وزارة التعليم العالي والبحث العلمي

BADJI-MOKHTAR– ANNABA – UNIVERSITY
UNIVERSITE BADJI –MOKHTAR – ANNABA

جامعة باجي مختار- عنابة

FACULTÉ DES SCIENCES DE L'INGÉNIEUR
DÉPARTEMENT D'INFORMATIQUE

كلية علوم المهندس
قسم الإعلام الآلي

ANNEE UNIVERSITAIRE 2006

سنة جامعية 2006

MEMOIRE

Présenté en vue de l'obtention du diplôme de **MAGISTER**

THEME

RECONNAISSANCE DE L'ECRITURE ARABE MANUSCRITE A BASE DES MACHINES A VECTEURS DE SUPPORTS

Option
Intelligence Artificielle Distribuée (IAD)

Par : Mehennaoui Zahra

DIRECTEUR DE MEMOIRE : Professeur SELLAMI Mokhtar

DEVANT LE JURY

PRESIDENT	:	Khadir M.T, MC	Université de Annaba
RAPPORTEUR	:	Sellami M, Pr	Université de Annaba
EXAMINATEURS	:	Baatouche M, Pr	Université de Constantine
EXAMINATEURS	:	Kholladi M.K, MC	Université de Constantine
EXAMINATEURS	:	Fareh N, MC	Université de Annaba

Table Des Matières

Liste Des Figures

Liste Des Tableaux

Liste Des Symboles

SVM	Support Vector Machines (Machines à vecteurs de suppurt)
RBF	Noyau à Base Radial
VC	Dimension de Vapnik-Chervonenk
x	Vecteur d'entrée arbitraire
y	Valeur de sortie associée à x
l	Nombre d'exemple d'apprentissage
f	Fonction de décision du SVM
w	Valeur de biais du SVM
b	Paramètres du SVM
M	Marge de séparation
F	Espace augmenté
R	Rayon de la petite hyper-sphère englobant les données dans F
h	Capacité du classifieur
$L(w,b,\alpha)$	Lagrangien primaire du SVM
α	Ensemble de multiplicateurs α_i de Lagrange
W(w)	Lagrangien dual
ξ_i	Slack variables (variable de relaxation)
C	Paramètre de régularisation
N_F	Dimension de l'espace augmenté F
d	Degré du noyau polynomial
$\phi(x)$	Image de x dans l'espace augmenté F
K(x,.)	Noyau de Mercer
KTT	Conditions de Karush-Kuhn-Tucker
SMO	Sequentiel Minimisation Optimisation (Optimisation par minimisation séquentiel)
$R(\alpha)$	Risque réel
$R_{emp}(\alpha)$	Risque empirique
P(x,y)	Probabilité d'observation du couple (x,y)

AOCR	Arabic optical character recognition
OCR	Optical character recognition
ASCII	American standard code for information interchange
ASMO	Arabic standard metrology organisation

ملخـــــص

المعالجة الآلية للكتابـــة و بالأخص الكتابة باليد تشكل تحدي العصر. طرق عديـدة استعملت لحـل هذه المشكلة، هذه الطرق أعطت نتائج ملفتة للانتباه. للأسف، الطرق الكلاسيكية تعتمد على تخفيض الخطأ التجريبي و تعاني من مشكلة "فوق الحفظ" (sur-apprentissage) و العدد الكبير للعوامل المعرفة من طرف المستعمل. مـن أجل حل هّذه المشاكل. ظهر إتـجاه جـديد في ميدان التعلم الإحصائي(Apprentissage Statistique) مع نظرية فابنيك و الآلات الحاملة للأشعة(Machines à Vecteurs de Supports) وتطبيقاتهم في ميدان التعرف على الأشكال.

العمل المقدم في هذه المذكـــرة يدخل في مجال المعالجة الآلية للكتابة العربية باليد و يعمل على ضرورة تجريب طريقة جديدة في ميدان التعلم الإحصائي"الآلات الحاملة للأشعة" من أجل معرفة الحروف العربية المكتوبة باليد.

قبل تحديد المجموعة التي ينتمي إليها الحرف، هذا الأخير يتطلب مجموعة من المعالجات: تحويل الصورة إلى صورة ذات لونين، تنظيف، تحديد الأبعاد، استخراج الخصائص. من أجل وصف صورة الحرف، استعملنا مجموعة من الخصائص الإحصائية المستخرجة من توزيع نقاط الصورة. استعملتنا مجموعة ثانية من الخصائص تأخذ بعين الاعتبار الأشكال الهندسية لـلحروف العربية. يجمع النظام المقترح مجموعة من الآلات الحاملة للأشعة حسب الإستراتجية "واحد ضد الكل". كل آلة تختص في فصل كل مجموعة عن باقي المجموعات.

RESUME

La reconnaissance de l'écriture, et particulièrement l'écriture manuscrite reste un défi d'actualité. Différentes techniques de reconnaissance de formes ont été utilisées pour la résolution de ce problème, certaines ont donné des résultats remarquables. Malheureusement, les techniques classiques se basent sur le principe de minimisation du risque empirique et souffrent des problèmes de sur-apprentissage et du grand nombre de paramètres à fixer par l'utilisateur. Pour tenter de résoudre ces problèmes, une nouvelle direction dans le domaine de l'apprentissage statistique a émergé de la théorie de Vapnik et les machines à vecteurs de support et leurs applications dans le domaine de la reconnaissance des formes.

Le travail présenté dans ce mémoire s'intègre dans le cadre général de la reconnaissance automatique de l'écriture arabe manuscrite, et répond à la nécessité d'expérimenter une nouvelle méthode d'apprentissage : les machines à vecteurs de support (SVM : Support Vectors Machines), appliqué à la reconnaissance des caractères arabes manuscrits.

Avant de décider la classe d'appartenance du caractère en entrée, il est nécessaire d'effectuer un certain nombre de traitements : binarisation, lissage, normalisation et extraction de contour. Afin de caractériser nos images de caractères arabes, nous avons opté pour une combinaison entre des caractéristiques statistiques provenant de la distribution des pixels, et des caractéristiques structurelles basées sur les motifs géométriques de l'alphabet arabe.

Le système proposé combine, selon le schéma un contre tous, plusieurs SVM, spécialisés, chacun, dans la séparation d'une classe des classes restantes.

Mots clés :, reconnaissance de l'écriture arabe manuscrite, Apprentissage statistique Machines à vecteurs de support, SVM, fonction noyau.

ABSTRACT

Word recognition, and specially handwritten word recognition remains a challenge. Different techniques of pattern recognition have been used to solve this problem, some of them made remarquable results. Fortunately, classical techniques are based on the principal of empiric risk minimisation and suffer of two problems: the problem of over-learning and the problem of multiple parameters to be fixed by the user.

Trying to solve these problems, a new direction in the field of statistic learning emerged from the Vapnik theory and support vectors machines and their application to pattern recognition.

The work we are presenting here is part of a hole: automatic handwritten recognition, it is a response to the necessity of experimenting a new learning method: SVM Support Vectors Machines, applied to handwritten Arabic character recognition.

Before deciding the class of a character, it is necessary to perform some pre-processing operations: binarization, smoothing, normalisation and contour extraction. Features extracted from the character image are a combination of two types: statistical features, calculated from pixel distribution and structural features, based on geometric caracteristics of the artic alphabet. The proposed system, combine, according to scheme one against others, many SVMs, each of them will separate every class from the remaining others.

Keywords: Arabic handwritten recognition, Statistic learning, Support vectors machines, SVM, Kernel function.

REMERCIEMENTS

Au terme de ce travail, je voudrai exprimer ma profonde gratitude envers dieu tout puissant qui, grâce à son aide, j'ai pu finir mon travail.

Mes remerciements les plus vifs vont à monsieur Mokhtar Sellami, Professeur à l'université Badji Mokhtar, Annaba, pour avoir accepté de diriger mon travail et la confiance qu'il m'a témoigné. Les mots vont ni me suffire, ni pouvoir exprimer ma gratitude envers lui, parce qu'il était non seulement mon encadreur, mais un fleuve intarissable de valeurs humaines. Je ne le remercierais jamais assez pour ses conseils, son orientation et surtout son « forcing »

Je suis très reconnaissante à Monsieur M.T.Khdir, Maître de conférence à l'université de Annaba, d'avoir accepter de présider le jury.

Mes remerciements les plus intenses sont adressés à monsieur M.Baatouche, professeur à l'université de Constantine, M.K.Kholladi, maître de conférence à l'université de Constantine, monsieur N.Khadir, maître de conférence à l'université de Annaba, pour l'honneur qu'il m'ont fait en acceptant de faire partie de ce jury et examiner ce travail.

Je tiens également à exprimer toute ma reconnaissance à madame Souici-Meslati.L pour ses critiques constructives et sa disponibilité à tout moment.

Je tiens également à exprimer ma profonde gratitude à monsieur Benouareth, pour son aide tout au long de ce travail, et à tous les membres de LRI.

Merci à mes parents, pour leur amour, soutien, et encouragements, qu'ils trouvent ici ma profonde gratitude et mon grand respect.

Un grand merci à Loubna, pour son aide, sa disponibilité et son humeur.

Merci à mes amies et mes collègues, pour leur soutien indéfectible.

DEDICACE

A la mémoire de ma très chère grande-mère, à la mémoire de mon grand-père. Vous n'êtes plus à mes cotés, mais vous resterez éternellement dans mon cœur.

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

A ma sœur et mes frères

A toute ma famille

A tous mes amis (es)

A ceux qui me sont chères.

Introduction Générale

Les techniques liées aux traitements de l'information connaissent actuellement un développement très actif en liaison avec l'information et présentent un potentiel de plus en plus important dans le domaine de l'interaction *Homme-Machine.* L'homme veut communiquer avec l'ordinateur avec la façon la plus simple, la plus naturelle pour faciliter et accélérer l'interaction et l'échange d'informations. Il cherche à rendre ces machines accessibles par la voix, capables de lire, de voir, de traiter et d'analyser rapidement l'information reçue.

Ecrire pour communiquer a été de tous les temps une préoccupation première de l'homme. L'écrit a été, et restera, l'un des grands fondements des civilisations et le mode par excellence de conservation et du transmission du savoir. En effet, beaucoup d'objets qui nous entourent comportent des traces écrites : les panneaux indicateurs, les notices d'emploi des produits, les journaux, les livres, les formulaires…etc.

Rendre la machine capable de lire permettrait de saisir les informations de manière plus aisée et de traiter les documents de façon plus rapide. La reconnaissance de l'écriture a aujourd'hui investi une multitude de domaines dans le monde. Nous pouvons citer : l'automatisation de la saisie des formulaires administratifs, du tri postal et de l'échange, la vérification et la lecture des chèques.

Le problème de la reconnaissance fiable de l'écriture manuscrite est encore un défi majeur. Beaucoup de problèmes contribuent à rendre cette tâche très compliquée, la qualité et la complexité du fond du document en est un, auquel il faut ajouter la diversité et la variabilité de l'écriture aussi bien d'un point de vue intra-scripteur, en fonction des conditions (vitesse, humeur,…), que d'un point de vue inter-scripteurs.

Pour l'instant, et bien que la recherche dans ce domaine se poursuive depuis plus de trente ans, le problème de la lecture automatique d'écriture cursive n'est toujours pas résolu. Il semble cependant que la reconnaissance d'écriture cursive ait un rôle important à jouer

dans les systèmes futurs de reconnaissance et donc, que ce domaine de recherche soit toujours d'actualité.

A l'heure actuelle, les problèmes de l'écriture latine manuscrite contrainte sont partiellement résolus, et la lecture automatique de l'imprimé a fait une grande avancée dans beaucoup de domaines. La recherche dans ce domaine s'oriente vers l'analyse de documents beaucoup moins contraints que ceux traités jusqu'à présent.

Contrairement au latin, la reconnaissance de l'écriture arabe manuscrite ou imprimée reste encore aujourd'hui au niveau de la recherche et de l'expérimentation, le problème n'est pas encore résolu même si dans certaines applications à vocabulaires limité et en mono-fonte, des résultats appréciables sont communiqués. Les travaux sont généralement axés sur la méthodologie de développement plutôt que sur la réalisation d'un produit fini commercialisable, qui reste encore au stade de rêve.

Le retard de l'écriture arabe par rapport à l'écriture latine peut être attribué à la complexité morphologique de l'alphabet arabe. De plus, le manque de protocoles communs de validation et de tests contribue énormément à ce retard.

Les recherches sur la reconnaissance de l'écriture arabe datent des années 80. Depuis, les recherches se sont multipliées dans ce domaine. Durant ces dernières décennies, plusieurs approches et méthodes ont été proposées par les chercheurs, dans le but, d'améliorer les taux de reconnaissance. C'est dans ce cadre que s'inscrit notre travail.

La variabilité de l'écriture manuscrite permet de confronter les algorithmes de classification et d'apprentissage à des problèmes difficiles et réalistes. La plupart des classifieurs classiques ont donné des résultats remarquables dans ce domaine, notamment les réseaux de neurones. Mais la nécessité de performances élevées dans des applications réelles, a poussé la recherche vers des modèles de classifications de plus en plus complexes. Plusieurs générations de machines d'apprentissage ont vu le jour dans le but de classifier, de catégoriser ou de prédire des structures particulières dans les données. La classification des caractères constitue, par ailleurs, la principale application des machines d'apprentissage, dont les différents travaux ont permis une nouvelle appréhension des modèles de classification.

Notre travail s'intègre dans le cadre de la reconnaissance de l'écriture arabe manuscrite et vient en complément aux actions de recherche menées par l'équipe RADAR (Reconnaissance te Analyse de Documents Arabes) au sein de LRI-Annaba. Nous

proposons un système qui se base sur l'utilisation d'une nouvelle méthode d'apprentissage : les machines à vecteurs de support (SVM : Support Vectors Machines), pour la reconnaissance des caractères arabes manuscrits.

Le SVM est un modèle discriminant qui tente de minimiser les erreurs d'apprentissage tout en maximisant la marge de séparation. La maximisation de la marge est une méthode de régularisation qui réduit la complexité du classifieur. Les SVM représentent une méthode de classification bien adaptée pour traiter des données de très hautes dimensions telles que les textes et les images. La formulation des SVM laisse très peu de place aux paramètres fixés par l'utilisateur. De plus, il s'agit d'un problème d'optimisation quadratique (convexe) sous contrainte, ce qui est assez complexe (en termes algorithmiques) mais donne des garanties sur la convergence à un optimum global, contrairement à l'optimisation numérique d'une fonction de coût non quadratique.

Dans le présent travail, nous développons une méthodologie de reconnaissance des lettres arabes manuscrites, en nous basant, d'une part, sur la nature statistique du classifieur SVM, et, d'autre part, sur les caractéristiques morphologiques de l'écriture arabe manuscrite. Ce problème est un problème multi-classes, alors que le SVM est un classifieur binaire qui ne traite habituellement que des données binaires. Pour cette raison, nous avons proposé un système qui combine plusieurs SVM, chacun se spécialisant dans une partie du problème.

Le principe de chaque SVM consiste à projeter les données de l'espace d'entrée non linéairement séparables dans un espace de plus grande dimension *(Feature space)*, de façon que les données deviennent linéairement séparables. Dans cet espace, on construit un hyperplan séparant les classes d'entrées.

