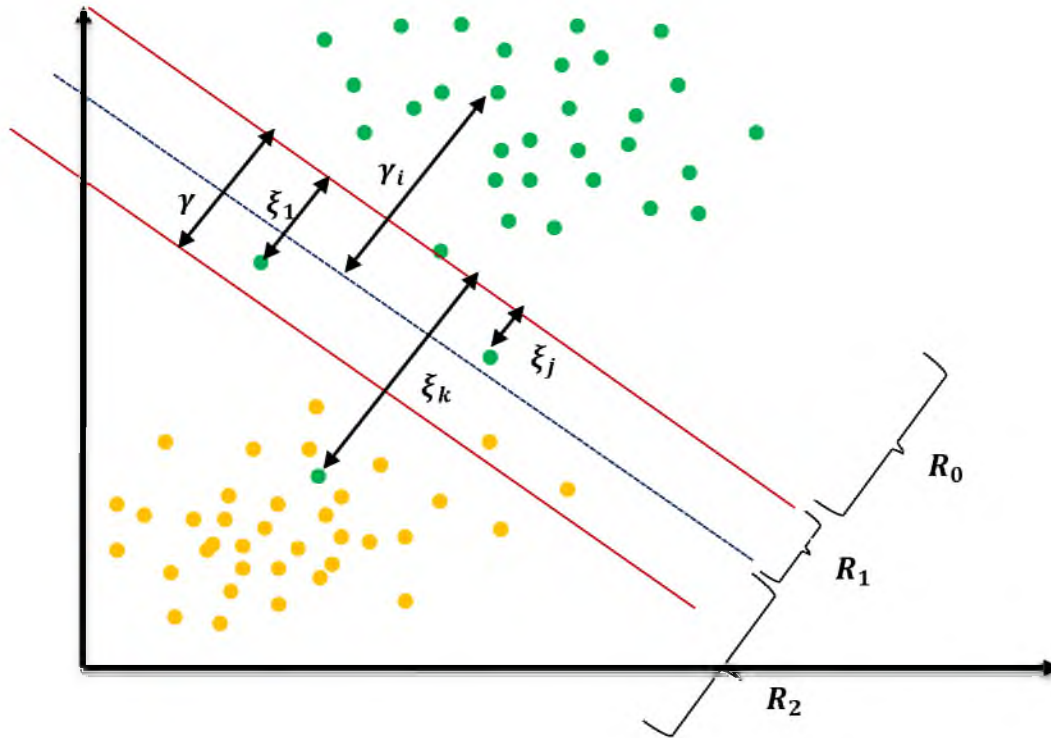


Figure 2. 5 : Données dans le cas non séparables.

ξ Variable d'écart à la marge objective γ qui est un réel strictement positif

En tenant compte de cette répartition, il est quasiment impossible de bien classer toutes les données. L'idée est de reformuler le problème d'apprentissage par l'introduction de variables auxiliaires dites variables ressorts pour assouplir les contraintes énoncés en (Eq 2.6), le problème s'énonce ainsi :

$$\begin{cases} \text{Min } w, b, \xi < \frac{1}{2} \|w\|^2 + C \sum \xi_i > \\ y_i(< w, x_i > + b) \geq 1 - \xi_i, \forall i = 1, \dots, n \\ \xi_i \geq 0, \forall i = 1 \dots n, y_i = \pm 1 \end{cases} \quad \text{Eq (2.12)}$$

C représente une constante de régularisation entre la marge et les erreurs de classement (Compromis entre la maximisation de la marge et la minimisation de l'erreur).

Sert à écarter le plus possible les deux droites (vecteurs support) et à pénaliser les points mal classés.

Cette fonction permet d'exprimer le problème d'optimisation primal par son dual :

$$\left\{ \begin{array}{l} \text{Max} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j x_i x_j \\ \sum_{i=1}^m \alpha_i y_i = 0 \\ 0 \leq \alpha_i \quad i = 1 \dots m \end{array} \right. \quad \text{Eq (2.13)}$$

Reste la même, la seule différence est que tout multiplicateur de Lagrange doit être borné supérieurement par **C**. Elle se présente comme suit :

$$\left\{ \begin{array}{l} \text{Max} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j x_i x_j \\ \sum_{i=1}^m \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \quad i = 1 \dots m \end{array} \right. \quad \text{Eq (2.14)}$$

Cette technique est appelée marge souple proposée par Cortes et Vapnik. Un autre exemple de non séparabilité linéaire où la présence de nuages de point nécessite un changement de la dimension du plan, ceci afin de se ramener à un cas connu :

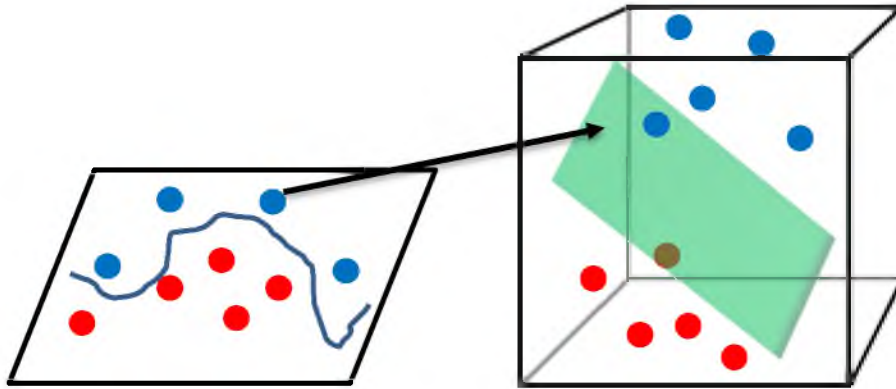


Figure 2. 6: Transformation de l'espace de représentation et l'hyperplan séparateur dans le cas non linéairement séparables

Il est généralement possible de trouver un séparateur linéaire en effectuant une projection d'un espace E à un espace F plus grand, voire infini en utilisant une fonction dite fonction de projection notée $\phi(x)$

$$\Phi: E \rightarrow F$$

$$x = (x_1, \dots, x_n) \rightarrow \phi(x) = (\phi_1(x), \dots, \phi_n(x), \dots)$$

$$\text{AVEC } \text{CARD}(E) < \text{CARD}(F)$$

Ainsi la fonction de décision devient $f(x) = \langle w, \phi(x) \rangle + b$ et la définition de la classe d'un nouveau point se calcul par : $y = \text{sgn}(\langle w, \phi(x) \rangle + b)$. Le problème d'optimisation exprimé en (Eq 2.13) est reformulé comme suit :

$$\begin{cases} \text{Max} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j \phi(x_i) \phi(x_j) \\ \sum_{i=1}^m \alpha_i y_i = 0 \\ 0 \leq \alpha_i \quad i = 1 \dots m \end{cases} \quad \text{Eq (2.15)}$$

Toute fois, l'expression $\langle \phi(x_j), \phi(x_i) \rangle$ demande des calculs complexes. Des fonctions permettent de calculer à partir des données initiales le produit scalaire, tel que :

$$\begin{cases} \forall x \in X, \forall z \in X: K(x, y) = \phi(x) \cdot \phi(y) \\ \text{Avec } K: X \times X \rightarrow \mathbb{R} \\ K: X \times X \rightarrow \mathbb{R} \end{cases} \quad \text{Eq (2.16)}$$

Est dite fonction noyau si et seulement si :

$$K(x, y)_{i,j = 1, \dots, l} \text{ est définie positive}$$

Ces fonctions possèdent les propriétés suivantes :

1. Positivité : $k(x, y) \geq 0$
2. symétrie : $k(x, y) = k(y, x)$
3. inégalité de Cauchy-Schwartz : $k(x, y) \leq x \cdot y$

Parmi les fonctions noyaux nous présentons :

1. **Noyau Linéaire:** $K(x_1, x_2) = x \cdot x'$
2. **Noyau polynômiale :**

Les noyaux polynômiaux sont de la forme:

$$K(x_1, x_2) = \delta + \langle x_1, x_2 \rangle^q$$

Dans ce cas, la règle de décision repose sur une statistique polynômiale de degré q et $\delta=1$ car ils sont en fonction du produit scalaire des observations, de tels noyaux sont dits projectifs.

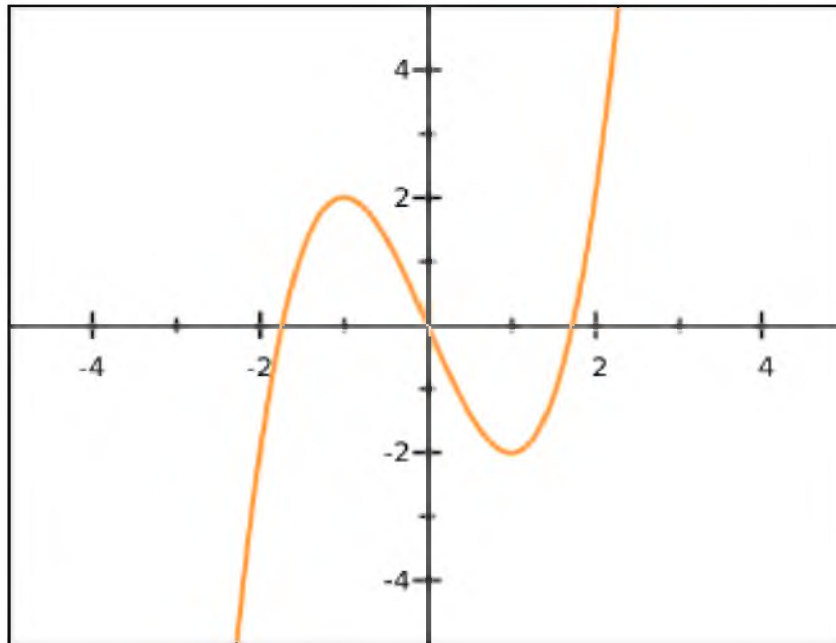


Figure 2. 7: Fonction polynômiale

3. Noyau RBF

Les noyaux « Radial Basis Function » sont de la forme:

$$K(x_1 x_2) = e^{-\|x_1 - x_2\|^2 / \sigma}$$

$\|x_1 - x_2\|$ présente la distance entre les observations et σ l'écart type. Bien qu'il n'existe pas de règle pour le choix du noyau, on retrouve fréquemment le noyau RBF en raison de son adaptabilité à un nombre de problèmes variés.

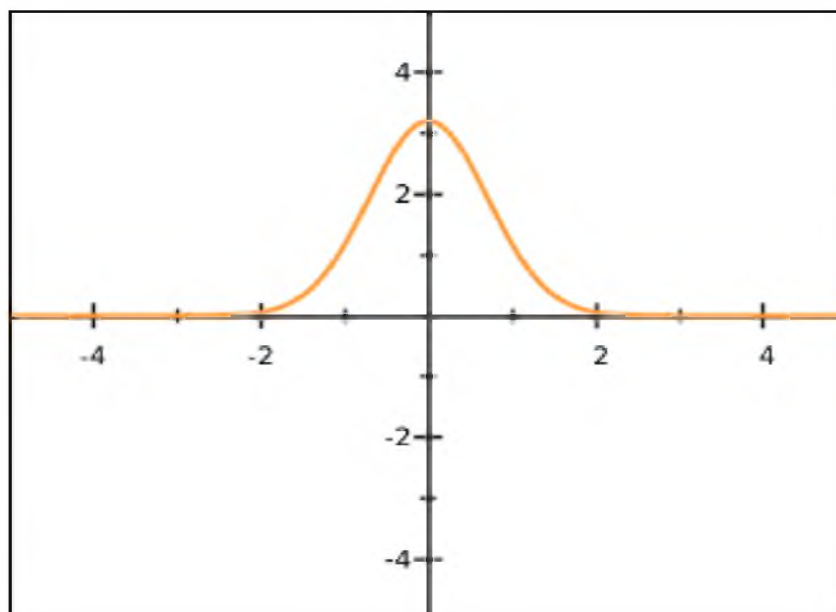


Figure 2. 8: Fonction RBF

Ainsi, le problème d'optimisation formulé par (Eq 2.12) peut être réécrit comme ci-dessous, cette technique se nomme Kernel Trick:

$$\left\{ \begin{array}{l} \text{Max} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \sum_{i=1}^m \alpha_i y_i = 0 \\ 0 \leq \alpha_i \quad i = 1 \dots m \end{array} \right. \quad \text{Eq (2.14)}$$

2.3. Implémentation multi classe :

Les machines à vecteur support sont dans leur origine binaire. Cependant, les problèmes du monde réel sont dans la plupart des cas multi classe, l'exemple le plus simple en est la reconnaissance des caractères optiques (OCR). Dans de tels cas, on ne cherche pas à affecter un nouvel exemple à l'une de deux classes mais à l'une parmi plusieurs, c-à-d que la décision n'est plus binaire et un seul hyperplan ne suffit plus.

Les méthodes des machines à vecteur support multi classe, réduisent le problème multi classe à une composition de plusieurs hyperplans biclasses permettant de tracer les frontières de décision entre les différentes classes [45, 46]. Ces méthodes décomposent l'ensemble d'exemples en plusieurs sous-ensembles représentant chacun un problème de classification binaire. Pour chaque problème un hyperplan de séparation est déterminé par la méthode SVM binaire. On construit lors de la classification une hiérarchie des hyperplans binaires qui est parcourue de la racine jusqu'à une feuille pour décider de la classe d'un nouvel exemple. On trouve dans la littérature plusieurs méthodes de décomposition :

2.3.1. Un contre tous (OAA : One Against All) :

C'est la méthode la plus simple et la plus ancienne. Selon la formulation de Vapnik [46], elle consiste à déterminer pour chaque classe k un hyperplan $H_k(w_k, b_k)$ la séparant de toutes les autres classes. Cette classe k est considéré comme étant la classe positive (+1) et les autres classes comme étant la classe négative (-1), ce qui résulte, pour un problème à K classes, en K SVM binaires. Un hyperplan H_k est défini pour chaque classe k par la fonction de décision suivante :

$$\begin{aligned} H_k(x) &= \text{signe}(\langle w_k | x \rangle + b_k) \\ &= \begin{cases} +1 & \text{si } f_k(x) > 0; \\ 0 & \text{sinon} \end{cases} \end{aligned} \quad \text{Eq(2.15)}$$

La valeur retournée de l'hyperplan permet de savoir si x appartient à la classe k ou non. Dans le cas où il n'appartient pas à k ($H_k(x) = 0$), nous n'avons aucune information sur l'appartenance de x aux autres classes. Pour le savoir, on présente x à tous les hyperplans, ce qui donne la fonction de décision de l'équation suivante :

$$k^* = \underset{1 \leq k \leq K}{\text{Arg}} (\text{Max} H_k(x)) \text{ Eq 2.16}$$

Si une seule valeur $H_k(x)$ est égale à 1 et toutes les autres sont égales à 0, on conclut que x appartient à la classe k . Le problème est que l'équation Eq (2.16) peut être vérifiée pour plus d'une classe, ce qui produit des régions d'ambiguïté, et l'exemple x est dit non classifiable. La figure 2.9 représente un cas de séparation de 3 classes.

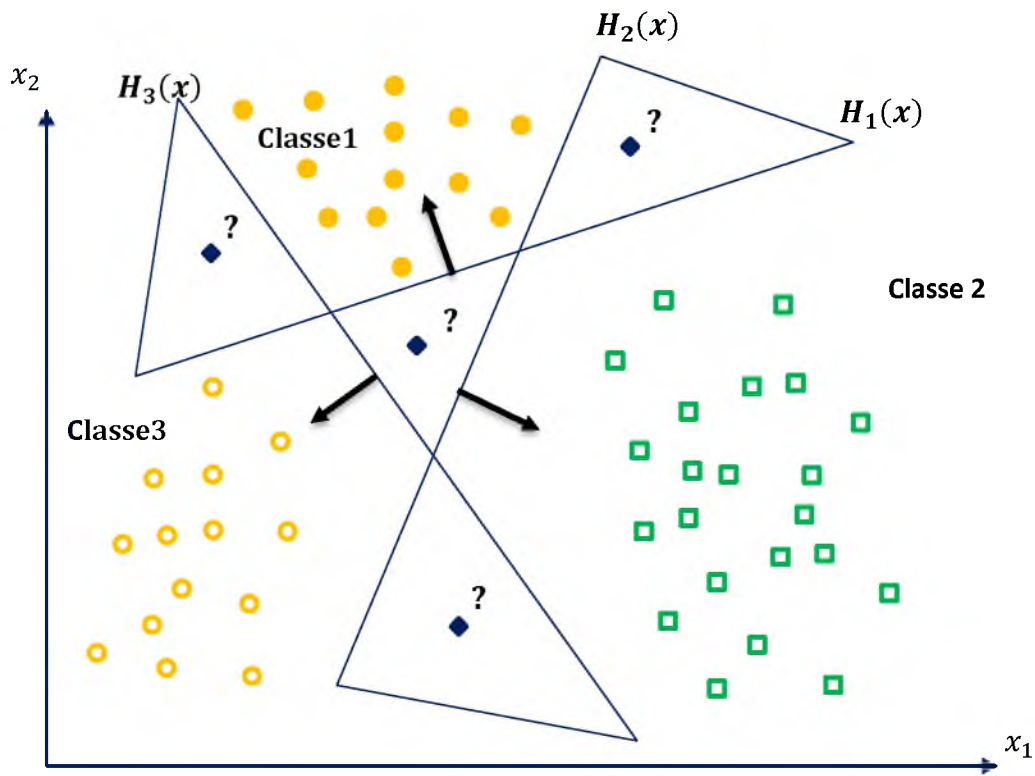


Figure 2.9 : Approche une-contre-reste avec des zones d'indécision

Pour surmonter cette situation, la méthode 1vsR utilise le principe de "le gagnant prend tout" ("winner-takes-all") : la classe k retenue est celle qui maximise $f_k(x) = (w_k, x) + b_k$ de l'équation suivant.

$$k^* = \underset{1 \leq k \leq K}{\text{Arg}} \text{Max} (\langle w_k | x \rangle + b_k) \text{ Eq 2.17}$$

Géométriquement interprétée, tout nouvel exemple x est affecté à la classe dont l'hyperplan est le plus loin de x , parmi les classes ayant $H(x) = 1$.

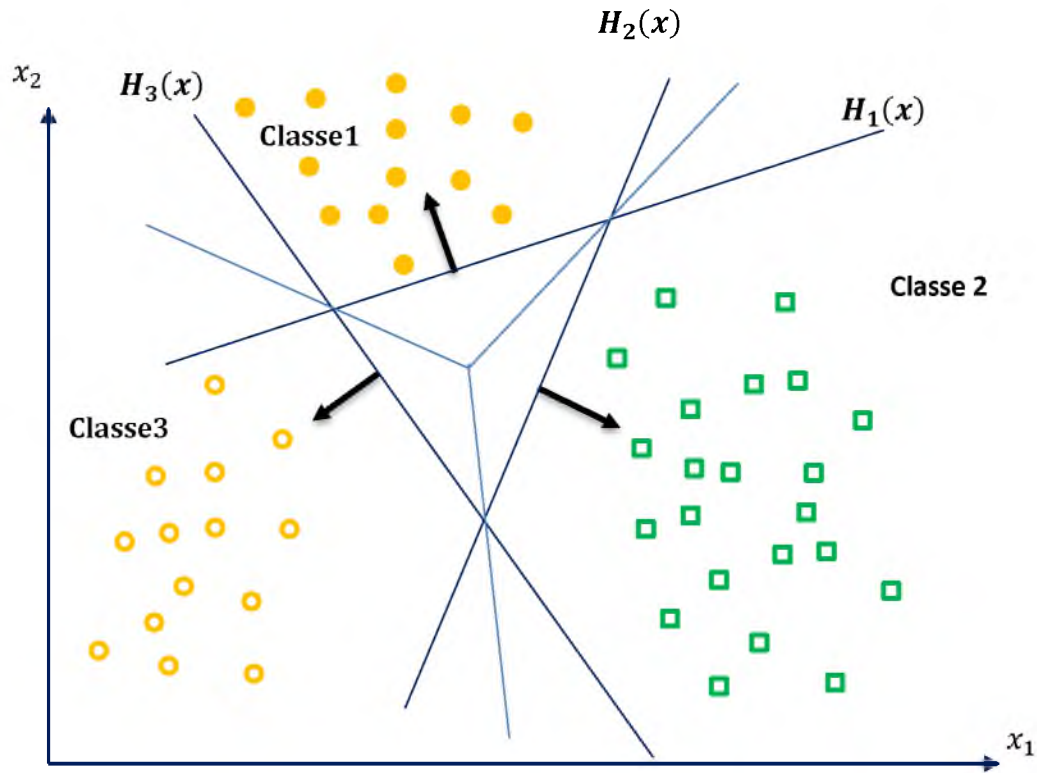


Figure 2. 10: Résolution des cas d'indécision dans la méthode 1vsR

La méthode 1vsR peut être utilisée pour découvrir même les cas de rejet où un exemple n'appartient à aucune des K classes. Pour cela, on prend les deux fonctions de décision les plus élevées, puis on calcule leur différence, si elle est au-dessous d'un certain seuil, l'exemple est rejeté.