Le manuscrit est structuré en quatre chapitres, dans ce qui suit nous donnons une brève description de leurs contenus respectifs:

Chapitre 1. Reconnaissance Optique des Caractères

Ce chapitre est consacré à la représentation des concepts généraux liés à la reconnaissance de l'écriture manuscrite, il fait un état de l'art dans ce domaine. Après avoir présenté les différents aspects liés à la reconnaissance de l'écriture et qui peuvent influencer la complexité des systèmes OCR, nous décrivons les caractéristiques morphologiques de l'écriture arabe. Par la suite, les étapes intervenant dans un système de reconnaissance de l'écriture sont données. A travers ces étapes, quelques méthodes proposées dans la

littérature sont citées. Le chapitre se termine par les difficultés liées à l'OCR arabe (AOCR). Suivant les étapes chronologiques d'un système OCR général, nous présentons une synthèse des principales particularités qui compliquent la tâche de l'AOCR. A la fin du chapitre, un tableau synthétique regroupant quelques systèmes de reconnaissance de l'écriture arabe est exposé.

Chapitre 2. Approches de Classification et Apprentissage Statistique

La première partie du deuxième chapitre est consacrée à la présentation des différentes approches de classification. Nous donnerons un aperçu sur les principales méthodes de classification utilisées en reconnaissance de l'écriture, particulièrement, mais issues de méthodes générales de classification en reconnaissance de formes. La deuxième partie du chapitre met l'accent sur l'apprentissage statistique, principalement basé sur la théorie de Vapnik-Chervonenkis. L'apprentissage statistique apparaît dans plusieurs domaines, mais dans le cadre de notre travail, nous nous intéressons principalement au problème d'apprentissage pour la reconnaissance des formes. Nous présentons deux points essentiels dans la théorie des machines à vecteurs de support : *la dimension VC* et la *Minimisation du Risque Structurel.*

Chapitre 3. Les Machines à Vecteurs de Support

Dans le troisième chapitre, nous poursuivons la description des machines à vecteurs de support. Après avoir présenté un bref historique des SVM, nous décrivons leur utilisation pour un problème de classification. Nous présentons la méthode générale de construction de l'hyperplan de séparation entre les données. Nous traitons le cas des données séparables ainsi que le cas des données non séparables. Par la suite, l'extension au cas non linéaire est décrite en montrant l'intérêt de l'utilisation des fonctions noyaux et la projection des données dans un espace augmenté. Quelques algorithmes d'apprentissage pour les SVM sont ensuite présentés. Le chapitre se termine par la description de quelques exemples de systèmes de reconnaissance de l'écriture à base des machines à vecteurs de supports.

Chapitre 4. Un Système à base des SVM pour la Reconnaissance des caractères Arabes Manuscrits

Dans ce chapitre, nous détaillons la méthodologie adoptée pour la conception et l'implémentation d'un système de reconnaissance des caractères arabes manuscrits à base des machines à vecteurs de support. Nous décrivons, en détail, les différentes phases

intervenant dans le système, ainsi que, les algorithmes choisis. Après avoir traité les images des caractères et extrait les caractéristiques décrivant ces images, la stratégie adoptée pour décider des classes d'appartenances des caractères est expliquée, puis nous décrivons la méthode de combinaison de plusieurs SVM pour un problème multi-classes. Nous présentons enfin la démarche suivie pour sélectionner les hyper-paramètres des SVM et les résultats obtenus.

Chapitre 1

Reconnaissance Optique des Caractères (OCR)

La reconnaissance de l'écriture relève du domaine de la reconnaissance des formes qui s'applique aux formes des caractères. L'objectif est d'attribuer à une forme un identifiant des prototypes des références déterminés dans une phase préalable.
Dans ce chapitre, nous rappelons quelques notions de l'OCR. Nous considérons les différents aspects et problèmes liés à la reconnaissance optique de l'écrit. Par la suite, nous présentons les différentes caractéristiques morphologiques de l'écriture arabe. Puis, nous donnons les différentes étapes intervenant dans un système OCR et nous terminons par une synthèse des particularités morphologiques de l'arabe et les difficultés liées à sa reconnaissance.

1. Introduction

Le but de la reconnaissance de l'écriture est de transformer un texte écrit en une représentation compréhensible par une machine et apte à être manipulée par les logiciels de traitements de textes [BEL 92]. Pour les écritures latines, le codage typique est opéré par le code ASCII (American Standard Code for Information Interchange), tandis que pour l'arabe on utilise généralement l'ASMO (Arabic Standard Metrology Organization).
La tâche de reconnaissance de l'écriture n'est pas triviale car les mots possèdent une infinité de représentations due au fait que chaque personne possède une écriture qui lui est propre, qu'il existe de nombreuses polices de caractères pour l'imprimé avec de nombreux

styles (gras, italique, souligné, etc.) et des mises en pages différentes et complexes. Tous ces facteurs rendent la reconnaissance de l'écriture bien complexe.

La reconnaissance automatique du texte manuscrit et imprimé est un domaine qui a rapidement intéressé les chercheurs avant même l'avènement des premiers ordinateurs et de nombreux articles ont été publiés à ce sujet [AMI 80-82, ALB 95, GOV 90, TAP 90]. Ces recherches avaient pour objectif de répondre à des besoins en matière d'automatisation du tri postal, de la lecture optique des chèques, de la traduction, et l'amélioration d'interfaces homme machine…etc.

A la base, l'intérêt pour la reconnaissance optique des caractères, connue plutôt sous la dénomination anglaise OCR (Optical Character Recognition), a commencé avec Tyring en 1900, de Albe en l'an 1912 et Thomas en l'an 1926 qui ont tenté de mimer l'interprétation humaine de l'information visuelle. Quelques années plus tard les premières expériences en reconnaissance de caractères ont pu être réalisées. Pendant les années soixante et soixante dix, les premiers systèmes de lecture automatique de texte imprimé ont vu le jour [BAL 70]. Toutefois, des systèmes fiables étaient restreints à quelques fontes seulement. Dans la même période, lindgrem dans [LIN 65, cité dans AYA 04], a introduit l'un des premiers systèmes de reconnaissance de l'écriture manuscrite. Les techniques de classification mises en œuvre durant cette période font appel à des méthodes d'appariement syntaxique (la plupart des méthodes statistiques n'étant pas connues encore). Entre les années 1980 et 1990, les réseaux de neurones, découverts deux décennies, plus tôt et trop vite abandonnés, suscitent un vif intérêt de la communauté des chercheurs grâce à l'algorithme de rétro-propagation du gradient. Ainsi, le perceptron multicouche a été rapidement reconnu comme le classifieur par excellence dans beaucoup de problèmes de reconnaissance de caractères.

Le degré de complexité des systèmes de reconnaissance de l'écriture est différent selon qu'il s'intéresse au manuscrit ou à l'imprimé. Si dans le premier cas, les problèmes ont été relativement résolus (cas des langues latines), et des études ont donné lieu à des systèmes commercialisés, la situation est complètement différente en ce qui concerne la reconnaissance des textes manuscrits et spécifiquement cursifs (cas de l'écriture arabe).

La reconnaissance de l'écriture arabe s'intègre dans le cadre général de la reconnaissance de l'écriture cursive, avec des spécificités et des problèmes qui lui sont propres. Contrairement à d'autres systèmes d'écriture (latine, chinoise, japonaise) peu de travaux ont été menés concernant la reconnaissance optique de textes arabes (AOTR). Le premier

travail publié sur la reconnaissance optique de textes arabes, remonte à 1975 [NAZ 75, cité dans ESS 99].

Un système de reconnaissance de l'écriture doit, idéalement, localiser, reconnaître et interpréter n'importe quel texte ou nombre écrit sur un support de qualité arbitrairement variable tels que les cartes, les formulaires, les agendas, les vieux manuscrits, etc. Ce type de système aide, dans ce cas, à convertir l'information contenue dans les vieux manuscrits pour être enregistrée dans des bases de données que tout le monde peut interroger et consulter à travers l'internet Par la suite, on s'est vite aperçu de la difficulté de réaliser des systèmes capables de reconnaître tout type de texte et on s'est tourné vers des systèmes dédiés, où l'on connaît à priori le style d'écriture que le système doit traiter. Par exemple, il existe des systèmes pour la lecture automatique des chèques bancaires, la reconnaissance du code de l'adresse postale, la lecture des formulaires d'impôts,…etc.

2. Les critères d'influences sur l'OCR

On classe souvent les méthodes de reconnaissance en fonction du mode d'acquisition de l'écriture :

- ***l'écriture en-ligne*** **:**(ou dynamique) est obtenue en saisie continue et se présente sous la forme d'une séquence de points ordonnée dans le temps avec un tracé est sans épaisseur [LOR 92]. Les systèmes en ligne prennent en compte l'information chronologique des mouvements du bras du scripteur. Cette information additionnelle augmente la précision bien que souvent coûteuse en temps de calcul. Dans ce cas, la donnée est de type signal où la reconnaissance est effectuée sur des données à une seule dimension ; l'approche doit tirer profit du lever du stylo et de la représentation temporelle. De plus, la réponse en continu du système permet à l'utilisateur de corriger et de modifier son écriture de manière interactive. L'analogie avec la reconnaissance de la parole est très fréquente et il n'est pas rare de voir des chercheurs appliquer des techniques issues de ce domaine [BEL 01]. L'acquisition du tracé est assurée généralement par une tablette graphique munie d'un stylo électronique.
- ***l'écriture hors-ligne :***(ou différée ou encore statique) est obtenue par la saisie d'un texte déjà existant, obtenue par un scanner ou une caméra. Dans ce cas, on dispose

d'une image binaire ou en niveaux de gris, ayant perdu toute information temporelle sur l'ordre des points. De plus, ce mode introduit une difficulté supplémentaire relative à la variabilité du tracé en épaisseur et en connectivité, nécessitant l'application de techniques de prétraitement. La figure 1.1 montre des exemples de données relevant de ces deux modes d'écriture. Le schéma de gauche montre une trajectoire de l'analyse du mouvement du tracé du mot *« sage »* par repérage des points importants. Le schéma de droite montre l'image du mot cursif *« dix »* en représentant ses pixels par des carrés noirs de même taille.

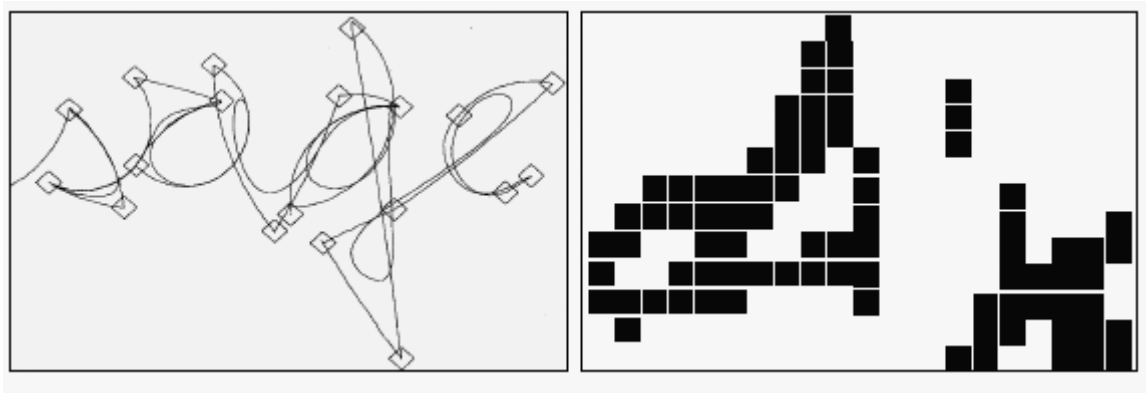


FIG. 1.1– *Différents modes de captures de mots : tracé à gauche du mot "sage" et image du mot "dix" à droite [BEL 01]*

Sans préjuger ici la difficulté d'un cas par rapport à l'autre, on peut seulement constater que dans le cas de la reconnaissance en ligne, les résultats sont souvent meilleurs pour des conditions similaires d'expérimentations (taille de vocabulaire, nombre de scripteurs, etc.) [BEL 01]. Cela vient des informations temporelles qui fournissent des connaissances précieuses sur la dynamique, la vélocité et la morphologie de l'écriture.

Il faut aussi faire une nette distinction entre texte imprimé et texte manuscrit, reconnaissance de caractères ou analyse de documents :

- ***Ecriture manuscrite ou imprimée :*** en alphabet latin, les mots d'un texte imprimé sont constitués de caractères séparés, et donc le problème de reconnaissance de texte peut se concevoir au départ comme un problème de reconnaissance de caractères. Pour un contexte manuscrit, l'écriture naturelle conduit à tracer des mots dont au moins une partie des lettres sont liées entre elles. Le problème de reconnaissance de l'écriture manuscrite sera donc plus un problème de reconnaissance de mots ou de

sous mots qu'un problème de reconnaissance de caractères. Pour l'écriture arabe, cette caractéristique est toujours présente que se soit pour le texte imprimé ou manuscrit, ce qui rend le problème de même complexité.

- ***Reconnaissance de caractères ou analyse de documents*** : dans le premier cas, la structure du texte est limitée à quelques lignes ou mots. La recherche consiste en un simple repérage des mots dans les lignes, puis à un découpage de chaque mot en caractères. Dans le second cas, il s'agit de données bien structurées dont la lecture nécessite la connaissance de la typographie et de la mise en page du document (structure physique et logique du document) [BEL 92]. Les fantaisies de présentation (comme dans certains journaux ou magazines), obligent parfois à reconnaître d'abord quelques caractères pour s'assurer du sens de la lecture. La tâche de lecture n'est pas ici un simple prétraitement, mais une démarche experte d'analyse de documents : localisation des régions, séparation des régions graphiques et photographiques, étiquetages sémantiques des zones textuelles à partir de modèles, détermination de l'ordre de lecture et de la structure du document.

Sur le plan méthodologique, l'évaluation de la complexité de certains problèmes doit prendre en compte plusieurs critères orthogonaux [LOR 92] :

- ***Disposition spatiale du texte :*** la classification du Tappert [TAP 90] indique que la représentation du texte peut subir deux types de contraintes :

1. *externes* conduisant à une écriture *pré-casée*, *zonée*, *guidée* ou *générale*
   - *pré-casée :* pour laquelle le scripteur doit s'efforcer d'écrire à l'intérieur des cases prédéfinies (ex, bordereaux)
   - *zonée :* où l'écriture doit s'effectuer dans des zones bien délimitées.
   - *guidée :* dans ce cas, le scripteur doit respecter la ligne de base
   - *générale :* correspondante à une écriture à emplacements libres
2. *internes* provenant des habitudes propres à chaque scripteur et conduisant à une écriture détachée, groupée, purement cursive ou mixte. Il est évident que l'écriture détachée reste la plus facile à réaliser du fait de la séparation quasi immédiate des lettres, au contraire de l'écriture cursive qui nécessite plus d'efforts du fait de l'ambiguïté des limites entre les lettres.

- ***Nombres de scripteurs :*** la difficulté de reconnaissance croit avec ce nombre, divisant l'échelle en trois ; mono, multi et omni-scripteurs. Le système le plus simple est sans

conteste un système de reconnaissance d'écriture mono-scripteur avec l'apprentissage de l'écriture propre à l'utilisateur considéré. En multi scripteurs, le système doit s'adapter à l'écriture d'un ensemble réduit de scripteurs potentiels, tandis qu'en omni-scripteurs, le système doit être capable de généraliser son apprentissage à n'importe quel type d'écriture. Pour un système mono-ou multi-scripteurs, il existe une phase *d'apprentissage initiale individualisé :* le système est capable de s'adapter et de tenir compte des particularités de l'écriture de chaque individu. De plus, si ce système est doté d'un mécanisme *d'apprentissage permanent,* il pourra s'adapter aux évolutions successives des écritures au cours du temps. Pour un système omni-scripteur, l'apprentissage ne peut être que très général. Il en résulte un ordre de grandeur de complexité très supérieur pour de tels systèmes [LOR 92].

- ***Taille du vocabulaire** :* on fait la différence entre les applications à vocabulaire limité (<100mots) et celles à vocabulaires très étendu (>100mots). Il est évident que dans le premier cas, la complexité est moindre, car la réduction du nombre limite l'encombrement mémoire et favorise l'utilisation de méthodes de reconnaissance directes et donc rapides, par balayage systématique de l'ensemble des mots du lexique. Dans le deuxième cas, des dizaines de milliers de mots formant un dictionnaire et pour lesquels se posent, à la fois des problèmes d'encombrement mémoire et de temps d'accès à chaque mot. Dans un tel cas, les seules méthodes efficaces envisageables sont les approches de type recherches arborescentes avec raffinements successifs [LOR 92]

Les degrés de généralité et de complexité d'un système de reconnaissance de l'écriture manuscrite peuvent être résumés par un schéma synthétique (figure 1.2) pour lequel l'origine des axes correspond à la fois au système le plus simple et le plus contraint, tandis qu'à l'opposé, la généralité et la complexité croissent au fur et à mesure qu'on s'en éloigne.

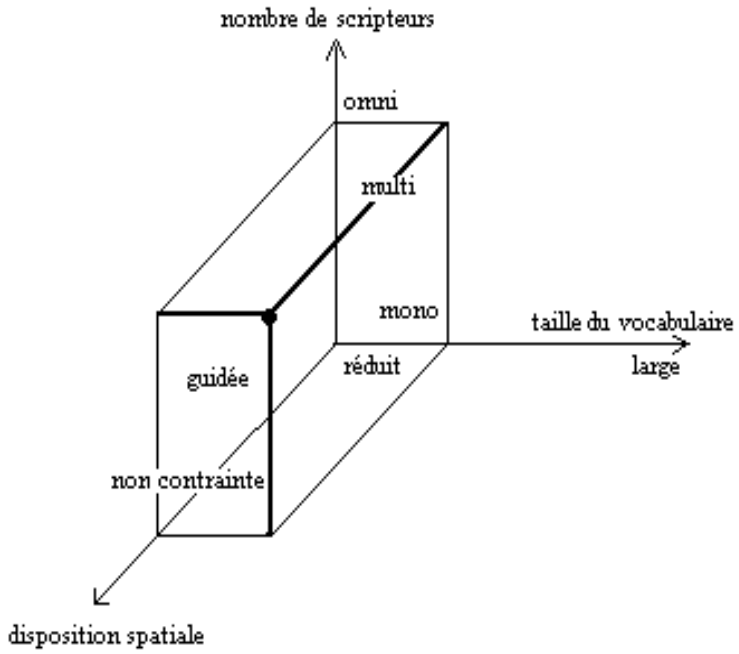


FIG. 1.2 – *Graphes de complexité des systèmes OCR*

D'autres types de critères peuvent influencer la complexité des systèmes d'OCR. Ils sont relatifs aux variations intrinsèques de l'écriture, dans le contexte d'écriture cursive.
Parmi ces variations, on peut noter :

- ***les déformations provoquées par les conditions matérielles et physiques dans lesquelles le texte a été écrit*** *:* par exemple, la position respective de la feuille, de l'avant bras du scripteur, l'angle du stylo par rapport à la direction de l'écriture peuvent influer sur la direction de la ligne de base, sur l'inclinaison de l'écriture, etc.
- ***Les variations propres aux scripteurs*** *:* (qui dépendent de son état physique et mental) traduisant le style personnel en terme de rapidité, de continuité et de régularité. Dans le cas des écritures rapides, plusieurs lettres peuvent fusionner en une forme globale différente, d'autres peuvent presque totalement disparaître. Tous ces éléments influent sur la forme des lettres (écriture penchée, bouclée, arrondie, linéaire, etc.) et bien sûr sur la forme des ligatures qui compromettent parfois le repérage des limites entre lettres.

La figure suivante présente une classification de différents types d'écriture selon la nature de l'écriture, le mode de saisie et l'application considérée.

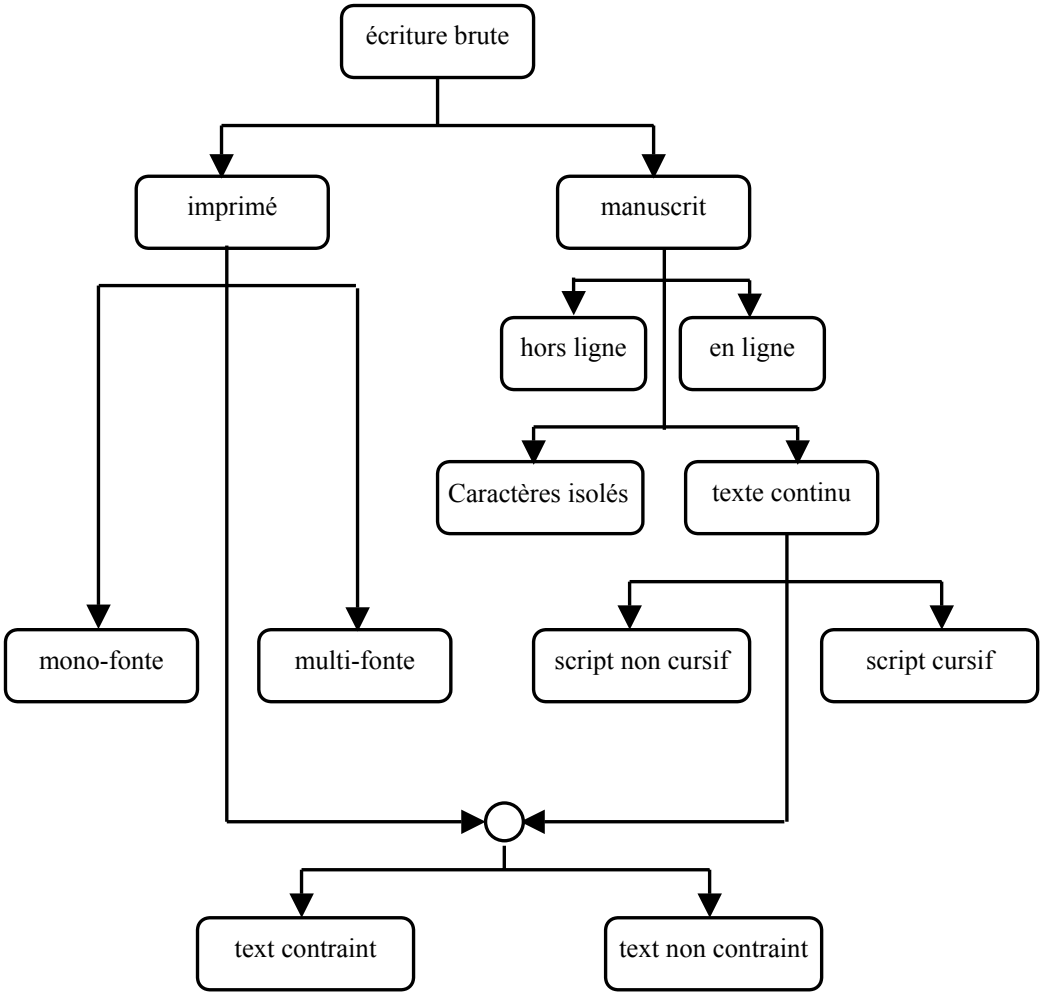


FIG. 1.3 – *Dichotomie des différentes écritures*

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, hébreu et hindou est décrite dans le tableau suivant :

Caractéristiques	Arabe	Latin	Hebreu	*Hindou*
justification	D à G	G à D	D à G	*G à D*
Cursivité	oui	non	non	*oui*
Diacritiques	oui	non	non	*oui*
Nombre de voyelles (vowels)	2	5	11	-
Nombre de lettres	28	26	22	40
Formes des lettres	1-4	2	1	1

TAB. 1.1 – : *Comparaison des caractéristiques de différentes langues [AMI 97]*

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 [ESS 99]. 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 [AMI 97]. 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 « ة » et « لا ». La considération du symbole « ~ » qui s'écrit uniquement sur le support du caractère « آ », fait apparaître d'autres graphismes.

caractère	Initiale	milieu	Finale	Isolé
Alif			ـا	ا
Ba	بـ	ـبـ	ـب	ب
Ta	تـ	ـتـ	ـت	ت
Tha	ثـ	ـثـ	ـث	ث
Nun	نـ	ـنـ	ـن	ن
Ya	يـ	ـيـ	ـي	ي
Jim	جـ	ـجـ	ـج	ج
Ha	حـ	ـحـ	ـح	ح
Kha	خـ	ـخـ	ـخ	خ
Dal			ـد	د
Thé			ـذ	ذ
Ra			ـر	ر
Za			ـز	ز
Waw			ـو	و
Sin	سـ	ـسـ	ـس	س
Chin	شـ	ـشـ	ـش	ش
Sad	صـ	ـصـ	ـص	ص
Dhad	ضـ	ـضـ	ـض	ض
Tad	طـ	ـطـ	ـط	ط
Dhad	ظ	ـظـ	ـظ	ظ
Ayn	عـ	ـعـ	ـع	ع
Ghyn	غـ	ـغـ	ـغ	غ
Fa	فـ	ـفـ	ـف	ف
Qaf	قـ	ـقـ	ـق	ق
Kaf	كـ	ـكـ	ـك	ك
Lam	لـ	ـلـ	ـل	ل
Mim	مـ	ـمـ	ـم	م
Ha	هـ	ـهـ	ـه	ه

Caractère	Initiale	Milieu	Finale	Isolé
Alif + ~			ـآ	آ
Alif + ء			ـإ	إ
			ـأ	أ
Waw + ء			ـؤ	ؤ
Ya+ ء		ـئـ	ـئ	ئ

Caractère	Initiale	Milieu	Finale	Isolé
Ta marbouta				ة
Lamalif				لا

Caractère	Initiale	Milieu	Finale	Isolé
Lamalif			ـلا	لا
Lamalif +~			ـلآ	لآ
Lamalif +ء			ـلأ	لأ
			ـلإ	لإ

TAB. 1.2 – (a) *les caractères arabes et leurs différentes formes, (b) les caractères additionnels « ة» 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 1.4) :

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

FIG. 1.4 – *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 ASMO attribue un seul code pour les différentes formes d'un même caractère, contrairement au latin où le code ASCII prévoit deux codes différents pour la même lettre dans sa forme majuscule et minuscule.

initiale	médiane	finale	isolée
علم	معلم	سمع	ورع
همس	مهد	لعبه	منتزه

TAB. 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 ;

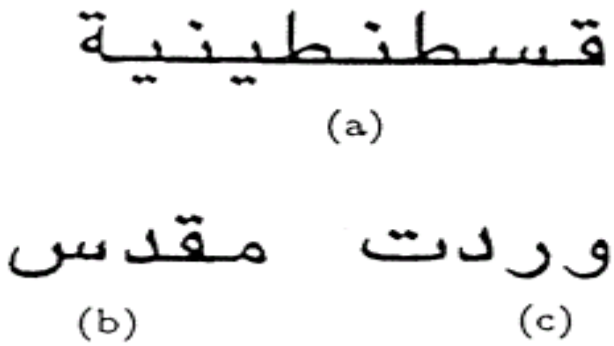


FIG. 1.5 – *Des mots arabes avec leurs composantes connexes*

- 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 peuvent être comprises comme verbe, nom ou adjectif. A titre d'exemple, «عـــــلم»

peut signifie « drapeau :عَـلَمٌ » ou « savoir :عِلْمٌ » 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 1.6)

ﺔ ﻓ ﻗ ﻘ ﻫ ﻩ ﻫ
ﺓ ﻡ ﻇ ﻣ ﻭ ﺹ ﻋ

FIG. 1.6 – *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 [ESS 99]. 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 (1.7) suivante montre un exemple de différents styles graphiques de l'écriture arabe

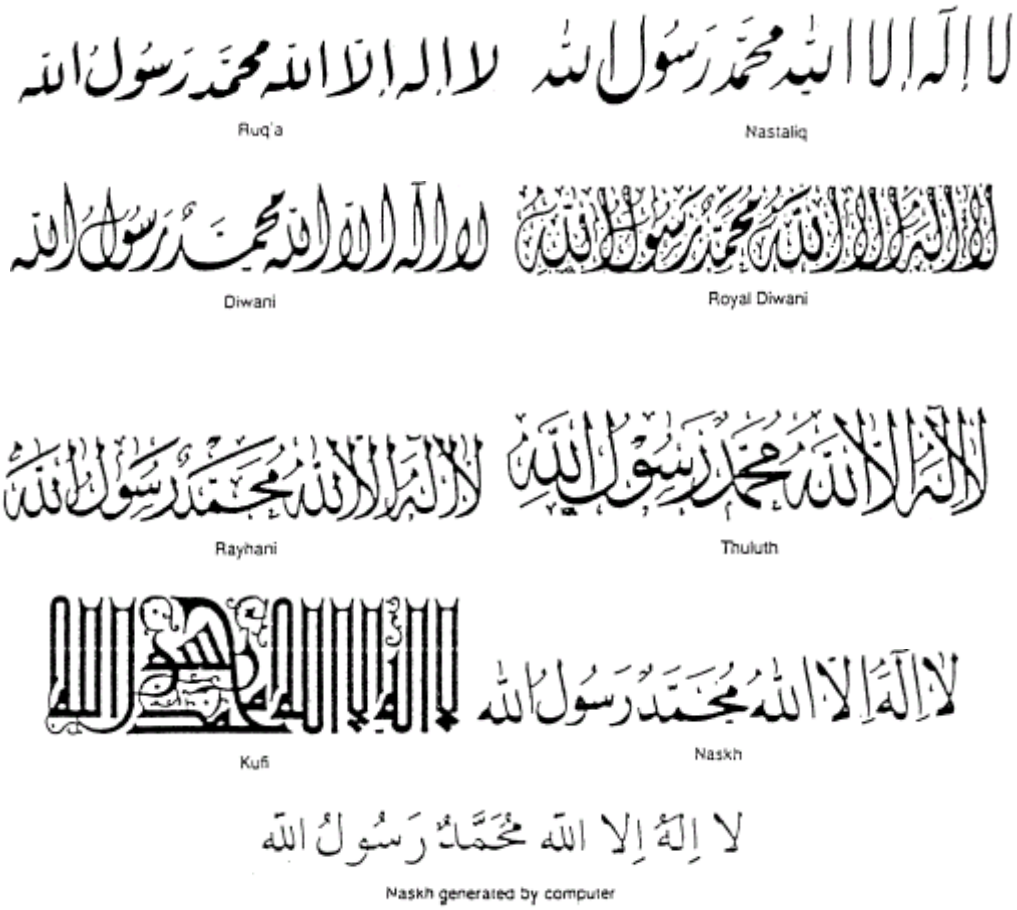


FIG. 1.7 – *Différents styles et fontes pour l'écriture arabe*

4. Approches de reconnaissance

Il existe deux approches pour la reconnaissance des mots : la reconnaissance globale et la reconnaissance analytique

1. ***Approche globale*** : dite encore *holistique*, a une vision générale du mot ; elle se base sur une description générale de l'image du mot. Elle considère le mot comme une seule entité indivisible et le décrit indépendamment des caractères qui le constituent [BEL 01]. Cette approche présente l'avantage de garder le caractère dans son contexte avoisinant, ce qui permet une modélisation plus efficace des variations de l'écriture et des dégradations qui peuvent l'entacher. Cependant, cette méthode est pénalisée par la taille mémoire, le temps de calcul et la complexité du traitement, qui croient linéairement avec la taille du lexique considéré, d'où une

limitation du vocabulaire. Cette approche est souvent appliquée pour réduire la liste de mots candidats dans le contexte d'une reconnaissance à grand vocabulaire. Il est nécessaire, dans ce cas, d'utiliser des primitives très robustes, comme dans les travaux de Govindaraju [GOV 94], pour ne pas manquer le mot réel parmi les candidats. Le mot est ensuite trouvé à l'aide de primitives de plus en plus précises (ou d'un classifieur de plus en plus fin).