Souvent, la méthode 1vsR est critiquée à cause de son asymétrie [49], puisque chaque hyperplan est entraîné sur un nombre d'exemples négatifs beaucoup plus important que le nombre d'exemples positifs. Par exemple dans le cas de l'OCR, le classifi du caractère 'A' est entraîné sur des exemples positifs représentant 'A' et des exemples négatifs représentant tous les autres caractères. La méthode une contre une suivante est une méthode symétrique qui corrige ce problème.

2.3.2. Un contre un (OAO : One Against One) :

Cette méthode, appelée aussi "pairwise", revient à Kner et ses co-auteurs [47] qui l'ont proposée pour les réseaux de neurones. Elle consiste à utiliser un classifi pour chaque paire de

classes. Au lieu d'apprendre K fonctions de décisions, la méthode 1vs1 discrimine chaque classe de chaque autre classe, ainsi $K(K-1)/2$ fonctions de décisions sont apprises.

Pour chaque paire de classes (k, s) , la méthode 1vs1 définit une fonction de décision binaire $h_{ks}: \mathbf{R} \rightarrow \{-1, +1\}$. L'affectation d'un nouvel exemple se fait par liste de vote. On teste un exemple par le calcul de sa fonction de décision pour chaque hyperplan.

Pour chaque test, on vote pour la classe à laquelle appartient l'exemple (classe gagnante). On définit pour le faire la fonction de décision binaire $H_{ks}(x)$ de l'équation.

$$\begin{aligned} H_{ks}(x) &= \text{signe}(f_{ks}(x)) \\ &= \begin{cases} +1 & \text{si } f_{ks}(x) > 0; \\ 0 & \text{sinon} \end{cases} \end{aligned} \quad \text{Eq(2.18)}$$

Sur la base des $K(K-1)/2$ fonctions de décision binaires, on définit K autres fonctions de décision

$$H_k(x) = \sum_{s=1}^m H_{ks}(x) \quad \text{Eq 2.19}$$

Un nouvel exemple est affecté à la classe la plus votée. La règle de classification d'un nouvel exemple x est donnée par l'équation:

$$k^* = \underset{1 \leq k \leq K}{\text{Arg}} (\text{Max}(H_k(x))) \quad \text{Eq 2.20}$$

Malheureusement, la fonction Eq (2.20) peut être vérifiée pour plusieurs classes, ce qui produit des zones d'indécisions. La méthode de vote affecte dans ce cas, un exemple aléatoirement à l'une des classes les plus votées.

La Figure suivant représente un exemple de classification de trois classes avec la zone d'indécision.

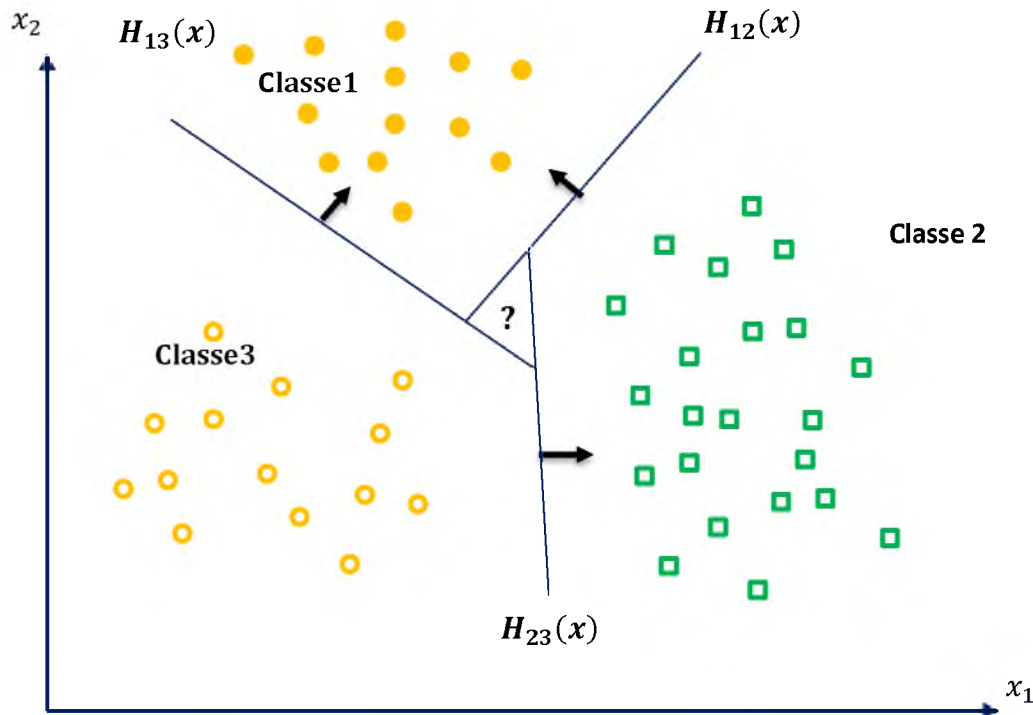


Figure 2. 11: Approche une-contre-une

Bien que La méthode 1vs1 utilise, pour l'entraînement, un nombre plus important d'hyper plans que la méthode 1vsR, elle est souvent plus rapide. Cela est dû, d'une part, au nombre limité d'exemples utilisés pour entrainer chaque hyperplan, et d'autre part, à la simplicité des problèmes à résoudre. En effet chaque deux classes prises à part sont moins chevauchées que toutes les classes.

2.4. Les domaines d'applications des SVM :

SVM est une méthode de classification qui montre de bonnes performances dans la résolution des problèmes variés. Cette méthode a montré son efficacité dans de Nombreux domaines d'applications tels que :

1. Classification de données biologiques/physiques.
2. Classification de documents numériques.
3. Classification d'expressions faciales.
4. Classification de textures.
5. E-learning.
6. Détection d'intrusion.
7. Reconnaissance de la parole.

8. CBIR : Content Based Image Retrieval.

2.5. Les avantages et les inconvénients des SVM :

2.5.1. Avantages :

SVM est une méthode de classification intéressante car le champ de ses applications est large, parmi ses avantages nous avons :

- Un grand taux de classification et de généralisation par rapport aux méthodes classiques.
- Elle nécessite moins d'effort pour designer l'architecture adéquate (petit nombre de paramètre à régler ou à estimer).
- La résolution du problème est convertie en résolution d'un problème quadratique convexe dont la solution est unique et donnée par des méthodes mathématiques classiques de programmation quadratique.

2.5.2. Inconvénients :

- L'inconvénient majeur du classificateur SVM est qu'il est désigné ou conçu pour la classification binaire (la séparation entre deux classes une +1 et l'autre -1) [48].

2.6. Conclusion :

Nous avons présenté dans ce chapitre le système d'apprentissage introduit par Vladimir Vapnik les « Support Vector Machine ». Cette méthode de classification est basée sur la recherche d'un hyperplan qui permet de séparer au mieux des ensembles de données. Nous avons expliqués les différents aspects de cette méthode de classification, le cas de deux classes et celui de multi-classes en présentant les théories qui sont à la base des SVM à savoir les fonctions noyaux et la notion de marge maximale. Les avantages que présentent les SVM sont l'utilisation de l'approche de Kernel pour traiter les cas non-linéaires, le nombre raisonnable de solutions et le contrôle de capacité obtenu par optimisation de la marge.

Chapitre III

Expérimentations et Résultats

3.1. Introduction :

Nous présentons dans ce chapitre une description détaillée de notre système de reconnaissance des montants littéraux 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 par l'utilisation des différents noyaux du SVM pour trouver le meilleur taux de reconnaissance possible.

3.2. Ressources matérielles et logicielles :

3.2.1. Ressources matérielles

Nous avons utilisé dans ce travail un PC portable (CORE 2 Duo CPU 2.30 Ghz, avec 4 Go de RAM), utilisé pour réaliser notre système de reconnaissance et aussi la création de notre base de données (les corpus).

3.2.2. Ressources logicielles

3.2.2.1. Bref présentation de C++ Builder Embarcadero :

C++Builder est un environnement de programmation visuel orienté objet pour le développement rapide d'applications (RAD). En utilisant C++Builder, vous pouvez, avec un minimum de codage manuel, créer de performantes applications pour Microsoft Windows XP, Microsoft Windows 2000 et Microsoft Windows7.

C++Builder fournit tous les outils nécessaires pour développer, tester et déployer vos applications, notamment une importante bibliothèque de composants réutilisables, une suite d'outils de conception, des modèles d'applications et de fiches, ainsi que des experts de programmation.

La figure 3.1 représente un exemple typique de l'interface de C++ Builder au cours d'une session de travail.