2. ***Approche analytique*** : permet de s'affranchir des limites de l'approche globale, mais nécessite une interprétation locale basée sur une segmentation du mot. La reconnaissance consiste à identifier les entités segmentées puis tendre vers une reconnaissance du mot, ce qui constitue une tâche assez délicate pouvant générer différents types d'erreurs [ESS 99]. La difficulté d'une telle approche a été clairement évoquée par Sayre en 1973 et peut être résumée par le dilemme suivant *« pour reconnaître les lettres, il faut segmenter le tracé et pour segmenter le tracé, il faut reconnaître les lettres »* [LOR 92]

5. Processus de reconnaissance

Du signal écriture sous ses différentes formes, à la prise de décision par un système, il existe un certain nombre d'étapes à mettre en œuvre. La figure 1.8 représente globalement le processus général de reconnaissance de l'écriture. Dans un premier temps une phase de prétraitement est réalisée sur l'image acquise. Elle permet de réduire au maximum la variabilité intrinsèque à l'écriture ainsi que les bruits possiblement introduits lors de l'acquisition. Une seconde étape, optionnelle, est celle de la segmentation. L'écriture étant une concaténation de caractères, il est normal lors de la reconnaissance d'essayer de segmenter l'écriture à reconnaître en caractères. La troisième étape à être réalisée directement sur les données présentées en entrée du système est l'extraction de caractéristiques. Son but est la réduction de la quantité d'information et l'extraction des caractéristiques les plus pertinentes pour la reconnaissance. La prochaine étape est la classification, c'est le stade de décision dans un système de reconnaissance de texte. A cette étape les primitives extraites dans l'étape précédente sont utilisées pour identifier le segment de texte selon des règles établies préalablement. Généralement, à ce niveau, on utilise des modèles obtenus dans une phase d'apprentissage pour classifier les données de

test. La dernière étape dans un système de reconnaissance de texte est le post-traitement. Grâce à l'utilisation des informations d'ordre supérieur, cette étape peut améliorer le taux de reconnaissance en raffinant les décisions prises par l'étape précédente.

Notons que, les étapes (prétraitement, segmentation, post-traitement) ne sont pas nécessairement exécutées par tous les systèmes OCR.

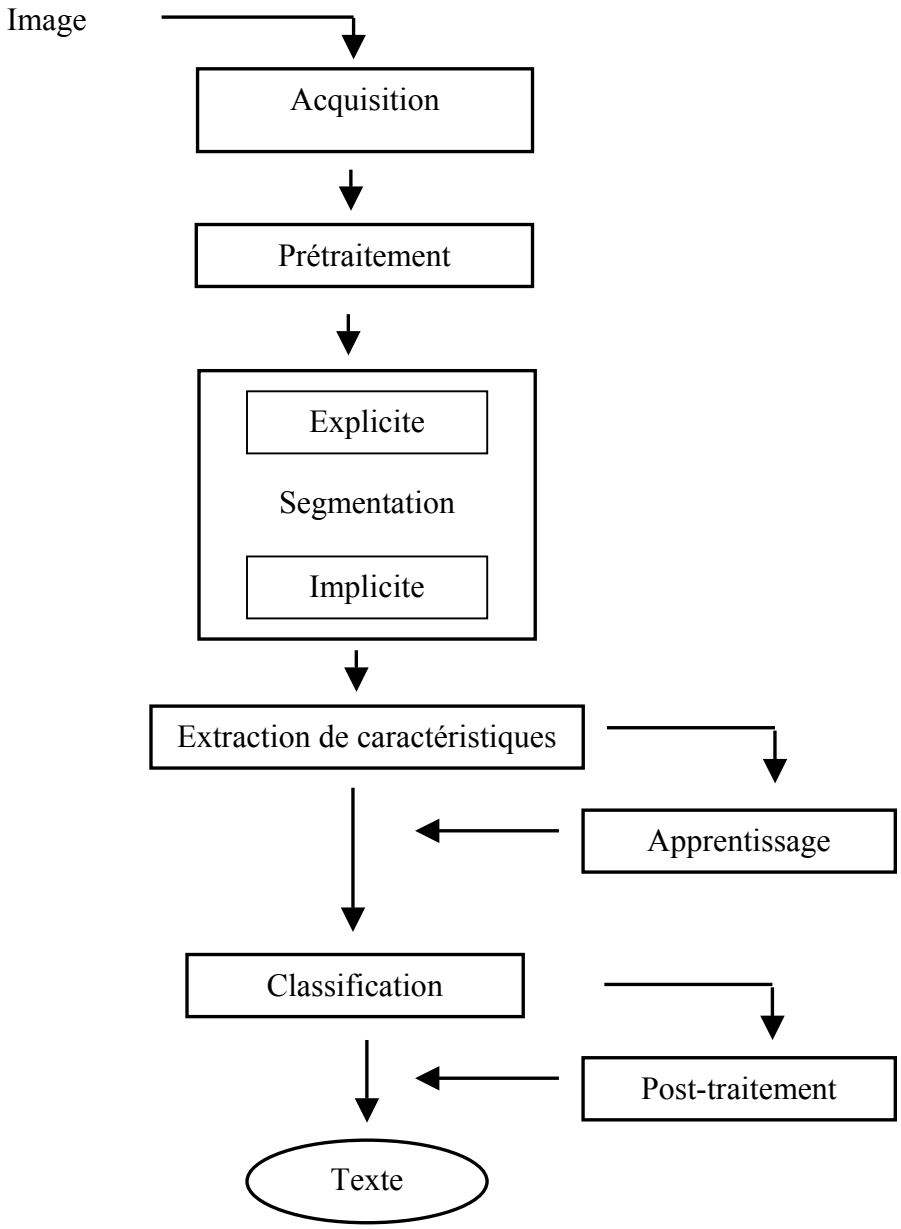


FIG. 1.8 – *Un modèle général pour les systèmes OCR*

5.1. L'acquisition

La première étape dans aun système OCR consiste à convertir l'écriture en grandeurs numériques adaptées au système de traitement avec un minimum de dégradations possibles. En mode hors ligne, selon l'outil d'acquisition utilisé (scanner ou caméra), une image en couleur ou en niveau de gris est obtenue.

5.2. Le prétraitement

Lorsque l'acquisition est réalisée, la plupart des systèmes comportent une étape de prétraitement. Généralement, ces prétraitements ne sont pas spécifiques à la reconnaissance de texte, mais sont des prétraitements classiques en traitement d'image. Le prétraitement a pour but de préparer l'image du tracé à la phase suivante d'analyse. Il s'agit essentiellement de réduire le bruit superposé aux données et ne garder, autant que possible, que l'information significative de la forme présentée. Le bruit peut être dû au dispositif d'acquisition, aux conditions d'acquisition (éclairage, mise incorrecte du document…), ou encore à la qualité du document d'origine

Parmi les opérations de prétraitements généralement utilisées, citons : la binarisation, le redressement de l'écriture, le lissage, la squelettisation et la normalisation.

5.2.1. La binarisation

Elle permet de passer d'une image de niveaux de gris à une image binaire composée de deux valeurs 0 et 1, plus simple à traiter. En général, on utilise un seuil de binarisation approprié qui traduit la limite des contrastes fort et faible dans l'image. Mais pour des images peu contrastées ou à contraste variable (càd la distribution de niveaux de gris n'est pas clairement bimodale), il est difficile de fixer ce seuil à une valeur précise. Une méthode classique pour déterminer un seuil de binarisation consiste à calculer l'histogramme des niveaux de gris de l'image. Notons que, la valeur du seuil sera égale à la valeur du niveau de gris se trouvant dans la vallée entre les deux pics de l'histogramme. Les pixels ayant un niveau de gris supérieur à ce seuil appartiennent au fond et ceux ayant une valeur inférieure appartiennent à l'objet [ZAH 90].

Pour des images de niveaux de gris, on peut trouver dans [TRI 95] une bonne synthèse des méthodes de binarisation, proposant des seuils adaptatifs (càd s'adaptant à la différence de distribution des niveaux de gris). Mais le défi reste pour les fonds texturés où il est difficile de trouver une modalité claire dans la distribution. Liu et Srihari [LIU 97] proposent une solution pour les images d'adresses postales. La recherche du seuil passe par plusieurs étapes : binarisation préliminaire basée sur une distribution de mixture multimodale, analyse de la texture à l'aide des histogrammes de longueurs des traits, et sélection du seuil à partir d'une arbre de décision.

Toutefois, il est à remarquer que certains auteurs ne passent pas nécessairement par cette étape intermédiaire et extraient directement les primitives utiles à la reconnaissance à partir de l'image en niveau de gris [PET 90], [EIK 96].

5.2.2. Le lissage

L'image des caractères peut être entachée de bruit introduit durant l'acquisition et au cours des différentes transformations. Ce bruit correspond soit à des absence de points (trous), soit à des empâtements et donc à une surcharge de points. Le lissage consiste à examiner le voisinage d'un pixel et éliminer les pixels isolés d'une part *(nettoyage)*, et à boucher les trous vide d'autres part *(bouchage)* [BEL 92]

Une méthode fréquemment utilisée pour le lissage, consiste à parcourir l'image pixel par pixel, en utilisant une fenêtre de taille 3×3 et à changer la valeur du pixel en se basant sur les valeurs de ses 8 voisins. Un point 1 est mis à 0 s'il n'y a pas assez de points noirs autour de lui et vice versa [MAH 96] [AMI 96]. Une méthode statistique similaire consiste à inverser le pixel si la somme des pixels voisins est inférieure à un seuil déterminé [MAH 94]. Cependant, le choix du seuil est critique : un faible seuil peut éliminer aussi bien les imperfections que les discontinuités naturelles et discriminantes dans une lettre ; par exemple ; un « ـا » peut facilement se transformer en un « ـ » à la suite d'un mauvais choix du seuil.

5.2.3. Redressement de l'écriture

C'est une opération fréquente en reconnaissance de l'écriture, souvent due soit à un mauvais positionnement du document sur le scanner, soit à une mise en page irrégulière de l'auteur, conduisant à une inclinaison de l'image (voir la figure 9). Parmi les techniques de détection de l'angle d'inclinaison les plus utilisées, nous pouvons citer la méthode de Trincklin, les méthodes de projections, la transformée de Hough [AMI b96, BER 98], la méthode des k plus proches voisins [KWA 02, SEH 00].

La méthode de Trincklin [BEL 92] utilise la méthode des moindres carrés pour évaluer l'inclinaison du texte. Cette méthode est rapide, peu sensible au bruit et appropriée pour de nombreux types de documents incluant des graphiques mais elle nécessite d'avoir des lignes relativement bien justifiées sur la gauche, elle n'est pas appropriée pour les documents multi-colonnes et l'angle d'inclinaison est compris entre -10^0 et $+10^0$.

La méthode de projections [BAG 97] est basée sur la calcul de l'histogramme horizontal de l'image du document pour chaque angle appartenant à l'intervalle de détection qui varie entre -10^0 et $+10^0$. Cette méthode est facile à implémenter, appropriée pour des documents à structure simple, mais elle n'est pas appropriée pour des documents complexes contenant des graphiques.

Sehad et al.[SEH 05] proposent une méthode, pour la détection de l'angle d'inclinaison de documents imprimés, basée sur le régression linéaire et la transformée en ondelettes.

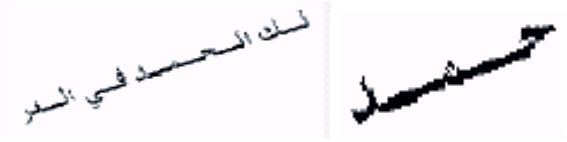


FIG. 1.9 – *Texte incliné [SHE 05]*

5.2.4. La normalisation

Permet de ramener les images des mots à des tailles standard. La normalisation introduit généralement de légères déformations sur les images mais elle reste indispensable pour certains types de systèmes et de méthodes de reconnaissance qui sont sensibles aux petites

variations dans la taille et la position comme c'est le cas dans les templates matching et les méthodes de corrélation.
Plusieurs méthodes de normalisation ont été rencontrées dans la littérature. La plupart ont été utilisées pour la normalisation de l'écriture chinoise manuscrite qui renferme une large variété de symboles [WAK 97, LEE 93]. Elles sont aussi utilisées pour la normalisation des chiffres manuscrits. Dans certains travaux sur l'AOTR, cette opération se fait par le calcul du rectangle englobant chaque caractère (càd Bounding Box : le plus petit rectangle contenant le caractère).

5.2.5. Squelettisation (Thinning)

Comme l'image binaire se présente comme une succession de traits d'épaisseurs variables, mal définis, et généralement avec bruit, l'opération de squelettisation est appliquée pour simplifier l'image du caractère en une image à « ligne » plus facile à traiter en la réduisant au tracé du caractère. Les points du tracé résultant constituent généralement les lignes centrales des points initiaux. Elle est définie comme étant l'équivalent de la forme, mais avec une épaisseur très réduite qui peut varier entre 1 et 2 pixels. La squelettisation (amincissement), est une opération essentielle dans plusieurs systèmes AOTR [AMI 92], [AMI a96] [BEN 98]. Généralement, les systèmes off-line de reconnaissance d'écriture manuscrite, utilisent la squelettisation pour atténuer la variabilité inhérente aux styles d'écriture. Cependant, la squelettisation des caractères arabes peut induire en erreur : deux points diacritiques sont souvent confondus avec un seul point [ESS 99]

Selon la qualité du document à traiter, le type d'écriture (manuscrite ou imprimée) et la méthode d'analyse adoptée, une ou plusieurs techniques de prétraitements sont appliquées. Pour des méthodes d'analyse de "matching" par exemple, une normalisation des tailles des caractères est nécessaire. La squelettisation est effectuée si on a choisi d'extraire les caractéristiques (généralement de type structurel) sur le squelette, dans ce cas, une extraction des composantes connexes peut être omise. Des méthodes plus ou moins complexes de lissage sont opérées en fonction de la dégradation du document considéré.

5.3. Segmentation du mot

L'analyse du mot passe souvent par sa décomposition en caractères, souvent plus facile à traiter. Cependant, cette séparation n'est pas toujours possible. Par ailleurs les méthodes analytiques par opposition aux méthodes globales, présentent l'avantage de pouvoir se généraliser à la reconnaissance d'un vocabulaire sans limite à priori, car le nombre de caractères est naturellement fini. De plus, l'extraction des primitives est plus aisée sur un caractère que sur une chaîne, le nombre de modèles à considérer est fini, la complexité des calculs est généralement réduite. Généralement, la performance de la segmentation affecte directement la fiabilité du système global : outre des erreurs de confusion, certaines erreurs de segmentation peuvent engendrer même des rejets.
Deux techniques ont été utilisées dans la reconnaissance des mots :

1. ***Segmentation implicite*** : consiste à segmenter le mot en parties inférieures aux lettres appelées *graphèmes* et à retrouver les lettres puis les mots par combinaison de ces graphèmes
2. ***Segmentation explicite*** *:* consiste à segmenter le mot exactement en caractères en utilisant des propriétés générales de l'écriture d'un pseudo mot. Dans cette technique, la segmentation devient l'étape la plus critique dans le processus de reconnaissance, car une erreur à ce niveau conduira automatiquement à une classification erronée. Ce type de segmentation est fréquemment illustré par des règles identifiant les points de segmentation des caractères.

Le problème de segmentation des mots latins qu'ils soient imprimés ou manuscrits a été largement étudié [LU 96]. Malgré qu'il soit possible parfois, d'appliquer les résultats de ces études pour l'arabe, en général ils ne sont pas suffisants pour segmenter les mots arabes. Un caractère arabe peut être constitué de plus d'une partie et la nature cursive de l'écriture arabe rend l'utilisation de ces méthodes insuffisante [SAR 99].
Plusieurs méthodologies de segmentation des mots arabes, explicite ou implicite, ont été développées utilisant des techniques variées (histogrammes de projection, contour, squelette, fenêtre glissante…). Cependant la segmentation reste un problème ouvert. Une liste détaillée d'algorithmes de segmentation se trouve dans [ELB 95]

La segmentation effectuée dans [AMI 98] est basée sur la propriété que tous les caractères arabes ont une longueur plus grande que leurs largeurs. Cette propriété est essentielle pour la segmentation des mots en détectant la ligne de base. Amin et al [AMI 86] segmentent les pseudo-mots imprimés en caractères en utilisant la projection verticale.
Si un mot à traiter est projeté verticalement :

$$v(j) = \sum_{i} w(i, j) \tag{1.1}$$

où *w(i,j)* est zéro ou un, et *i, j* représentent respectivement la ligne et la colonne de l'histogramme de projection. Un point de connexion entre deux caractères a une somme inférieure à la valeur moyenne *Av* (l'équation 1.2) :

$$Av = (1/\ \ Nc)\sum_{j=1}^{Nc} Xj \tag{1.2}$$

Nc est le nombre de colonnes, *Xj* est le nombre de points noirs dans la colonne *j*.
Par conséquent, chaque partie ayant une valeur inférieure à la valeur moyenne *Av* est un point de connexion entre deux caractères.

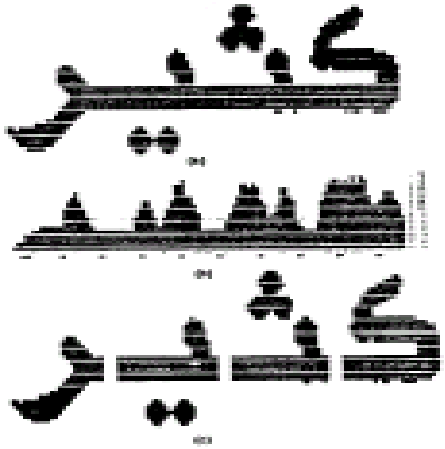


FIG. 1.10 – *Un exemple d'un mot arabe segmenté en caractères [AMI 97]*

La détermination de points d'interconnexions représentant les positions où la valeur de l'histogramme de projection tombe au-dessous d'un certain seuil, conduit des fois à segmenter un caractère en plusieurs parties.

Pour remédier à ce problème, certains chercheurs utilisent des règles heuristiques qui prévoient le bon découpage des caractères ayant une largeur inférieure à un certain seuil.

Par exemple, Amin et al [AMI 89, 91] affirment, en examinant les caractères arabes, que la distance entre deux pics successifs de l'histogramme de projection n'excède pas un tiers de la largeur du caractère :

$$|d_k| < d_l/3 \tag{1.3}$$

où d_k est la distance entre le $k^{ème}$ et $(k+1)^{ème}$ pics de l'histogramme et d_l la largeur totale du caractère. Si l'inégalité (3) n'est pas vérifiée par l'histogramme de projection verticale, les caractères restent non segmentés. En plus, à la fin du mot la règle suivante est vérifiée :

$$L_{k+1} > 1{,}5L_k \tag{1.4}$$

où L_k est le k ème pic dans l'histogramme, cette règle est appliquée à la fin du mot ou pseudo-mot, à cause de l'inter connexité des caractères arabes et de leurs formes à la fin du mot.

Les méthodes de segmentation en utilisant l'histogramme de projection verticale sont adaptées aux caractères imprimés ayant des largeurs stables (ex. fonte unique). Ces méthodes sont inefficaces dans la segmentation des textes manuscrits ou composés, qui ont des ligatures. Ceci est dû au fait que les points de connexions ne sont pas situés le long de la ligne de base, ils ne pouvent pas ainsi être détectés sur la projection verticale. Avec une telle méthode, il faut faire attention lorsqu'il s'agit de reconnaître les textes italiques ou soulignés [BEN 00]. Dans plusieurs systèmes qui segmentent en caractères, la segmentation est la cause majeure des erreurs. Par exemple, le système IRACI [AMI 80], qui reconnaît les caractères manuscrits isolés, a une performance de 95,4% comme taux de reconnaissance, beaucoup plus grande que le système IRACII (taux = 80%) qui procèdent à la segmentation du mot puis à sa reconnaissance.

Almuallim et Yamaguchi ont proposé une méthode structurelle basée sur quatre phases pour la reconnaissance des mots arabes manuscrits. La première phase est le prétraitement, où le mot aminci et la ligne de base sont détectés. Comme il est difficile de segmenter un mot cursif en caractère, les mots sont segmentés en « traits séparés », qui seront classifiés comme caractères complémentaires, traits avec boucles et traits sans boucles. Par la suite, ces traits sont classifiés en utilisant leurs propriétés géométriques et topologiques. A la dernière phase, les positions relatives des traits sont examinées pour être combinées par la suite dans plusieurs étapes pour former une chaîne de caractères qui représente le mot reconnu [ALM 87].

Dans les travaux développés par Abdelazim et al.[ABD 90], la segmentation consiste à extraire des graphèmes qui peuvent correspondre à une partie du caractère, au caractère, à un groupe de caractères ligaturés verticalement , à un point diacritique supérieur ou encore à un point diacritique inférieur. La segmentation en composantes connexes dans [ABD 90] se base sur la détection du contour. Une méthode inspirée d'une technique de détermination des zones de silence dans un signal de parole est appliquée à chaque composante connexe.

Olivier et al [OLI 96] ont proposé un algorithme de segmentation des mots arabes en graphèmes. Cet algorithme commence par extraire le contour supérieur du mot en utilisant le code de Freeman. Leur souci majeur était d'éviter les recouvrements fréquents dans l'écriture arabe qui pourrait survenir dans l'écriture manuscrite cursive. La segmentation consiste à extraire les minima locaux du contour supérieur ; ces minima sont considérés comme points de segmentation primaire (PSP). Ensuite, ces points sont retenus comme points de segmentation décisifs (PSD) s'ils respectent certaines règles observées sur l'image originale du mot.

5.4. Extraction de caractéristiques

Pour la prise de décision, un système de reconnaissance n'a besoin que de l'information pertinente pour différencier un objet d'un autre. Dans ce but, une étape d'extraction de caractéristiques est réalisée. C'est une phase critique lors de la construction d'un système de reconnaissance. L'une des raisons pour lesquelles cette étape pose un problème est qu'une grande majorité des techniques d'extraction s'accompagne d'une perte d'information irrémédiable.

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

L'extraction de caractéristiques en reconnaissance de l'écriture est confrontée au grand problème de la variabilité intra-classe. En effet, d'un point de vue visuel, un caractère peut prendre différentes formes, en fonction de sa position dans le mot. Cependant, les plus grandes variations sont introduites par le scripteur. L'écriture étant propre à chaque individu, le tracé résultant de l'écriture d'un même mot par deux personnes peut être bien différent. De plus, pour un même scripteur, un certain nombre de contraintes influencent la réalisation du tracé de son écriture. Nous pouvons citer entre autres l'outil, le support et même l'humeur de l'individu.

En plus de la taxonomie présentée dans la figure 1.3 (page 13), le domaine de la reconnaissance de caractères peut être décrit par la méthode de collection de données, des méthodes d'extraction de caractéristiques, des méthodes de classification ou du format de représentation des données. Trier et al [TRI 96] ont réalisé une taxonomie des méthodes d'extraction selon le format de représentation de l'image que nous présentons dans le tableau 1.4.

Caractérisques	Image en gris	Image binaire	Contour	Squelette
Appariement	X	X		X
Motifs déformables	X			X
Transformation unitaire	X	X		
Transformation log-polaire	X	X		
Moments géométriques	X	X		
Moments de Zernik	X	X		
Ondelettes	X	X		
Algébriques	X	X		
Histogrammes de projection		X		
Masques	X	X		
Profil de contour			X	
Code de Freeman			X	
Spline			X	
Descripteurs de Fourrier			X	X
Description graphique				X
Zonage	X	X	X	X

TAB. 1.4 – *Taxonomies des méthodes d'extraction de caractéristiques selon la représentation de l'image [TRI 96]*

Le tableau montre quatre formats de représentation d'image arrangés en colonnes qui sont : image en gris, image binaire, le contour et le squelette. Pour chaque format, des méthodes d'extractions de caractéristiques raisonnables sont indiquées. Notons que le choix des caractéristiques et du format de représentation ne sont pas totalement dépendants ni tout à fait indépendants.

Selon Amin [AMI 98] les caractéristiques peuvent être classées en deux catégories :

- ***les caractéristiques locales*** : qui sont souvent géométriques (par exemple : concavité/convexité, les fins de traits, les jonctions (en T ou Y), ainsi que les intersections en (X)..
- ***les caractéristiques globales*** : qui sont souvent topologiques (connectivité, nombre de composantes connexes, …etc.) ou statistiques (transformé de Fourier, moments invariants…etc.).

Dans la suite de cette section, nous présentons quelques caractéristiques, souvent utilisées en reconnaissance de l'écriture.

5.4.1. Transformations et développements en séries

Dans le domaine de la reconnaissance des formes, il est intéressant d'essayer d'extraire, à partir des images, des informations "non visibles". De telles techniques sont regroupées dans cette catégorie. Elles utilisent une transformation globale de manière à changer d'espace de représentation et ainsi faciliter l'extraction de caractéristiques pertinentes.

La transformée de Fourier est l'une des méthodes les plus utilisées en reconnaissance de formes et de caractères [FIL 98][MAH 94]. Les caractéristiques extraites sont en fait les descripteurs de Fourier basés sur les coefficients complexes des séries de Fourier. Elles sont invariantes aux rotations et aux changements d'échelle [MAH 94]. La propriété d'invariance aux rotations implique des problèmes de reconnaissance de certains caractères comme "6" et "9". Il faut donc rajouter d'autres caractéristiques au vecteur de manière à régler ce problème.

Une autre transformation globale, assez proche de celle de Fourier est celle des ondelettes [MAL 97],[SHE 99]. Le principal intérêt des ondelettes est que ces dernières permettent d'obtenir une information fréquentielle localisée concernant un signal ou une fonction de base choisie. Malgré certains avantages, cette technique est peu utilisée en reconnaissance de formes. La raison en est que les caractéristiques extraites ne sont pas invariantes à la translation.

Aussi la transformée de Hough est utilisé dans [ALB a95] pour représenter le squelette d'un caractère comme un ensemble de segments de ligne puis utilisent la longueur, la position, et la pente des segments de ligne au squelette comme des caractéristiques.

Une autre grande famille appartenant à cette catégorie est celle des moments invariants. L'invariance recherchée est liée à la rotation, à la translation et au changement d'échelle. Les caractéristiques extraites par ces techniques sont considérées comme le résultat d'une transformation globale appliquée uniquement aux pixels de la forme analysée. Il existe plusieurs formulations des moments invariants, comme celle de HU [HU 62, cité dans TRI 96] et celle de li [LI 92]. Cependant, les plus utilisées actuellement sont celles dérivées des polynômes de Zernik [GRA 03]. La raison est que ces derniers ont des performances supérieures en termes d'invariance. Les moments de HU ont été utilisé pour la reconnaissance de caractères arabes manuscrits dans [ELD 90] et [ELK 90]. Azizi dans [AZI a02]

a utilisé les sept moments de HU pour la reconnaissance des mots arabes manuscrits. ces moments sont des mesures statistiques de la distribution des points dans l'image et ils peuvent être exprimés directement à partir des moments centraux normalisés d'ordre 2 et 3.

5.4.2. Les caractéristiques structurelles

Les caractéristiques extraites de cette famille permettent de mettre en évidence les propriétés topologiques et géométriques de la forme. De part leur nature, elles peuvent être utilisées pour caractériser une forme tant d'un point de vue local que global. Concernant la reconnaissance de caractères, une propriété intéressante de ces caractéristiques est leur faible sensibilité aux distorsions et donc aux variations de styles [GRA 03]. Une autre qualité, appréciée pour des applications industrielles, est que leur extraction n'est pas coûteuse en termes de temps de calcul, en particulier par rapport à celles de la catégorie précédente. Cependant, l'extraction de ce type de caractéristiques n'est pas toujours triviale.

Les caractéristiques structurelles utilisées dépendent de la forme à classifier. Dans la littérature les caractéristiques employées pour la reconnaissance de l'écriture arabe comprennent :

- le nombre de traits (strokes), leurs tailles, directions et pentes [AMI 80, 82] ;
- les points extrémaux (end points) [AMI 80, AMI 82, ZAH 90] ;
- la hauteur et la largeur du caractère [AMI 80, AMI 82, ZAH 90] ;
- la catégorie de la forme (partie primaire, point diacritique, etc) [AMI 96, ZAH 90] ;
- le nombre de points diacritiques et leurs positions par rapport à la ligne de base, et les zigzags (Hamza's) [MAH 94, ALY 92, SOU 06];
- les points de rebroussement et d'inflexion [ZAH 90, BEN 98] ;
- le nombre d'occlusions (boucles) [AMI 96, SOU 98, SOU 06] ;
- les concavités et les convexités dans les quatre principales directions (ouest, nord, est, sud) [MAH 94] ;
- les jambages (descendants) et les hampes (ascendants) [SOU 98, ESS 97,SOU 06] ;
- la taille du rectangle englobant le tracé ou le caractère [AMI 80, AMI 82, ZAH 90] ;
- le nombre de composantes connexes [SOU 98, BEN 98] ;

- les courbures avec leurs directions (ouest, est, nord, sud) [AMI 85, BEN1 98, AMI a96] ;
- les segments droites et obliques avec leurs directions [AMI a96, BEN 98].