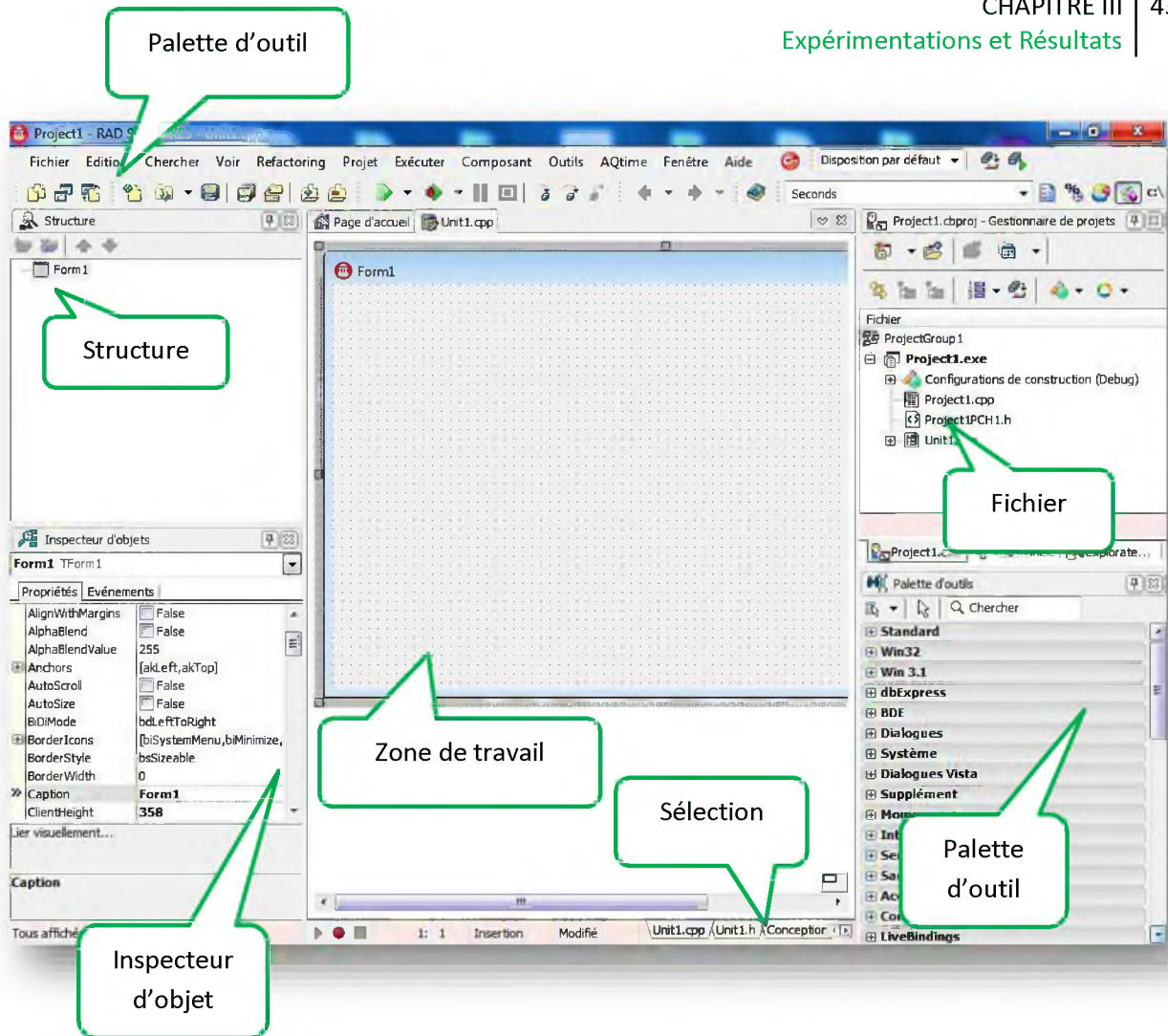


Figure 3. 1 : L'interface de C++ builder Embarcadero

3.2.2.2. Environnement de développement

Nous avons développé notre système à l'aide du langage C++ Builder. Qui est un environnement de programmation visuel orienté objet pour le développement rapide d'applications (RAD). En utilisant C++Builder, vous pouvez, avec un minimum de codage manuel, créer de performantes applications.

C++Builder fournit tous les outils nécessaires pour développer, tester et déployer des applications, notamment une importante bibliothèque de composants réutilisables, une suite d'outils de conception, des modèles d'applications et de fiches, ainsi que des experts de programmation.

SVMmulticlass

Nous avons utilisé l'implémentation *SVMmulticlass* qui est un moteur SVM développé par Thorsten Joachims en 2008.

SVMmulticlass est une implémentation open-source de référence pour les machines à support vectoriel.

Elle est disponible sur http://www.cs.cornell.edu/People/tj/svm%5Flight/svm_multiclass.html avec documentation, exemples, et références bibliographiques.

Le code est implémenté en C et devenu un des moteurs les plus utilisés actuellement. Il possède des nombreuses fonctionnalités et des caractéristiques très attractives pour l'utilisateur :

- ❖ Il permet d'utiliser des nombreux kernels prédéfinis, et même d'utiliser des kernels définis par l'utilisateur.
- ❖ Il implémente des algorithmes rapides pour l'optimisation en termes de temps de calcul et gestion de mémoire [37].
- ❖ Il fournit tout un ensemble de paramètres qui permettent d'évaluer les performances des SVM après chaque processus d'apprentissage, et plus concrètement :
 - Il montre une estimation de la dimension de Vapnik-Chervonenkis.
 - Il calcule la précision sur la base d'apprentissage selon une procédure «leaveone out » [38].
- ❖ Il permet d'aller au-delà de la classification pour aborder des problèmes de régression, et il admet une variante appelée SVM-Struct pour les problèmes multi variés et structurés.

3.3. Le but de notre application :

On va établir un système permettre de reconnaître automatique de montants littéraux des chèques en utilisant les machines à vecteur support.

3.4. Les phases de notre system :

3.4.1. L'acquisition des images :

Les images sont acquises par, Y. Al-Ohali et al. [39]. Ont achevé la réalisation d'une base de données pour la reconnaissance de chèques arabes manuscrits sur le nom CNEMPARMI.

Cette base se compose de 7000 mots de montant des chèques saoudiens issus de la pratique bancaire, scannés en niveaux de gris avec une résolution de 300 dpi.

واحد	اثنان	اثنين	ثلاث
ثلاثة	أربع	أربعة	خمس
خمسة	ست	سبعة	سبع
سبعة	ثمان	ثمانية	تسع
تسعة	عشر	عشرة	أحد
أحدى	اثنى	عشرون	عشرين
ثلاثون	ثلاثين	أربعون	أربعين
خمسون	خمسين	ستون	ستين

(a)

ستة آلاف ريالاً

عشرون الف ريال

(b)

Figure 3. 2: Des échantillons de base de données standard (a) et montants des chèques (b).

3.4.2. Prétraitement

Dans cette étape on procède à une série d'opérations :

- Binarisation (pour le cas général), et des techniques pour éliminer le bruit par seuillage.
- Dilatation, cette opération est effectuée parfois pour agrandir l'épaisseur du tracé.
- Erosion, utilisé pour éliminer le bruit (les pixels éloignés) pour pouvoir extraire le cadre qui inclut le caractère.
- Normalisation, La dernière étape du prétraitement est la normalisation de l'image. L'ensemble de données à différentes tailles d'image. Il est important de faire de l'image dans la base de données de la même taille et de rendre le processus de reconnaissance rapide. Après avoir testé plusieurs tailles, la taille 130 * 130 donnait le meilleur taux de reconnaissance. Par conséquent, toutes les images sont normalisées en taille 130 * 130.

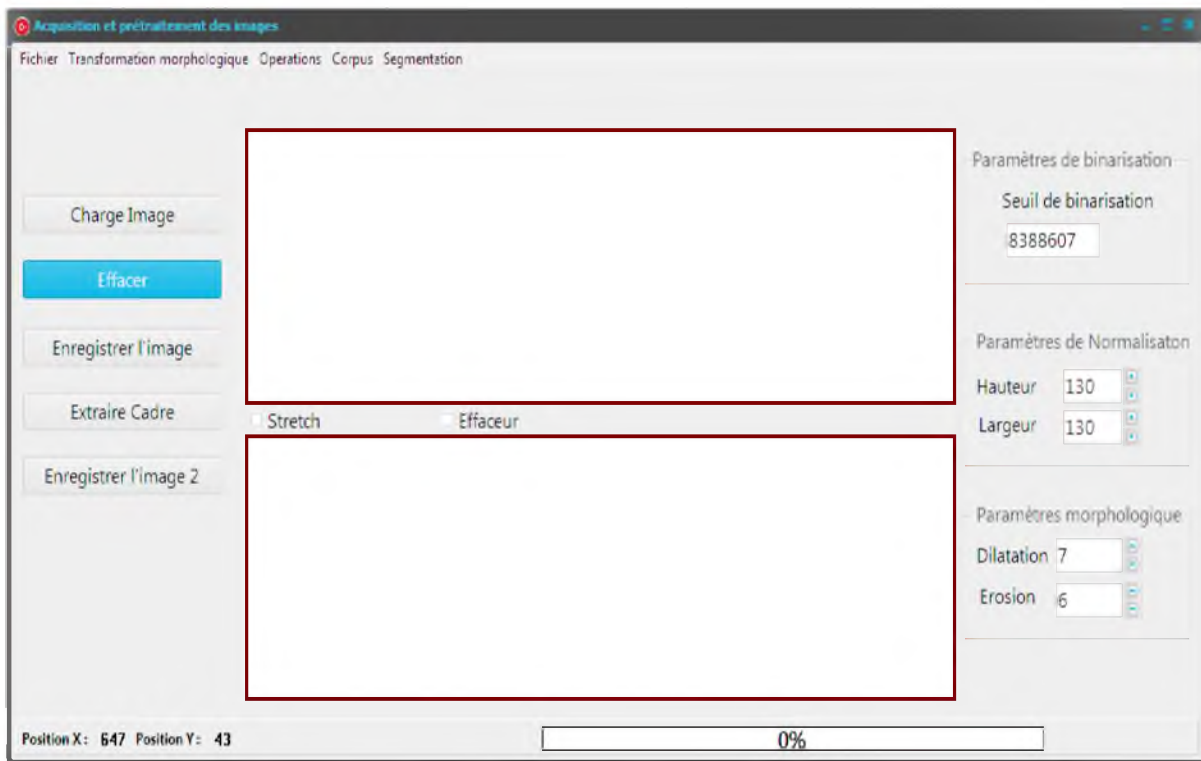


Figure 3. 3: L'interface d'Acquisition et prétraitement des images

3.4.3. La Segmentation

L'écriture arabe est cursive ; Par conséquent, les mots et les sous-mots sont séparés par des espaces, de sorte que les limites de mots sont toujours représentées par un espace. Selon cela, les distances entre Chaque paire de sous-mots consécutifs sont obtenues.

Les distances entre les mots sont plus grandes que les distances entre les sous-mots, de sorte que les mots peuvent être segmentés en comparant cette distance par rapport à un seuil approprié.

Pour déterminer un tel seuil, les critères bayésiens d'erreur de classification minimale sont utilisés comme suit.

Donnée d'une distance de d ; La probabilité qui représente une séparation du mot ou du sous mot est désignée par $P_m(d)$ et $P_{s_m}(d)$, respectivement. Ces deux probabilités conditionnelles sont obtenues en analysant manuellement plus de 135 images contenant plus de 880 mots. Prenez $P_m(d)$ par exemple, nous trouvons toutes les distances possibles pour séparer un mot, puis calculons leur histogramme, et $p_m(d)$ est estimé par cet histogramme.

Certes, ces distances se trouvent sur la base de notre projection verticale obtenue de l'image.

Ensuite, une distance optimale d_0 est obtenue selon les critères d'erreur de classification minimum Bayésien :

$$d_0 = \operatorname{argmin}(\operatorname{err}(d))$$

$$err(d) = \int_d^{\infty} Ps_m(x)dx + \int_0^d Pm(x)dx$$

En comparant la distance d avec cette distance ou seuil optimal d_0 . Si nous avons $d_0 > d$, il s'agit de deux mots.

Sinon, il s'agit de deux sous-mots [40]. Selon l'image originale de la Figure 3.4, les derniers mots segmentés sont représentés sur la Figure 3.5.