5.4.3. Allongements horizontaux et verticaux

Les directions des allongements de pixels d'un caractère permettent d'en exprimer la structure de son tracé. La technique d'obtention de ces caractéristiques consiste à effectuer une projection des pixels du caractère sur un axe perpendiculaire à la direction de recherche des allongements. La détection des maxima locaux sur l'histogramme résultant permet d'obtenir la position et la valeur des allongements. Amin et al [AMI 98] utilisent une transformation basée sur les projections horizontales et verticales de la forme. Durant le balayage de chaque ligne de l'image du caractère, tous les points connectés sont désignés selon leur ordre d'apparition. Al Yousfi et al. [ALY 92] ont utilisé une méthode basée sur la projection verticale pour segmenter les mots arabes.
Un problème associé à cette technique est qu'elle induit la détection d'un grand nombre de maxima, ne correspondant pas forcément à des allongements. Chim et al utilisent ces caractéristiques, complétées par une détection de segments de ligne, dans son système de reconnaissance de chiffres [CHI 98].

5.4.4. Intersections avec des droites

Les propriétés projectives d'un caractère peuvent être mises en évidence en effectuant le comptage d'intersections entre une ou plusieurs droites et le caractère. Dans cette catégorie, nous pouvons classer une méthode basée sur les intersections. Il s'agit des lieux caractéristiques ou " characteristic Loci" [KNO 69, cité dans TRI 96] qui consiste à étiqueter chaque pixel du fond de l'image, en fonction du nombre d'intersections entre les segments de droites horizontales issues de ce point et le caractère. Cette méthode tolère des distorsions et des variations légères, et le calcul y est facile.

5.4.5. Superposition de représentations ou de prototypes

Template Matching, Pattern matching, Model matching

Dans ces méthodes de *Template matching*, les images et les prototypes (template) doivent avoir la même taille M qui désigne le nombre de pixels de l'image.Elles se basent sur la comparaison de la forme pixel par pixel à un ensemble de patrons (template) de formes. Dans ce cas, l'image entière du caractère ou du mot est utilisée comme un vecteur caractéristique [BEL 92]. Au niveau de la reconnaissance, une mesure de similarité (ou dissimilarité) entre chaque patron T_j et l'image de la forme est calculée. Le patron T_k, ayant la mesure la plus élevée de similarité (ou la petite dissimilarité), est identifié à condition que cette similarité soit supérieure à un certain seuil. Cette méthode est rapide du point de vue calcul, mais elle est très sensible aux distorsions. Pour remédier à cela, les chercheurs utilisent différentes variantes de cette méthode dont l'une des plus fondamentales consiste à choisir plusieurs pixels dans des positions clés pour la corrélation.

5.4.6. Description en graphes

Cette technique représente le squelette du caractère sous forme de graphe, par l'extraction de traits approximatifs du squelette. Un caractère devient un ensemble d'arcs reliés entre eux. Chacun d'eux étant décrit par sa longueur et son orientation [TRI 96].

5.5. Apprentissage

La décision nécessite de définir clairement la connaissance que nous avons sur les formes à traiter. Cette définition repose sur l'apprentissage qui se charge d'acquérir la connaissance et de l'organiser en classes ou modèles de références [BEL 92]. L'idéal serait d'apprendre au système autant d'échantillons que de formes d'écritures différentes, mais cela est impossible à cause de la grande variabilité de l'écriture qui conduirait à un grand nombre de modèles. La tendance consiste à remplacer ce nombre par une meilleure qualité des traits caractéristiques. Les procédés d'apprentissage sont différents selon qu'il s'agit de reconnaître des caractères imprimés ou manuscrits, ou de reconnaître un texte monofonte ou multifonte. D'une manière générale, nous distinguons deux types de techniques d'apprentissage : *supervisé* et *non supervisé*

5.5.1. Apprentissage supervisé

Les différentes familles des formes sont connues à priori et la tâche d'apprentissage est guidée par un superviseur ou professeur. L'apprentissage est réalisé lors d'une étape préliminaire à la reconnaissance en introduisant un grand nombre d'échantillons de référence. Le concepteur indique, pour chaque caractère en entrée, sa classe d'appartenance appropriée. Le choix des caractères de références est fait à la main en fonction de l'application, les échantillons choisis sont les plus représentatifs possibles de la typographie des caractères de l'application.

5.5.2. Apprentissage non supervisé

Appelé encore, suivant l'approche utilisée, classification automatique. Il consiste à doter le système d'un mécanisme automatique qui s'appuie sur des règles précises de regroupement pour trouver les classes de références. Dans ce cas, les échantillons sont introduits en grand nombre par l'utilisateur sans indicateur sur leur place. Ce type d'apprentissage est intéressant car il permet de renseigner le concepteur sur les ambiguïtés entre les caractères afin d'agir en conséquence (par exemple, en ajoutant des échantillons pour renforcer la représentativité d'une classe), mais il n'assure pas toujours une classification correspondante à la réalité (celle du concepteur).

5.6. La décision

Le processus de reconnaissance peut toujours se résumer à une décision de la classification. Pour cela, il faut choisir une représentation qui permette une description de l'objet à analyser (caractère traité), puis une règle de décision qui s'appuie sur cette description. A partir de la description du caractère traité, le module de reconnaissance cherche, parmi les modèles de références en présence, ceux qui lui sont les plus "proches". La notion de proximité a un sens différent en fonction de la nature de la représentation et du type de la méthode utilisée.

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 des modèles ne correspond à sa description. Dans les deux premier cas, le décision peut être accompagnée d'une mesure de vraisemblance, appelée aussi *score* ou taux de reconnaissance [BEL 92].
Différentes approches de décision ont été proposées dans la littérature, mais, de façon classique, on peut distinguer cinq catégories :

- Approche structurelle ;
- Approche statistique ;
- Approche stochastique ;
- Approche connexionniste ;
- Approches hybrides.

Une différence essentielle entre ces approches réside dans la représentation de la forme, par exemple le vecteur de caractéristiques dans l'approche statistique et agencement de primitives pour l'approche structurelle.
Nous reviendrons sur ces approches dans la première partie du deuxième chapitre (page 44).

5.7. Le post-traitement

L'objectif du post-traitement est d'améliorer le taux de la reconnaissance. Il est responsable de sélectionner une solution parmi un ensemble en ayant recours à des informations de haut niveau (lexicales, syntaxiques, sémantiques, pragmatiques,…) qui ne sont pas disponibles au niveau du classifieur. Cette étape du processus est souvent implémentée comme un ensemble de techniques qui dépendent de plusieurs facteurs entre autres : les fréquences d'apparition des mots, les lexiques, et autres informations sur le contexte.
Des vérifications contextuelles classiques, telles que la recherche dans un dictionnaire, les probabilités d'occurrences de bigramme et de trigramme…, sont appliqués dans les différents travaux qui prévoient un post-traitement.
La méthode basée sur un dictionnaire est traditionnellement simplifiée pour accélérer la recherche et réduire la complexité de calcul. Le dictionnaire est construit à partir des mots réduits à leurs racines, les suffixes et les préfixes sont ainsi éliminés. Cependant, des

modèles plus élaborés sont construits afin de spécifier la relation racine-suffixe-préfixe [ABD 92].

Par ailleurs, le post-traitement, malgré l'amélioration des scores de reconnaissance qu'il peut apporter, n'est pas très employé en AOCR ce qui peut s'expliquer par le manque de dictionnaires électroniques de validation et de statistiques relatives aux occurrences des n-grammes. Ces derniers sont relatifs à l'application considérée et aux lexiques.

6. Difficultés liées à l'OCR arabe

La reconnaissance de l'écriture arabe (AOCR : Arabic OCR) remonte aux années quatre vint [AMI 80], depuis, plusieurs solutions ont été proposées. Elles sont aussi variées que celles utilisées dans le latin.

Dès les premiers travaux de reconnaissance de l'écriture arabe (imprimé ou manuscrit), les deux modes de reconnaissance, statistique et dynamique, ont été considérés [AMI 80]. Cependant, les travaux en ligne restent relativement peu nombreux [ESS 99]. Les caractéristiques morphologiques de l'écriture arabe, compliquent la tâche de l'OCR à différents niveaux du traitement. Dans ce qui suit, nous présentons une synthèse des principales particularités de l'AOCR suivant les étapes chronologiques d'un système OCR général afin de mieux localiser les différents problèmes liés à la reconnaissance de cette écriture et de les localiser dans leur phase de traitement.

6.1. Prétraitement

Au stade de prétraitement, le problème classique est lié aux boucles qui risquent d'être bouchées ou ouvertes et aux points diacritiques qui peuvent être éliminés à la suite de certaines opérations de prétraitement ou encore confondus avec le bruit [ESS 02]. En effet, les prétraitements risquent d'altérer surtout la forme des points diacritiques de manière à les confondre avec le bruit s'ils sont trop amincis, ou à induire en erreur quant à la détection de leur nombre par la méthode des densités, si leur taille à considérablement augmenté par filtrage, par exemple. Les points risquent également d'être accolés au corps du caractère associé à cause d'une dégradation ou d'une normalisation de taille. Un autre problème typique rencontré à la suite d'une mauvaise squelettisation, particulièrement dans le cas du

manuscrit, provient de la confusion de deux points diacritiques avec un seul point, très souvent, dans les deux cas, nous obtenons un segment de droite [ZAH 90].
Pour ces différentes raisons, dans la plupart des travaux, les points sont éliminés au début du traitement et les étapes de ce dernier sont alors effectuées sur le corps du caractère. Par la suite, des traitements ultérieurs sont effectués sur les points diacritiques de manière individuelle [ESS 99].

6.2. Segmentation

La reconnaissance des différentes composantes connexes, nécessite d'abord leur extraction de la page, ceci suppose une "décomposition page" au préalable, qui consiste à retrouver la structure physique du document en délimitant les différentes entités structurelles homogènes : caractère, blocs de texte, blocs graphiques…etc. Dans le cas de l'arabe, la "décomposition page" est généralement limitée à la séparation des lignes de texte [ESS 99], à l'extraction des mots et à la délimitation des caractères.
Les méthodes de segmentation en ligne de texte se basent souvent sur la projection horizontale pour extraire les lignes. Cependant la présence des points diacritiques complique cette extraction et conduit parfois à la fusion des paragraphes. Dans certaines fontes, deux ou trois caractères peuvent se chevaucher verticalement. Très peu de travaux ont tenté à résoudre le problème de la ligature. El Badr et al. [ALB b95] considèrent les lieux des ligatures verticales parmi l'ensemble des formes préalablement apprises au système.
A l'issue de l'analyse de ces algorithmes de segmentation et les algorithmes présentés dans la section 5.3, nous retenons les points suivants :

- les liaisons entre les caractères sont variables, elles sont soit trop courtes soit, au contraire, relativement longues, ce qui occulte les marques de séparations ;
- l'existence de caractères ligaturés verticalement complique la tâche de segmentation, souvent, des caractères sont segmentés comme une seule entité (problème de sous segmentation) ;

- des points de segmentation indésirables peuvent apparaître dans le tracé de l'écriture. En effet, une dégradation du document peut introduire des irrégularités sur le tracé.

Malgré les diverses approches proposées, la segmentation en caractères ne peut se faire de façon correcte. La segmentation en graphèmes est relativement plus simple à réaliser que la segmentation en caractères. Par ailleurs, une coopération segmentation/reconnaissance (méthode dite récursive) permet de mieux gérer les erreurs de segmentation et améliore les scores de la reconnaissance. Une autre approche consisterait à faire coopérer deux ou plusieurs méthodologies de segmentation et considérer les points de segmentation communément votés.

6.3. Extraction de caractéristiques

La synthèse des travaux considérés montre que les différents types de primitives (structurelles, géométriques, statistiques, transformations globales, corrélations…) et les différentes méthodes de classification (statistiques, structurelles, syntaxiques,…) qui existent dans la littérature, ont été pratiquement toutes utilisées dans la description de l'écriture arabe. Toutefois, les caractéristiques les plus utilisées dans les systèmes de reconnaissance des mots arabes sont les caractéristiques perceptuelles [SOU 06],[FAR 05], [SNO 02]. Plus récemment, en reconnaissance de l'écriture cursive, il a été montré que la plupart de l'information discriminante est contenue dans la partie primaire du mot cursif (ascendant, descendant, les boucles). Par ailleurs, les boucles constituent les primitives les plus informatives dans la zone centrale du mot [SOU 06]. Pour la reconnaissance des caractères, le calcul des moments, les projections, les concavités et les convexités sont appliqués dans un nombre relativement important de travaux [ALY 92], [MAH 94].
En effet, la sélection des primitives pour l'arabe reste une étape complexe à cause volume des corpus des formes à considérer. Le nombre de fontes et styles est important pour l'imprimé. Dans le cas du manuscrit, notamment omni-scripteurs sans contraintes, les formes sont innombrables et difficiles à modéliser. De plus, certaines caractéristiques des lettres, particulièrement les points diacritiques et les boucles, sont sensibles au bruit et à la

dégradation. Donc, on peut dire qu'il existe un manque d'étude préliminaire permettant de sélectionner les primitives les plus discriminantes, pour un style et une qualité donnée d'écriture, et les mieux appropriées relativement au type de classifieur retenu.

7. Conclusion

Nous avons présenté dans ce chapitre les concepts généraux liés à la reconnaissance optique des caractères, en précisant les différents aspects qui influencent l'OCR. Nous avons ensuite exposé les principales propriétés morphologiques de l'écriture arabe. Nous avons également rappelé les différentes étapes intervenant dans un système de reconnaissance en citant quelques travaux déjà réalisés dans le domaine de la reconnaissance de l'écriture arabe, tout en soulevant les problèmes liés au traitement de cette écriture. Les spécificités liées aux caractéristiques morphologiques de l'écriture arabe, dans leur grande majorité, compliquent la tâche de l'OCR à différents niveaux de traitement. Nous avons vu que le problème majeur se ramène à la segmentation des chaînes de caractères et à l'extraction de caractéristiques, ce qui est dû essentiellement à la cursivité de l'écriture et à la sensibilité de certaines caractéristiques de l'arabe à la dégradation.
Le tableau 1.4 en fin de chapitre, regroupe certains systèmes de reconnaissance de l'écriture arabe en précisant pour chacun le mode utilisé "en ligne" ou "hors ligne", l'approche de reconnaissance globale ou analytique, le type de segmentation externe ou interne, la représentation choisie, ainsi que le score réalisé (lorsqu'ils sont précisés).
En outre, l'absence d'outils de test ou des protocoles de validation communs, surtout pour l'écriture arabe, constitue un problème majeur aux chercheurs en AOCR. En effet, ces outils permettraient, entre autres, d'évaluer de manière cohérente les résultats des différentes techniques développées, afin de pouvoir conclure convenablement quant aux résultats trouvés. Récemment, un certain nombre de tentatives de constructions de bases d'images, pour la reconnaissance de textes arabes sont reportés dans la littérature. A titres d'exemple, la base "IFN/ENIT" de mots arabes manuscrits contient des noms de 946 villes/villages
tunisiens avec leurs codes postaux. Cette base est disponible pour la recherche à travers le monde. Un ensemble de bases de mots arabes est décrit dans [SOU 06].

	Système	Approche	Segmentation	Primitives	Classification	Performance
[ABD 90]	Hors ligne, Imp MF	Analytique	Implicite	Structurelles/Statistiques	Structurelle/Statistique/Arbre de décision	RC 99%
[ALY 92]	Hors ligne,Manus,Imp	Analytique	Implicite	Moments	Classifieur bayésien	RC 99,5%
[AME 94]	Hors ligne,Mots,Manus	Globale	-	Structurelles	Dictionnaire	
[AMI 89]	Hors ligne, MF	Analytique	Implicite	Chaine de code	Arbre de décision	SC 98,9%, RC 83%
[AMI 92]	Hors ligne, mots	Analytique	Implicite	Structurelles/Chaine de codes	-	-
[AMI 94,96]	Hors ligne, Car, MS	-	-	Structurelles	Réseau de neurone/Classiication structurelle	RC 90-92%
[AMI 97]	Hors ligne, mots	Globale	-	Structurelles	Réseau de neurone	RC 98%
[AMI a00]	Hors ligne,Mots,Imp,MF	Globale	-	Structurelles	Arbre de décision	RC 92%
[AMI b00]	Hors ligne,Mots,Imp,MF	Globale	-	Structurelles/Statistiques	Réseau de neurone flou	RC 95,25%
[AYA 00]	Hors ligne,Chiffre,Manus	-	-	Statistiques/Morphologiques	Neuro-flou	RC 95,12%
[AYA 02]	Hors ligne,Chiffre,Manus	-	-		SVM	RC 95%
[AYA 04]	Hors ligne,Chiffre indiens,Manus	-	-	Statistiques/Structurelles	SVM	RC 89%
[AZI b02]	Hors ligne,Mots,Manus	Globale	-	Structurelles/Statistiques	Multiclassifieurs (classifieurs neuronaux)	-
[BEN 00]	Hors ligne, Mots,Manus	Hybride (Globale/Analytique)	Implicite	Structurelles	HMM	-
[CHE 98]	Hors ligne,Mots,Manus	Globale	-	Indices visules (structurelles)	HMM	RC 84,70%
[ELK 90]	Hors ligne, Imp	Analytique	Implicite	Moments	Distances	RC 95-100%
[ESS 99]	Hors ligne,Mots	Globale	-	Structurelles	PHHM	RC 99,84%
[FAH 05]	Hors ligne,Mots,Manus	Globale	-	Structurelles	Multiclassifieurs (réseau de neurone, KPPV, KPPV flou)	RC 94%
[MAH 94]	Hors ligne, Manus,Car	-	-	Descripteurs de Fourier	KPPV	RC 98%
[MIL 98]	Hors ligne,Manus	Analytique	Implicite	Topologiques/Statistiques	HMMs	RC 79,5- 82,5%
[OLI 96]	Hors ligne, Manus,MS	Analytique	Implicite	Chaine de code	-	SC 97,41%
[SAR 05]	Hors ligne,Mots,Manus	Analytique	Explicite	Topologiques/Statistiques	Classifieur neuronal	SC 86%-RC 84%
[SOU b97]	Hors ligne,Manus	Analytique	Implicite	Statiqtiques/structurelles	Réseau de neurone	RC 76,17-85,75%
[SOU 06]	Hors ligne,Mots,Manus	Globale	-	Structurelles	Classifieur perceptuel	RC 91,81%
[SOU 06]	Hors ligne,Mots,Manus	Globale	-	Structurelles	Classifieur neuro-symbolique	RC 92%

Imp : Imprimé, Manus : Manuscrit, Car : Caractère, MS : Multi scripteurs, MF : Multi fontes, RC : Taux de reconnaissance, SC : Taux de segmentation

TAB 1.5 : *Caractéristiques de quelques systèmes de reconnaissance d'écriture*

Chapitre 2

Approches de Classification & Apprentissage Statistique

Dans le présent chapitre, nous commençons par présenter les différentes méthodes de classification. Par la suite, nous introduisons les concepts fondamentaux du Machine Learning (ML) supervisé dans le cas de la classification. Ensuite, nous effectuons une analyse statistique de l'apprentissage, principalement basée sur la théorie de Vapnik-Chervonenkis.

1 Les approches de classification

Les méthodologies de reconnaissance sont nombreuses. En réalité, il n'existe pas de méthode *"spécifique*" pour la reconnaissance de caractères, ce sont plutôt des adaptations des méthodes d'optimisation issues des méthodes classiques de reconnaissance de formes particulièrement en traitement de signaux ou d'images. Dans ce qui suit, nous rappelons les principales approches de classification.

1.1. Approche statistique

Elle est basée sur l'étude statistique des mesures que l'on effectue sur les formes à reconnaître. L'étude de leur répartition dans un espace métrique et la caractérisation

statistique des classes, permettent de prendre une décision de reconnaissance du type *"plus forte probabilité d'appartenance à une classe"*.

Les approches statistiques bénéficient des méthodes d'apprentissage automatique qui s'appuient sur des bases théoriques fondées, telles que la théorie de la décision bayésienne, les méthodes de séparation linéaire, les méthodes de classification non supervisée….En reconnaissance, le problème revient à affecter une forme inconnue à l'une des classes obtenues pendant l'apprentissage.

Dans la deuxième partie de ce chapitre, nous reviendrons en détail sur la théorie d'apprentissage statistique.

Parmi les méthodes statistiques les plus couramment utilisées, nous décrivons les trois suivantes :

1.1.1. Classification bayésienne

La théorie bayésienne de décision a été élaborée dans le cadre des statistiques mathématiques, son application à la reconnaissance des formes a été formalisée par CHOW en 1965 [CHO 65, cité dans HAL 04].

La méthode de classification statistique bayésienne définit l'appartenance d'une forme à une classe avec un minimum d'erreurs et définit le risque de la décision à prendre. La règle de décision bayésienne est un cas particulier de la règle général de décision avec coût, celle-ci consiste à associer à chaque classe un coût d'erreur de décision. Le critère à minimiser est, dans ce cas, le coût d'erreur qui traduit ainsi le risque de décision prise. La décision bayésienne associe, par exemple, un coût unitaire à chacune des erreurs et un coût nul à chacune des bonnes réponses. L'objectif est donc de construire un système dont la probabilité d'erreur globale est minimale. La décision qui minimise cette erreur est celle qui associe à chaque point X de R^n représentant une forme inconnue, la classe C_i (i=1…c), dont la probabilité en X est la plus forte càd $P(C_i/X)$ est maximal.

1.1.2. Méthode des K plus proches voisins (KPPV)

Cette méthode consiste, étant donné un point $x \in R^n$ représentant le caractère à reconnaître, à déterminer la classe de chacun des k points les plus proches de x parmi l'ensemble des caractères d'apprentissage et à retenir pour la décision la classe la plus représentée. Si $k=1$,

x est donc attribué à la classe de son plus proche voisin. L'algorithme de KPPV renvoie les k formes les plus proches de la forme à reconnaître suivant un critère de similarité. Une stratégie de décision permet d'affecter des valeurs de confiance à chacune des classes en compétition et d'attribuer la classe la plus vraisemblable (au sens de la métrique choisie) à la forme inconnue. Dans sa version initiale, le critère de similarité entre deux formes est basé sur la distance euclidienne.

El Wakil et Skoukry [ELW 89] utilisent un processus de classification de trois niveaux : le premier niveau est la recherche dans un dictionnaire, le deuxième est un classifieur 1 plus proche voisins, le troisième est un classifieur k plus proches voisins. Dans [FAH 05], les auteurs proposent un système pour la reconnaissance des mots appartenant au vocabulaire des montants littéraux. L'étape de reconnaissance est effectuée par une combinaison parallèle de trois types de classifieur : réseau de neurones, k plus proches voisins, k plus proches voisins flou.

La méthode des KPPV présente l'avantage d'être facile à mettre en œuvre et fournit de bons résultats. Son principal inconvénient est lié à une vitesse de classification faible dû au nombre important des distances à calculer [HAL 04].

1.1.3. Discrimination fonctionnelle

Ces méthodes sont basées sur la définition des fonctions permettant de séparer des classes représentées par les vecteurs de leurs échantillons. Ainsi, on peut distinguer deux catégories de fonctions :

- **Fonctions de discrimination linéaires :** le problème consiste en la recherche d'un hyperplan séparateur permettent de discriminer les classes. Dans le cas de non séparabilité des classes, on passe dans un espace de plus grande dimension ou l'on cherche un séparateur linéaire : ce sont les méthodes SVM (Support Vector Machines) [VAP 98].
- **Fonctions de discrimination linéaires par morceaux :** Il arrive que les classes ne soient pas linéairement séparables mais qu'elles soient formées de sous classes qui, elles, sont linéairement séparables. La méthode consiste à traiter chaque sous-classe comme une classe distincte [BEL 02].

1.2. Approche structurelle

Si l'approche statistique permet de se placer dans un cadre mathématique fondé, elle présente cependant le défaut de ne pas tenir compte de la nature physique des formes et de leurs mesures : en particulier une métrique de type mathématique oblige à considérer les coordonnées caractérisant un point de façon indifférenciée. En utilisant de tel modèle, on ne peut pas exprimer certains types de contraintes auxquelles pourtant obéissant les formes étudiées.

Les méthodes structurelles se basent sur la structure physique des caractères. Elles cherchent à décomposer le caractère en primitives et à décrire leurs relations. Les primitives sont de type topologique comme un arc, une boucle…etc, et une relation peut être la position relative d'une primitive par rapport à une autre. Dans cette approche, on distingue plusieurs méthodes.

1.2.1. Méthodes syntaxiques

Ces méthodes sont directement issues de la théorie des langages formels. Elles se basent sur une grammaire formelle sous formes de règles de production où le vocabulaire est constitué de primitives. La reconnaissance consiste à déterminer si la phrase de la description du caractère peut être générée par la grammaire [BAP 88]. Plusieurs exemples d'applications sont considérés dans [BEL 92].

Les méthodes à base de règles proposées en 1960 ont été vite abandonnées pour la difficulté à inférer automatiquement des règles générales et réalisables pour des bases de caractères et mots à grand vocabulaire. L'utilisation des règles floues et des grammaires basées sur des informations statistiques sur la fréquence d'occurrence de caractéristiques particulières, ont suscité, de nouveau, l'intérêt d'utilisation de méthodes à bases de règles [TAP 90].

1.2.2. Les graphes

Cette méthode consiste à construire un graphe ou les noeuds contiennent les primitives et les liens entre ces primitives. Amin et al dans [AMI 98] ont utilisé un arbre pour représenter un caractère, les nœuds de cet arbre sont les primitives et les feuilles sont les

étiquettes des caractères. Classifier, un caractère, dans ce cas revient à chercher un chemin à travers l'arbre jusqu'à une feuille. BenOuareth dans [BEN 98] a utilisé le concept d'appariement (Matching) des graphes pour la reconnaissance des mots arabes manuscrits. Il calcul le degré de similarité entre le graphe construit à partir du squelette du mot inconnu, et d'autres graphes de références représentant le lexique de la reconnaissance. Abuhaiba et al. [ABU 94], après avoir transformé le squelette d'un caractère en un arbre convenant à la classification, utilisent le concept des ensembles des directions pour modéliser les caractères arabes. La méthode consiste à établir un ensemble de modèles des graphes flous contraints pour les caractères arabes (appelés FCCGM : Fuzzy Constraind Character Graph Models). Ces modèles sont tout simplement des graphes, ayant des arcs étiquetés d'une manière floue, utilisés comme des prototypes de caractères.

1.2.3. Les comparaisons de chaînes

Les caractères sont représentés par des chaînes de primitives. La comparaison du caractère traité avec un modèle de référence, consiste à mesurer la ressemblance entre deux chaînes et à se prononcer sur celle-ci. La mesure de ressemblance peut se faire par calcul de distance ou par examen de l'inclusion de toute ou une partie d'une chaîne dans l'autre [ESS 99].

1.3. Approche connexionniste

Les réseaux de neurones ont connu un essor important grâce l'algorithme de retro-propagation du gradient. Ce classifieur a trouvé application dans beaucoup de domaines tels que la reconnaissance de caractères, reconnaissances de visages…etc.
La démarche connexionniste est très intéressante pour la classification car les réseaux neuronaux sont dotés de capacités d'apprentissage. Les algorithmes mis au point pour les entraîner permettent de modéliser des fonctions de discrimination très complexes à partir d'exemples. Même si chaque neurone effectue une fonction de discrimination très simple, les connexions entre neurones permettent de combiner les fonctions entre elles pour aboutir à des fonctions complexes difficiles à analyser et à spécifier explicitement. La capacité d'apprentissage, à partir des exemples, est importante surtout dans le cas où la constitution de recueils d'expertise pour un système expert devient difficile (reconnaissance intuitive

ou implicite). Les réseaux de neurones peuvent apprendre à retrouver des règles à partir des exemples. De plus, dans les réseaux de neurones, la mémoire correspond à une carte d'activation de neurones. Cette carte est en quelque sorte un codage du fait mémorisé ce qui attribue à ces réseaux l'avantage de résister aux bruits (pannes) car la perte d'un élément ne correspond pas à la perte d'un fait mémorisé. Cependant, l'un des principaux reproches fait aux réseaux de neurones est leur incapacité à expliquer les résultats qu'ils fournissent. Ces réseaux se présentent comme des boîtes noires dont les règles de fonctionnement sont inconnues. La qualité de leurs performances ne peut pas être mesurée par des méthodes statistiques, ce qui cause une certaine méfiance de la part des utilisateurs. En pratique, se pose dans tous les cas un problème essentiel, relatif à l'architecture du réseau neuronal car aucune procédure formelle n'est disponible pour répondre aux questions (dans le cas du PMC) : combien faut-il mettre de couches de neurones, et combien de neurones par couches ? [SOU 97]

En OCR, les primitives extraites sur une image d'un caractère constituent les entrées du réseau. La sortie activée du réseau correspond au caractère reconnu. Amin et Al sadoun [AMI a96] proposent une approche c onnexionniste pour la reconnaissance des caractères arabes manuscrits. Après la construction du graphe représentant le caractère, des caractéristiques comme les boucles, les courbes sont extraites à partir de ce graphe. Un réseau de neurone de cinq couches est utilisé pour classifier ces primitives.

Amin et Mansour [AMI 97] ont utilisé un réseau de neurones de trois couches, avec une couche d'entrée de 270 neurones pour la reconnaissance des textes arabes manuscrits.

Souici et al [SOU a97] après avoir extrait des caractéristiques statistiques et géométriques des caractères arabes manuscrits, utilisent un perceptron pour la classification de ces caractéristiques. Ils ont obtenu un taux de reconnaissance de 82%.

1.4. Approche stochastique

L'approche stochastique consiste à utiliser la modélisation par un processus stochastique en prenant en compte la variabilité des caractères. Les paramètres des modèles sont calculés à l'aide des probabilités calculées de manière très fine par apprentissage. Le caractère est considéré comme un signal continu observable dans le temps à différents endroits constituant des *"états d'observations"*. Le modèle décrit ces états à l'aide de

probabilités de transitions d'états et de probabilités d'observation par état. La comparaison consiste à chercher dans ce graphe d'états, le chemin de probabilité forte correspondant à la suite d'éléments observés dans la chaîne d'entrée.

Ces méthodes sont robustes et fiables du fait de l'existence d'algorithmes d'apprentissage efficaces. Si l'apprentissage est lent, le reconnaissance est par contre très rapide car les modèles comprennent généralement peu d'états et le calcul est relativement immédiat.

Parmi les méthodes stochastiques, les HHM suscitent un vif intérêt dans la communauté des chercheurs en reconnaissance de l'écriture.

Les chaînes de markov cachés (HMM) sont définis comme étant des modèles doublement stochastiques. Le premier processus stochastique associé est une chaîne de markov du premier ordre qui n'est pas directement observable, que nous appelons *processus caché.* Le second processus est un processus d'émission qui génère une séquence de variables prenant ses valeurs dans un espace dit d'observations [BEL 02].