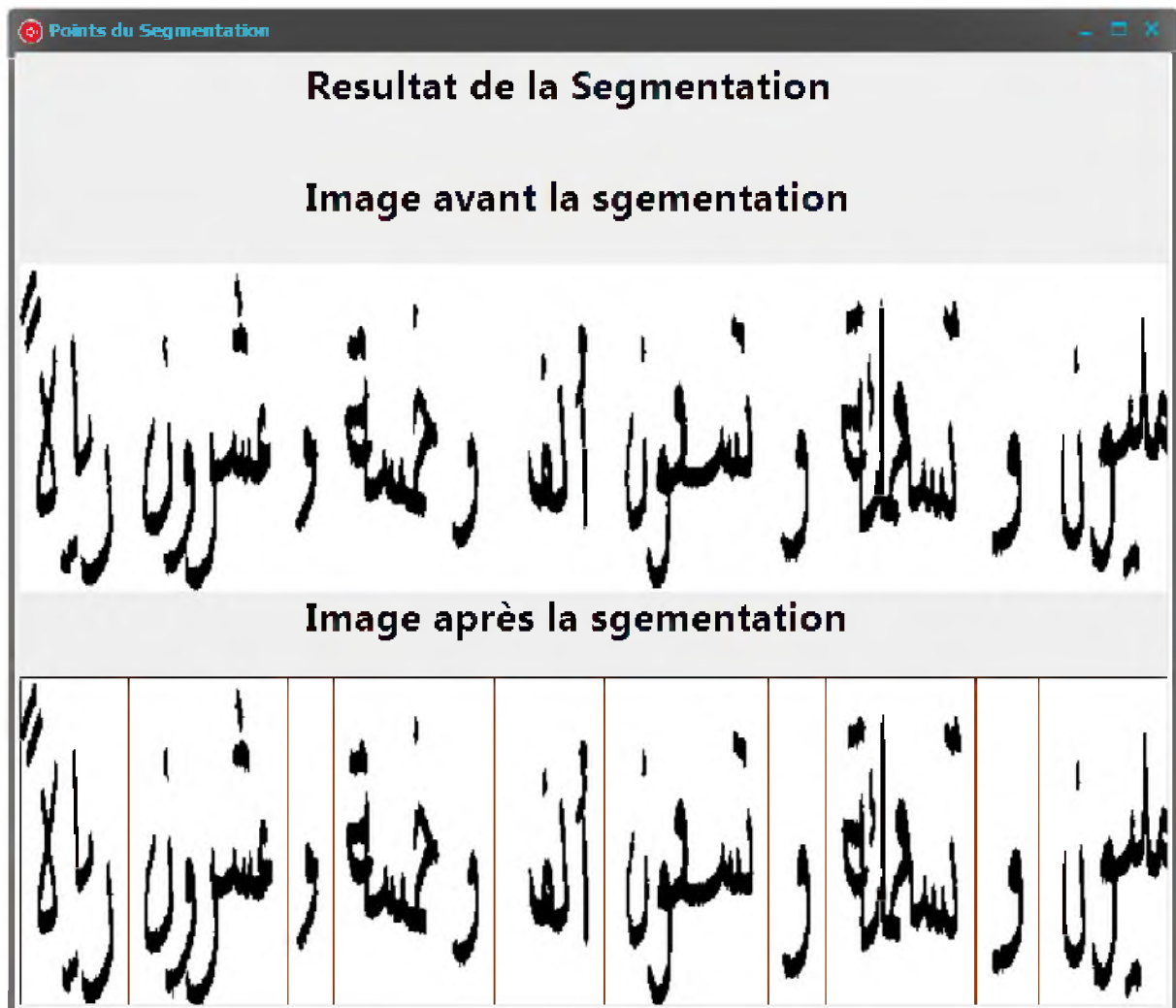


Figure 3. 4: Interface de la segmentation

L'algorithme de extraction de seuil d_0 est ont détaillé par organigramme suivant :

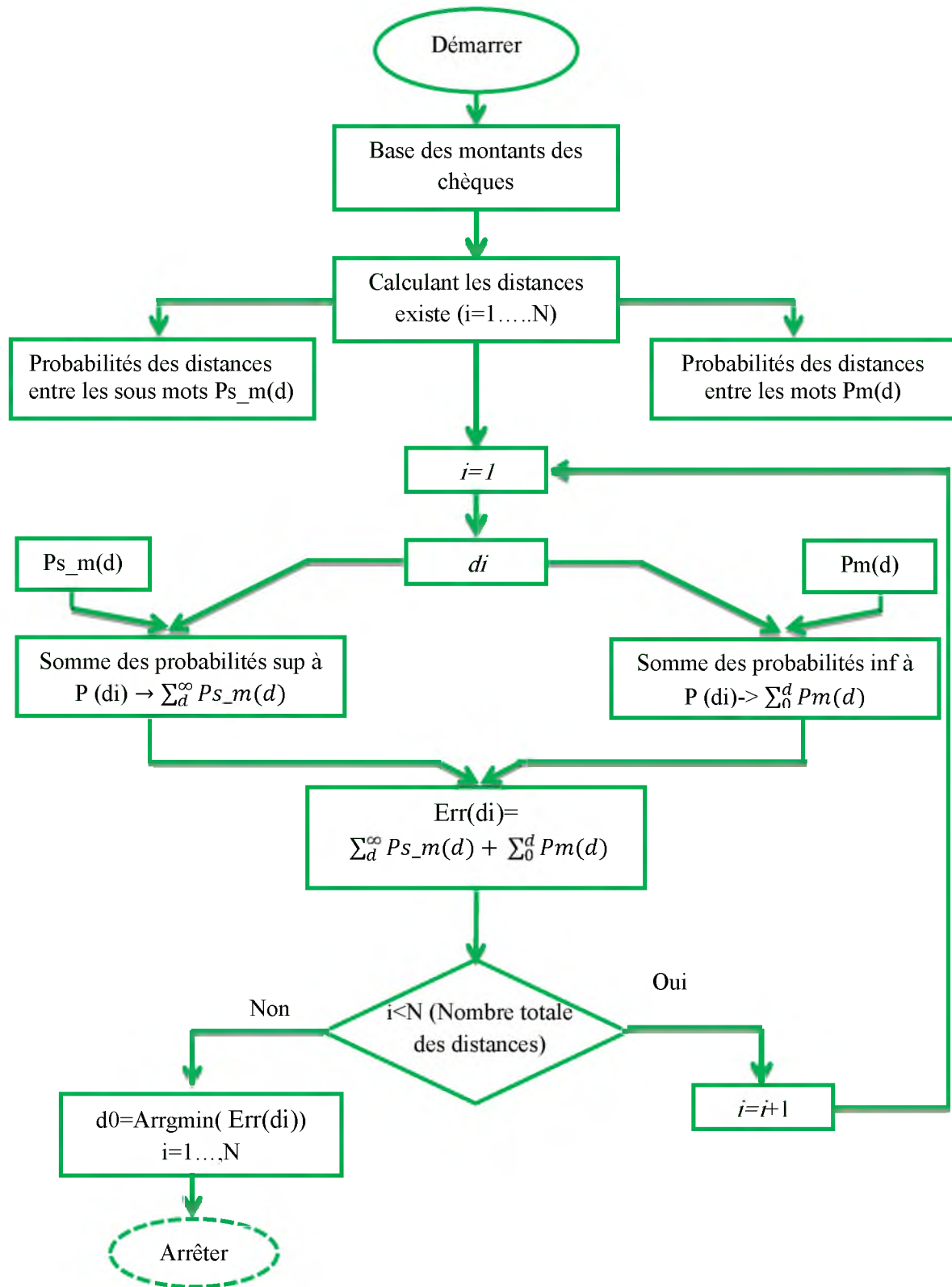


Figure 3. 5 : L'organigramme de l'extraction de seuil d0

La figure 3.6 représente L'organigramme de la segmentation

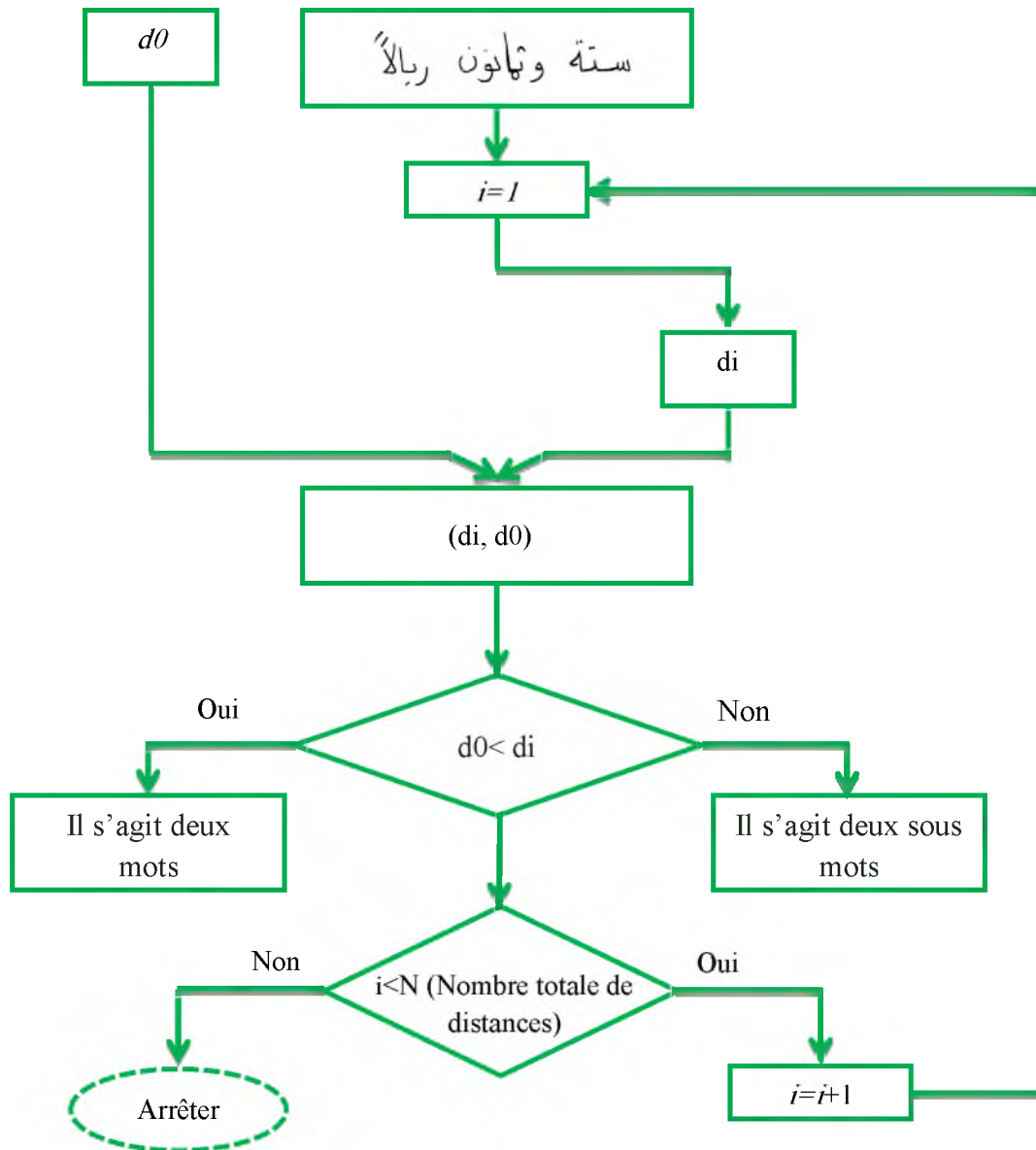


Figure 3. 6: L'organigramme de la segmentation

3.4.4. L'extraction des primitives :

Donc, dans notre système, l'extraction des primitives basé sur la Construction de la matrice de distribution

3.4.4.1. Construction de la matrice de distribution :

La construction de la matrice de distribution est l'une des phases importantes dans notre système.

Pour une matrice de distribution de taille $N \times N$, le principe consiste à superposer une grille ($N \times N$) sur l'image du caractère et calculer pour chacune des cellules $[i,j]$ résultantes le nombre de pixels noir, puis attribuer ce nombre à la case $[i,j]$ de la matrice de distribution.

Algorithme : de Construction de la matrice de distribution

Entrée :

IM : Image de montant littéraux 130*130 ;

Nb : Compteur ;

D : Matrice de distribution 26*26 ;

I,J,L,H : Entier;

L=0 ; H=0 ;

Tanque L < 130 faire

Tanque H < 130 faire

Nb =0 ;

Pour (I=L Jusqu'à faire L+5)

Pour (J=H Jusqu'à faire H+5)

Si pixel IM (I, J) est noire alors Nb=Nb+1 ;

D[L DIV 5, H DIV 5]=Nb ;

H=H+5;

L=L+5;

Figure 3. 7: Algorithme de Construction de la matrice de distribution

3.4.4.2. Corpus :

Le corpus est un fichier texte possède une structure particulière pour pouvoir être utilisée directement par l'application *SVMmulticlass*. Chaque ligne du corpus représente le vecteur de caractéristique d'une image, la première valeur de la ligne est le numéro de la classe, ensuite chaque valeur de vecteur de caractéristique est précédé par un index (Index:Valeur).

Dans notre système de reconnaissance le vecteur de caractéristique est de taille 676, les 676 premier représentent les valeurs de matrice de distribution (26x26).

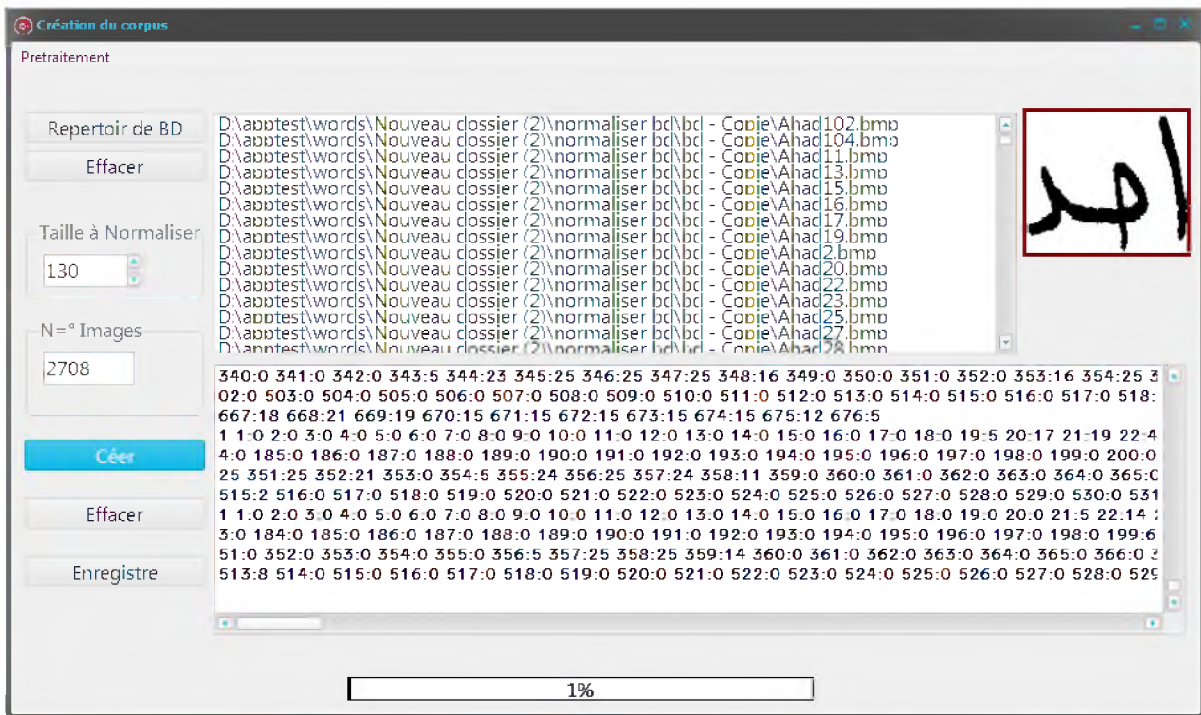
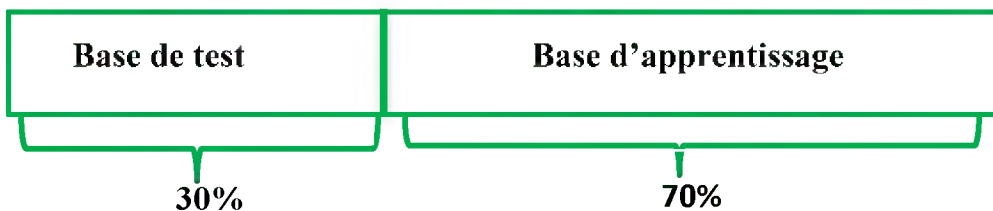


Figure 3. 8: Interface de création des corpus

3.5. Expérimentations et Résultats

Nous avons utilisé 4062 images de la base CENMPARMI avec 39 classes. Chaque classe a 105 images écrites dans un style différent. Dans le système de reconnaissance de mots, 70% de base de données utilisé à pour l'apprentissage (2708) et 30% pour les tests (1354).

La figure montre la décomposition de la base construite.



3.5.1. Résultats de la segmentation

Nous avons testé cette techniques sur 135 images dans l'ensemble des chèques et les résultats sont résumés dans le Tableau 3.1 Ci-dessous.

Segmentation des mots	Le nombre de mots existants	segmentation correcte	Précision	Erreur
	562	508	90.39%	9.61%

Tableau 3. 1: Résultats général de la segmentation

À partir du Tableau 3.1, on peut voir que le taux de segmentation correct pour les images est de 90.39%, mais l'erreur de segmentation est de 9.61%. En raison de la variation de l'écriture manuscrite, en particulier des espaces irréguliers entre les sous-mots et les mots, tels que les espaces trop grand entre les sous-mots et Les mots, (qui peuvent être pris comme deux mots et conduire à une segmentation excessive), ou des espaces trop petits entre les mots, (qui conduiront sous segmentation en fusionnant incorrectement deux mots ensemble).

La figure 3.9 illustre les erreurs.

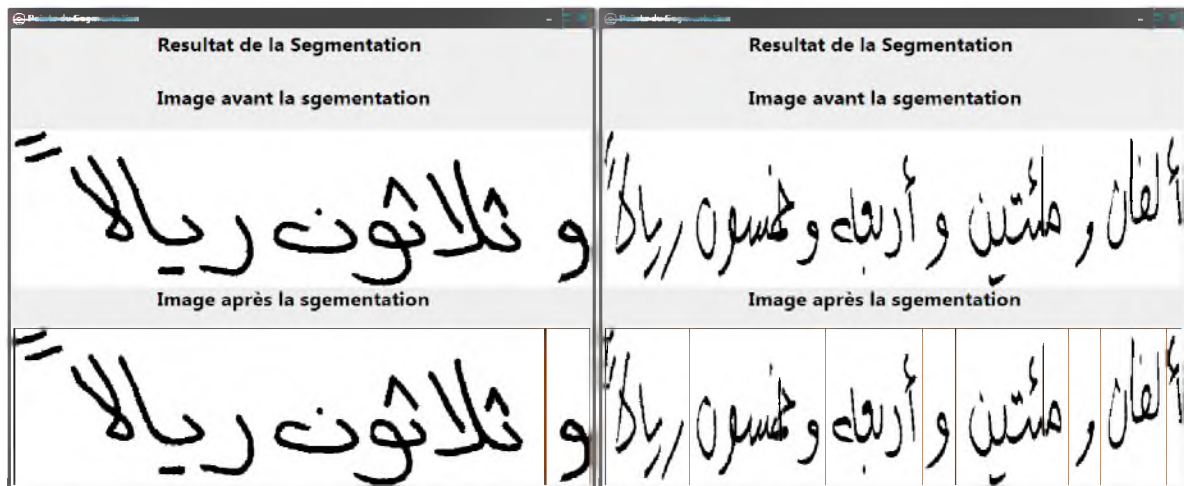


Figure 3. 9: Exemples des erreurs sur la segmentation

3.5.2. Résultats de reconnaissance des montants littéraux :

Les résultats obtenus sont rapportés dans le tableau suivant :

Base de Données	Noyau	C	σ	D	coefficient	Durée d'apprentissage	Durée de prédiction	Taux de prédiction
Base Standard	polynomial	100	/	2	10	13.14s	4.15s	81.46
CNEMPA RMI	RBF	1000	2.1e-05	/	/	15.53s	5.01s	83.53

	Linéaire	77	/	/	/	11.62s	3.76s	80.35
--	----------	----	---	---	---	--------	-------	-------

Figure 3. 10: Le taux erreur et prédiction de la classification à partir des différents noyaux

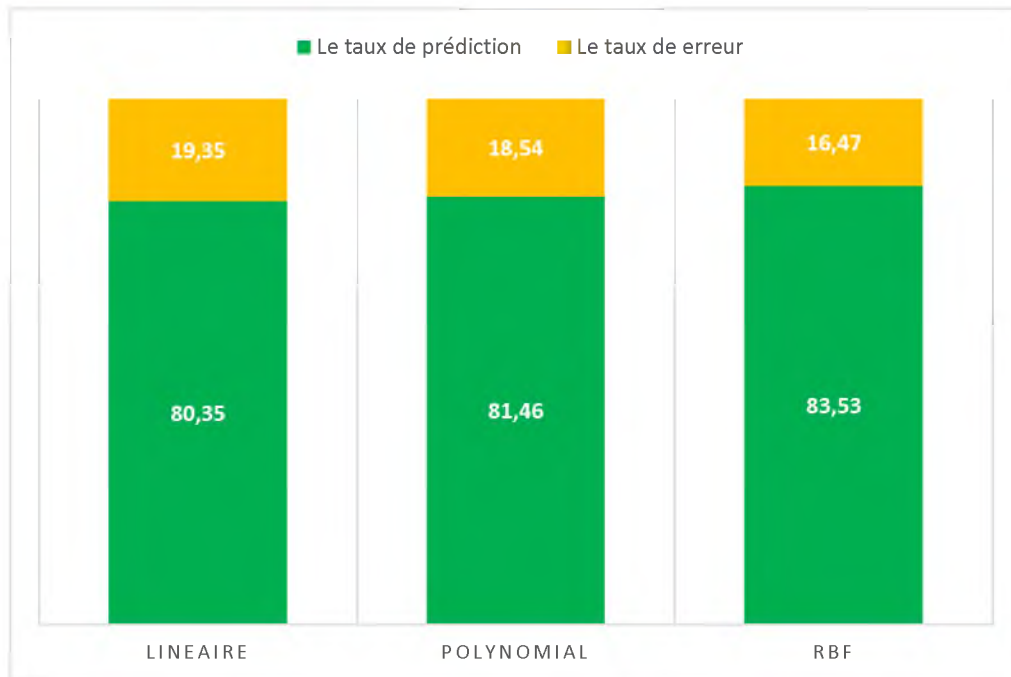


Tableau 3. 2 : Le taux de reconnaissance obtenu avec le différent noyau

Notre objectif est d'effectuer une série d'expérimentations et une recherche exhaustive afin de sélectionner le type de la classification une-contre-une, le noyau et les paramètres du noyau qui donnent les meilleurs résultats en termes de taux de prédiction et le temps CPU.

D'après les résultats, on a le taux d'erreur est très élevé en utilisant un noyau linéaire, par contre, remarqué le taux de reconnaissance est significativement augmenté avec un noyau polynomial de degré 2 et coefficient =10. Le noyau RBF a donné des résultats meilleurs que les autres noyaux, particulièrement avec $\sigma = 2,1e-05$ (83,53%). un noyau linéaire (80.35%), et un noyau polynomial de degré d=2 et coefficient =10 (81.46%). Le temps d'apprentissage et en fonction du type du noyau utilisé, les valeurs accordées à ces paramètres, et la taille de la base d'apprentissage. Le noyau RBF présente un temps d'apprentissage beaucoup plus long que le noyau linéaire et polynomial.

Pour le cas de la base de montant littéraux manuscrits, une analyse de la matrice de confusion du système obtenue, présentée sur le Figure 3.16 et Figure 3.17 ci-dessous.

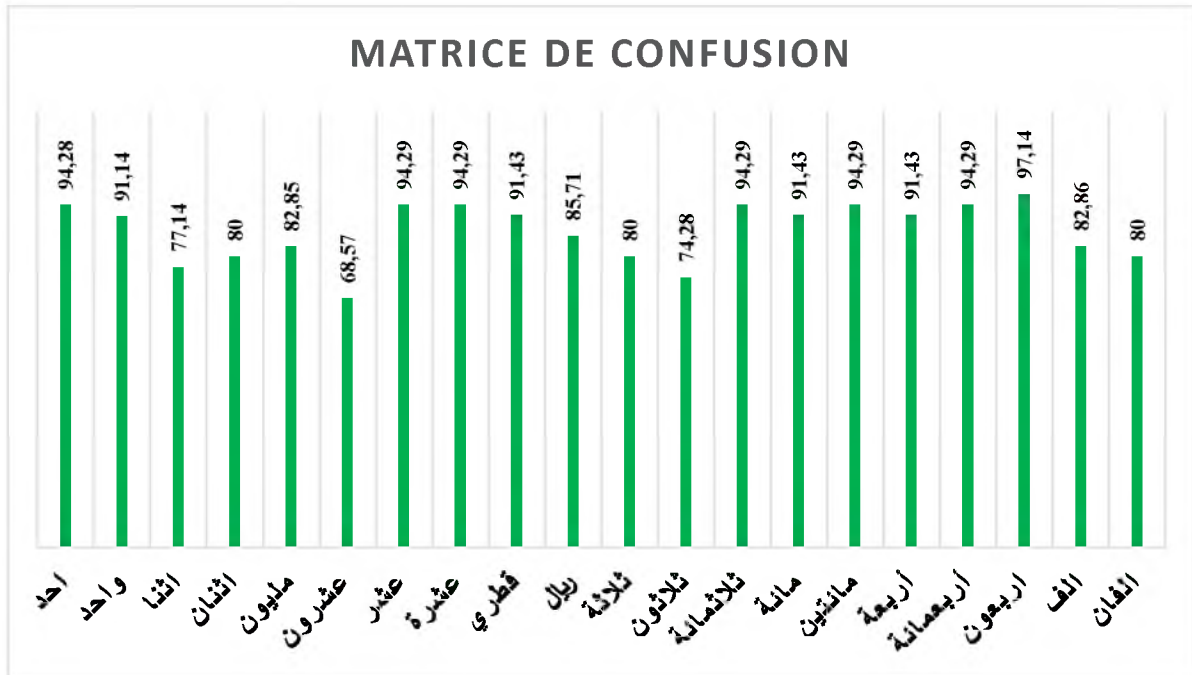


Figure 3. 11: Matrice de confusion obtenue par le système, évalué sur la base de montants littéraires arabe manuscrits

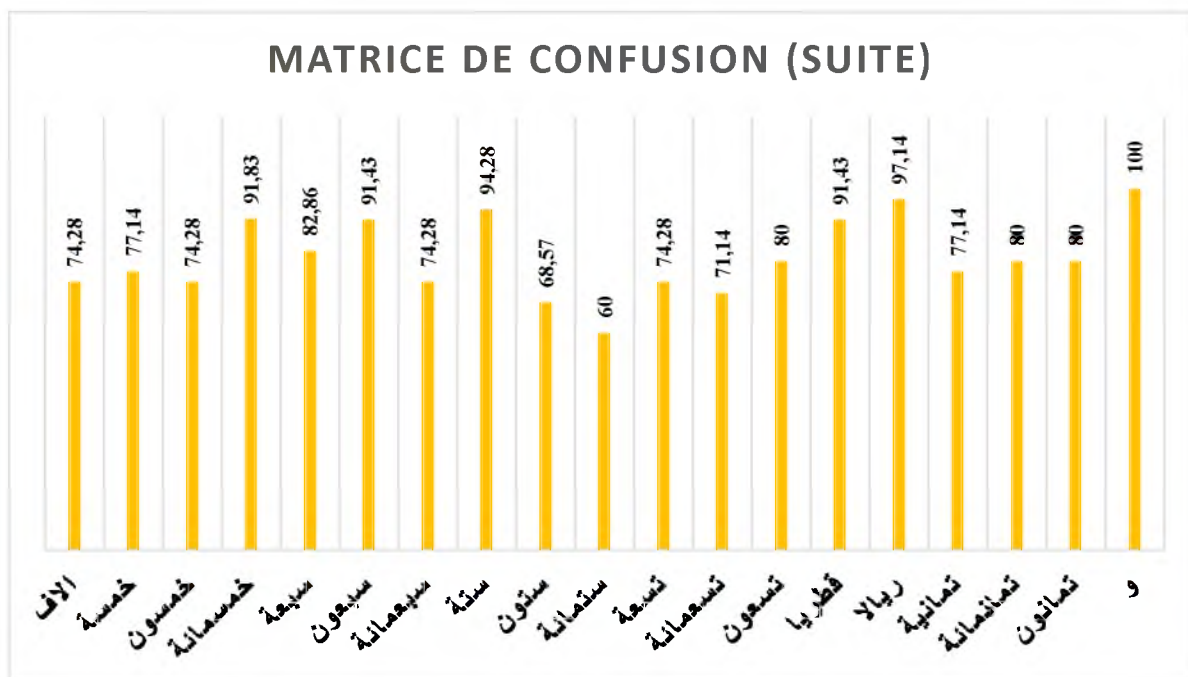


Figure 3. 12: Matrice de confusion obtenue par le système, évalué sur la base de montants littéraires arabe manuscrits (suite).

Nous permet de faire trois remarques importantes.

En premier lieu, on remarque que tous les montants ont été connues au moins 74 % sauf le montant « و », qui a 100% de la reconnaissance.

En second lieu, le taux d'erreur est très important pour certains montants ayant une grande similarité morphologique. Par exemple, le montant (عشرون) est confondu 6 fois avec le montant (ستون), 1 fois avec le montant (عشرة). Un autre exemple, le montant (تسعمائة) est confondu 6 fois avec le montant (ستمائة), 1 fois avec les deux montants (تسعون, تسعة).et aussi le montant (ستمائة) est confondu 7 fois avec le montant (تسعمائة), 1 fois avec le montant (ثمانمائة), 1 fois avec le montant (ستون). Dernier exemple le montant (ستون) est confondu 2 fois avec le montant (مليون), 1 fois avec les deux montants (خمسون, عشرون).

En troisième lieu, certaines confusions entre quelques montants sont absurdes, du fait qu'il n'y ait aucune ressemblance entre ces montants, par exemple celle entre le montant (عشرون) et le montant (خمسائة), celle entre le montant (ستمائة) et le montant (ثلاثة) et celle entre le montant (ستمائة) et le montant (خمسائة), celle entre le montant (ستون) et le montant (سبعمائة). Ces types d'erreurs expliquent que les caractéristiques utilisées sont insuffisantes pour modéliser les montants littéraux.

Les erreurs les plus élevées de confusion entre certains montants sont illustrées dans le Tableau 3.3.

	ستون	مليون	خمسائة	عشرة	سبعون	تسعة	تسعون	ثمانمائة	ستمائة	تسعمائة	ثلاثة	ستون	عشر	خمسون	عشرون	سبعمائة
عشرون	6	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0
تسعمائة	0	0	0	0	0	1	1	1	6	0	1	0	0	0	0	0
ستمائة	1		3		1	0	0	1	0	7	1	0	0	0	0	0
ستون	0	2	0	0	0	0	0	0	0	0	0	0	2	1	1	5

Tableau 3.3 : Matrice de confusion pour certains montants

3.5.3. Résultats de reconnaissance des montants littéraux après la segmentation :

Nous avons fait la classification sur 197 images dans l'ensemble de test et les résultats sont résumés dans le Tableau 3.4 Ci-dessous.

Classification des mots	Le nombre de mots segmenté correct	Reconnaissance correcte	précision	Erreur
	508	452	88,97%	11.03%

Tableau 3. 4: Résultats général de la classification

3.6. Conclusion

Dans ce chapitre, nous avons présenté un système à base des SVM pour la reconnaissance des montants littéraux arabes manuscrits, en nous basant sur plusieurs traitements : Binarisation, extraction de caractéristiques et finalement la décision. Le système a été testé sur la base standard est donne un résultat de 83,53%.

Nous avons présenté une méthode efficace pour la segmentation des mots manuscrits arabes. La méthode a été testée sur 135 images contient 562 mots. Le résultat de la segmentation à donner un taux de 90.39%. Les causes d'erreurs sont principalement dues à la grand espace entre les sous mots, et petit espace entre les mots.

Nous avons testé la classification sur les images segmentés correct nous avons obtenu un résultat de 88,97%, nous constatons que le chevauchement entre l'écriture arabe manuscrite entraîne un faible taux de reconnaissance.

Conclusion générale

Le problème de la reconnaissance de l'écriture arabe manuscrite non-contrainte est complexe. Dans un premier lieu, il faut trouver une modélisation qui tire profit de ses propriétés bidimensionnel intrinsèques. De plus, cette modélisation doit prendre en compte des erreurs qui peuvent affecter les données qui arrivent souvent entachées de bruit dû à l'acquisition ou aux prétraitements (ex., binarisation) qui se trouvent en amont du système de reconnaissance.

La modélisation de l'écriture manuscrite peut se faire à travers deux approches totalement différentes : une approche globale ou bien une approche analytique.

Dans notre système, nous avons opté pour une stratégie *Une-Contre une* pour décider de la classe d'appartenance de chaque mot. Cette stratégie consiste à construire autant de SVM que de classes, et à entraîner chaque classifieur SVM pour séparer une classe i des classes restantes.

Les fonctions noyaux utilisées et leurs paramètres sont déterminés de façon expérimentale. La mise en œuvre des SVM est très simple, cependant la difficulté majeure liée à l'utilisation de ces classifieurs était l'adaptation des hyper-paramètres conditionnant le processus d'apprentissage et la variable de compromis C .

D'après le premier jeu de tests effectués, la majorité des erreurs de reconnaissance peuvent être attribuée à la phase d'extraction de caractéristiques où les méthodes employées donnent, dans certains cas, des fausses alertes. Donc, si on pourra dans le futures restent envisageables de Vérifier d'autres types de caractéristiques pour améliorer encore plus le taux de la reconnaissance.

Bibliographie

- [1]- Azizi Rebiai, « Une approche hybride pour la reconnaissance d'écrite arabe manuscrite », Thèse de Magister de l'université de de Constantine, 2006/2007.
- [2]- A. Amin ,W. Mansoor, "Recognition of Printed Arabic Text using Neural Networks", Proc. 4th Int. Conf. on Document Analysis Recognition, Ulm, Germany, 1997.
- [3]- N.Essoukri Ben Amara, Utilisation des Modèles de Markov Cachés Planaires en Reconnaissance de l'Ecriture Arabe Imprimé, Thèse de Doctorat, Ecole National d'Ingénieur de Tunisie, Février 1999.
- [4]- Maged Mohamed Mahmoud Fahmy, « Automatic Recognition Of Handwritten Arabic Characters Using Their Geometrical Feature », 2000.
- [5]- Farès Menasri, "Contributions à la reconnaissance de l'écriture arabe manuscrite," UNIVERSITE PARIS DESCARTES, Thèse de doctorat 2008.
- [6]- Oivind Due Trier and Torfinn Taxt, "valuation of binarization methods for docu-," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3), p. 312–315, 1995.
- [7]- R. M. Bozinovic, S. N. Srihari. Off-line Cursive Script Word Recognition. *IEEE trans. On Pattern Analysis and Machine Intelligence*, vol. 11, no. 1 ; 1989.
- [8]- F. Grandidier. *Un nouvel algorithme de sélection de caractéristiques – application à la lecture automatique de l'écriture manuscrite*. PhD thesis, Univ du Québec, 2003
- [9]- E. Thiel, "Les distances de chanfrein en analyse d'images : fondements et applications," Grenoble 1, Thèse de doctorat 1994.
- [10]- F. Carmagnac, "Classification supervisée et semi-supervisée : contributions à la classification d'images de documents," PhD thesis 2003.
- [11]- B. Al-Badr and S.A. Mahmoud, "Survey and bibliography of Arabic optical text recognition," *Signal processing*, vol. 41, pp. 49-77, 1995.
- [12]- A. Belaid. (1997) Analyse de documents : de l'image à la représentation par les normes de codage.
- [13]- A. Bennasr, A. Zahour, and B. Taconet, "Extraction des lignes d'un texte manuscrit arabe," *Vision interface*, 1999.
- [14]- R.G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 7, july 1996.
- [15]- M. Lemaitre, "Approche markovienne bidimensionnelle d'analyse et de reconnaissance de documents manuscrits", thèse de doctorat, Université Paris 5 René Descartes, 2007.

- [16]- I. Trier, A. Jain and T. Taxt, "Feature extraction methods for character recognition - A survey", *Pattern Recognition*, vol.29 (4), pp.641-662, 1995.
- [17]- D. Zhang and G. Lu, "Review of shape representation and descriptions techniques", *Pattern recognition*, vol.37 (1), pp.1-19, 2004.
- [18]- M. Delalandre, E. Trupin, and J.-M. Ogier, "Analyse structurelle en interprétation de documents : un bref survol", *International Conference on Image and Signal Processing*, pp.640-649, 2003.
- [19]- M. Shridhar and A. Badreldin, "Recognition of isolated and connected handwritten numerals", *Proc, IEEE International Conference on Systems, Man and Cybernetics*, pp.142-146, 1984.
- [20]- X. Dupre, "Contributions à la reconnaissance de l'écriture cursive à l'aide de modèles de Markov cachés", PhD thesis, Univ Rene Descartes - Paris V, 2003.
- [21]- B. Verma, "A contour code feature based segmentation for handwriting recognition", *ICDAR'03*, pp.1038-1042, 2003.
- [22]- A. Senior, "Offline handwriting recognition using recurrent neural networks", PhD thesis, Cambridge University, 1994.
- [23]- P. Dargenton, "Contribution à la segmentation et à la reconnaissance de l'écriture manuscrite par l'ordinateur", Thèse de doctorat: INSA Lyon, 224pp, Décembre 1994.
- [24]- Y. Yamashita, K. Higuchi, Y. Yamada, and Y. Haga, "Classification of handprinted Kanji characters by the structured segment matching method", *Pattern Recognition Letters*, Vol.1, pp.475-479, 1983.
- [25]- C. -L. Liu, Y. -J. Liu, and R. -W. Dai, "Preprocessing and statistical/structural feature extraction for handwritten numeral recognition", In A. C. Downton and S. Impedovo, editors, *Progress of Handwriting Recognition*, World Scientific, Singapore, pp.161-168, 1997.
- [26]- A. Alper Atici, F. T. Yarman-Vural, "A heuristic algorithm for optical character recognition of Arabic script", *Signal Processing*, vol.62 (1), pp.87-99, October 1997.
- [27]- R. Kapoor, D. Bagai and T. Kamal, "Representation and extraction of nodal features of DevNagri letters", *ICVGIP*, 2003.
- [28]- A. Rachidi, M. El Yassa and D. Mammass, "A pretopological approach for handwritten isolated Arabic characters recognition", *Conférence ISCCSP'06, Marrakech*, 13-15 Mars 2006.

- [29]- T. Rath, V. Lavrenko and R. Manmatha, "A statistical approach to retrieving historical manuscript images without recognition", Tech. rep., Center for Intelligent Information Retrieval technical, 2003.
- [30]- R. El-Hajj, "Reconnaissance hors ligne de textes manuscrits cursifs par l'utilisation de systèmes hybrides et de techniques d'apprentissage automatique", Thèse de doctorat, ENST, Paris, Juillet 2007.
- [31]- R. El-Hajj, L. Likforman-Sulem, C. Mokbel, "Arabic handwriting recognition using baseline dependent features and Hidden Markov Modeling", ICDAR 05, Seoul, Corée du Sud, 2005.
- [32]- R. El Ayachi, K. Moro, M. Fakir, B. Bouikhalene, "On the Recognition of Tifinaghe Scripts", Journal of Theoretical and Applied Information Technology, vol.20 (2), pp.61-66, 2010.
- [33]- S. Touj, N. B. Amara and H. Amiri, "Generalized Hough transform for Arabic optical character recognition", ICDAR'03, pp.1242-1246, 2003.
- [34]- C. Schmid, "Appariement d'images par invariants locaux de niveaux de gris", PhD thesis, INPG, 1996.
- [35]- A. Belaïd et Y. Belaïd, *Reconnaissance des formes méthodes et applications*, InterEdition, 1992.
- [36]- 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.
- [37]- Thorsten Joachims, « Making Large-Scale SVM Learning Practical » University of Dortmund, Computer Science Department, 1998.
- [38]- Robert Azencott, « Application des SVM pour la discrimination de contenus musicaux » Cours d'introduction aux théories de l'apprentissage, Paris, le 31 janvier 2005.
- [39]- Y. Al-Ohali, Mohamed Cheriet, and Ching Y. Suen, "Databases for recognition of handwritten arabic cheques," *Pattern Recognition*, vol. 36, no. 1, pp. 111-121, 2003.
- [40]- Jawad H AlKhateeb, Jianmin Jiang, Jinchang Ren, and Stan S Ipson, « Component-based Segmentation of Words from Handwritten Arabic Text», International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:2, No:5, 2008.
- [41]- Fadoua BOUAFIF SAMOUD, Samia SNOUSSI MADDOURI, Kamel HAMROUNI « Segmentation de chèques bancaires arabes», International Conference: Sciences of Electronic, Technologies of Information and Telecommunications March 27-31, 2005.

- [42]- H.Mohamadally, B.Fomani « SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges » Versailles St Quentin, France 16 janvier 2006.
- [43]- 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
- [44]- chapitre 11 « Classifieurs SVM et Réseaux de Neurones pdf ».
- [45]- L. Hamel. "Knowledge discovery with support vector machines". Wiley Edition, 2009.
- [46]- A. Shigeo. "Support Vector Machines for Pattern Classification". Springer-Verlag London Limited, 2005.
- [47]- S. Knerr, L. Personnaz, J. Dreyfus, et al. « Single-layer ll learning revisited: A stepwise Procedure for building and training a neural network". Optimization Methods and Software, 23–34, 1990.
- [48]- Duda, R. O., P. E. Hart et D. G. Stork, " *Pattern Classification*", John Wiley and Sons Inc, 2001.
- [49]- B. Scholkopf and A.J. Smola. "Learning with Kernels Support Vector Machines", Regularization, Optimization, and Beyond. MIT Press, 2002.
- [50]- P.Dargento, N.Vincent, H.Emptoz Construction d'un graphe structurel représentatif d'une forme. ICOHD'93, Sixth International Conference on Handwriting and Drawing, P Conference on Handwriting and Drawing, p. 231-233, 1993
- [51]- J.Caesar, B.Gloger, Pre-processing and feature extractionfor a handwriting recognition system, second international conference on document analysis and recognition, pp.408-411, 1993.