Les HMM connaissent un essor important avec une large gamme d'applications, ils ont été appliqués avec succès aux problèmes de reconnaissance automatique de la parole (RAP). Plusieurs problèmes ouverts en reconnaissance automatique de l'écriture sont analogues à des problèmes connus en RAP. Ainsi, les distorsions élastiques des caractères manuscrits sont analogues aux distorsions temporelles des signaux en parole ; le problème de la segmentation du manuscrit cursif est similaire à celui de la segmentation et de la détection des points limites en parole continue [BEN 00]…etc. De ce fait, tout comme la parole, l'écriture se prête bien à une modélisation stochastique. Amin et Mari [AMI 89] ont utilisé l'algorithme de viterbi pour la reconnaissance de textes arabes imprimés multifontes.

Benouareth [BEN 00] a proposé un système pour la reconnaissance des mots arabes manuscrits à vocabulaire limité. L'ensemble d'informations obtenu après une analyse locale du mot, permettra une représentation du mot en séquences de codes. Ces codes, considérés comme des observations d'états, seront utilisés par un classifieur HMM. Ainsi, chaque mot dans le lexique de la reconnaissance est modélisé par un HMM résultant de la concaténation des modèles des HMM-lettres, correspondant à la séquence de lettres composant le mot.

1.5. Les approches hybrides et multi-classifieurs

Afin d'améliorer les performances des systèmes de reconnaissance, la tendance actuelle est de construire des systèmes hybrides ou multi-classifieurs qui utilisent des primitives et/ou des approches de natures différentes en combinant plusieurs classifieurs. Cette hybridation se base sur la complémentarité qui peut exister entre deux approches et tente aussi de les faire coopérer pour résoudre un problème donné. Chaque approche prendra à sa charge le traitement d'une tâche qui s'accommode le mieux avec son style de raisonnement, de plus elle viendrait pallier les inconvénients de l'autre.

Souici [SOU 06] propose une approche hybride neuro-symbolique pour la reconnaissance de mots arabes manuscrits. Une base de règles permet de modéliser les connaissances théoriques relatives à la description des mots en utilisant leurs caractéristiques structurelles perceptuelles. Elle est ensuite compilée sous forme d'un réseau neuronal multicouches. Ce réseau est ensuite affiné par un apprentissage empirique sur un base d'exemples.

Farah et al. [FAR 05], proposent un système multiclassifieur qui combine diverses sources d'informations pour reconnaître les 48 mots manuscrits appartenant au vocabulaire des montants littéraux. L'étape de reconnaissance est effectuée par une combinaison parallèle de trois types de classifieurs (réseau de neuronal de type perceptron muticouches, K plus proches voisins, K plus proches voisins flou) utilisant des caractéristiques globales perceptuelles des mots (nombre de sous mots, ascendants, descendants, boucles et points diacritiques). Le contexte grammatical des montants littéraux est utilisé pour prendre une décision finale sur les mots candidats obtenus.

Après l'extraction des caractéristiques à partir d'une image de mot présentée en entrée du système, chaque classifieur fournit en sortie une liste triée des trois meilleurs candidats avec les valeurs de confiance accordées, par le classifieur, à chacune des proposition. Pour chaque candidat, les valeurs de confiance sont normalisées, pour correspondre à des probabilités à posteriori des classes, puis sommées de sorte à établir une liste finale triée par ordre décroissant de valeurs de confiance. Le candidat en tête de liste sera la réponse du combineur.

Les taux de reconnaissance obtenus par la combinaison des classifieurs sont de l'ordre de 94%. Une étape de post-traitement a permis d'améliorer les résultats de près de 2%.

Dans [AZI b02], les auteurs proposent un système multi-classifieurs (dénommé MCNF) pour la reconnaissance des mots manuscrits en utilisant la combinaison parallèle de trois classifieurs neuronaux basés, chacun sur un type de caractéristiques.
Chaque classifieur neuronal est un perceptron à trois couches entraîner pour reconnaître les mots représentés par un ensemble homogènes de caractéristiques. Dans la phase de décision, chaque mot en entrée sera représenté par des vecteurs de caractéristiques statistiques, perceptuelles et géométriques. Le système affecte chaque vecteur de caractéristique au sous réseau correspondant et fournit des sorties de types mesure. Le résultat final sera la combinaison de ceux issus des différents sous réseaux neuronaux avec un calcul flou qui prend en compte les sorties et l'importance de chacun des réseaux en fonction des résultats obtenus dans la phase de test [AZI a02].

2. Apprentissage statistique

La résolution de problème par la construction de machines capables d'apprendre à partir des expériences caractérise l'approche fondamentale du *Machine Learning* (ML) [CAL 03]. Parmi les techniques du ML, on trouve, entre autres, les méthodes supervisées et non supervisées.
L'objectif de *l'apprentissage statistique* à partir d'exemples étiquetés appelé aussi *apprentissage supervisé* est de construire une fonction qui permet d'approcher au mieux une fonction inconnue qui génère des données aléatoires, indépendantes et identiquement distribuées (iid), et dont nous ne disposons que de quelques exemples. La figure suivante illustre un tel processus :

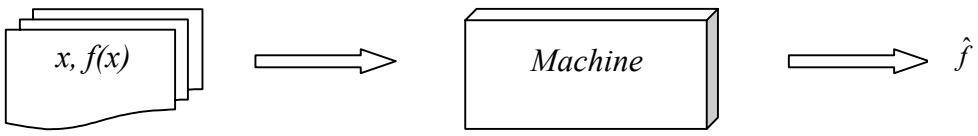


FIG. 2.1- *Machine Learning supervisé, $\hat{f}$ est une estimation de f(x) [CAL 03]*

Un système d'apprentissage statistique à partir d'exemples est composé de trois modules [VAP 98] :

- ***Un générateur*** : qui génère des données aléatoires appelées les vecteurs d'entrée. Ces vecteurs sont indépendants et identiquement distribués suivant une distribution de probabilité inconnue *P(x).*
- ***Un superviseur*** : qui associe à chaque vecteur d'entrée *x* une sortie *y* (la classe d'appartenance) suivant une distribution de probabilité également inconnue *P(x,y)*.
- ***Une machine d'apprentissage*** *:* qui permet d'implémenter une famille de fonctions $f_\alpha(x)$, $\alpha \in \Lambda$ où Λ est un ensemble de paramètres. Ces fonctions doivent produire pour chaque vecteur d'entrée *x* une sortie ŷ la plus proche possible de la sortie y du superviseur.

La figure suivante représente ces trois modules.

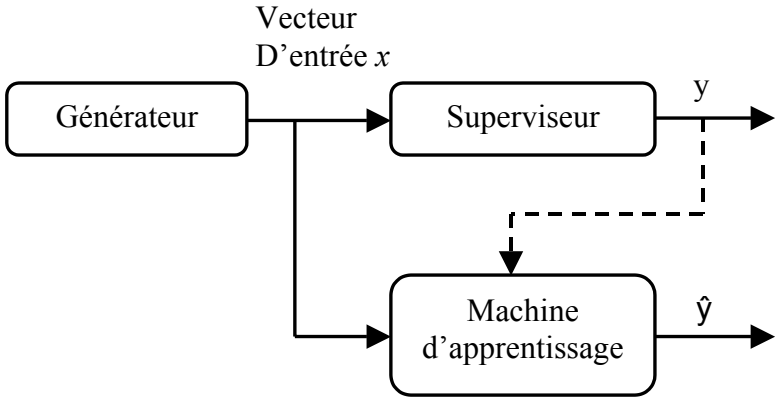


FIG. 2.2- *Les modules d'un système d'apprentissage [VAP 98]*

Le problème de l'apprentissage statistique à partir des exemples apparaît dans plusieurs domaines divers et variés, par exemple : la prédiction des termes des séries temporelles, la régression, la reconnaissance de formes, la fusion, etc.

Dans le cadre de ce travail, nous nous intéressons principalement au problème d'apprentissage pour la reconnaissance de formes.

2.1. Apprentissage statistique supervisé pour la reconnaissance de formes

Effectuer une classification consiste à déterminer une règle de décision capable, à partir d'observations externes, d'assigner un objet à une classe parmi plusieurs. Le cas le plus simple consiste à discriminer deux classes. D'une manière plus formelle, la classification binaire revient à estimer une fonction f_α: X $\rightarrow$ {-1, +1} à partir d'un ensemble d'apprentissage constitué de couples (x_i, y_i), qu'on suppose i.i.d, suivant une distribution de probabilité $P(x,y)$ inconnue, tel que:

$$D_l = \{(x_i, y_i) \in X \times Y \quad où \quad i = 1,...,l \; et \quad Y = \{-1,+1\}\}$$

De sorte que, f_α classifie correctement les exemples inconnus. Par exemple, on assigne x_t à la classe $(+1)$ si , $f_\alpha(x_t) \geq 0$, et à la classe (-1) sinon.

Soit F une famille de fonctions telle que :

$$F = \{f_\alpha(x) / \alpha \in \Lambda\}$$

L'objectif est de trouver $f_{\alpha^*} \in F$ telle que l'estimation $f_{\alpha^*}(x) = \hat{y}$ soit la meilleure possible. La fonction f_{α^*} correspondra à celle qui minimise le risque réel qui s'écrit :

$$R(\alpha) = \int L(f_\alpha(x), y) dP(x, y) \qquad (2.1)$$

La fonction $L(f_\alpha(x), y)$ est appelée fonction de coût, sa forme dépend principalement de la tâche à accomplir. Cette fonction mesure la distance entre la réponse produite $f(x)$ et la réponse désirée y_i. Pour la reconnaissance de formes, la fonction de coût prend la forme suivante :

$$L(f_\alpha(x), y) = \begin{cases} 0 & si \quad y = f_\alpha(x) \\ 1 & si \quad y \neq f_\alpha(x) \end{cases} \tag{2.2}$$

Malheureusement, le risque réel R(α) ne peut être directement minimisé dans la mesure où la distribution de probabilité sous-jacente *P(x, y)*, est inconnue.

La théorie l'apprentissage statistique développé par Vapnik permet d'estimer le risque réel en utilisant un principe d'induction à partir de ensemble de données D_l. Cette théorie emploie le principe de Minimisation du Risque Empirique [VAP 98].

2.2. Minimisation du Risque Empirique (ERM)

La minimisation du risque empirique consiste à employer un ensemble de données D_l pour construire une approximation stochastique du risque réel. Donc, on va chercher une fonction de décision proche de celle optimale à partir de l'information dont on dispose [MAK 01]. Le risque empirique est défini comme suit [VAP 98] :

$$R_{emp}(\alpha) = \frac{1}{2l} \sum_{i=1}^{l} L(f_\alpha(x_i), y_i) \tag{2.3}$$

Notons que dans cette équation, la distribution de probabilité *P(x, y)* n'apparaît pas. Pour bien sélectionner la fonction f_{α^*}, il faut trouver l'ensemble des paramètres $\alpha^* \in \Lambda$ telle que $R_{emp}(\alpha)$ soit minimale. C'est le principe de la minimisation du risque empirique (ERM). Cette approche consiste à minimiser R_{emp} sur Λ en s'appuyant sur le fait que $R_{emp}(\alpha)$ converge vers R (α) lorsque la taille des données tend vers l'infini [EVE 00] (la loi des grands nombres). Cela définit la consistance de l'approche. Le théorème fondamental sur la consistance a été démontré en 1989 par Vapnik et Chervonenkis [VAP 91]. La figure suivante présente une interprétation visuelle de cette propriété.

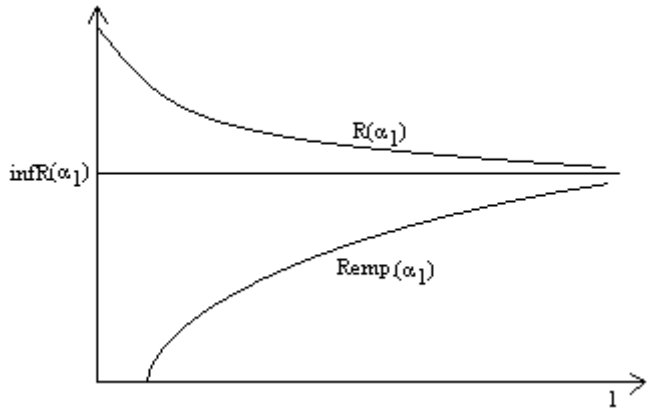


FIG. 2.3- *Consistance de l'approche ERM*

Bien que $R_{emp}(\alpha)$ est une estimée non biaisée du risque réel $R(\alpha)$, l'ensemble des paramètres α^* qui optimise le risque empirique $R_{emp}(\alpha)$ n'est pas le même qui optimise le risque réel $R(\alpha)$ [EVE 00]. Ainsi, en minimisant le risque empirique sur l'ensemble d'apprentissage, nous obtenons un modèle qui est efficace sur cet ensemble, mais dont nous n'avons a priori aucune garantie de performance sur de nouveaux exemples. Ce problème est bien connu sous le nom de sur-apprentissage (overfitting). La figure 2.4 représente une interprétation de ce phénomène.

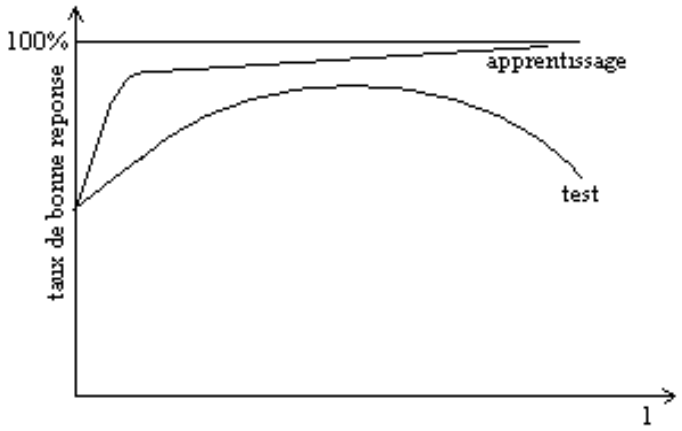


FIG. 2.4 - *L'effet du phénomène du sur-apprentissage*

La théorie employée pour éviter le sur-apprentissage est de restreindre la complexité de la classe *F* des fonctions à laquelle appartient la meilleure fonction de décision f_{α^*} [VAP

98]. Intuitivement, une fonction de décision simple capable de discriminer correctement les données est préférable à une fonction complexe. Certains chercheurs ont introduit un terme de régularisation pour limiter la complexité de la classe *F* [KLA 01]. Cela soulève le problème du modèle de sélection ; càd comment trouver la fonction de décision qui a une complexité optimale. Pour résoudre ce problème, Vapnik et Chervonenkis ont mis en place le principe de la *Dimension VC* et la *Minimisation du Risque Structurel* [VAP 98].

2.3. La Dimension VC

Dans le prochain chapitre, nous présentons une méthode de classification basée sur un hyperplan séparateur. *La dimension VC* est un terme de capacité particulièrement bien adapté à ce genre de classifieur. En effet, nous verrons dans la suite qu'il est possible d'agir directement sur cette dimension en jouant sur la marge entourant l'hyperplan.

Vers la fin des années soixante, Vapnik et Chervonenkis ont introduit un nouveau concept nommé *VC-Dimension*, qui présente une sorte de mesure de capacité de calcul d'une famille de fonctions *F*. La *dimension VC* d'un ensemble de fonctions *F*, notée *h* est le nombre maximum de points pouvant être séparés de toutes les manières possibles par les fonctions de *F* [BUR 98]. Cela veut dire qu'il doit exister une configuration de *h(= VC(F))* points, telles que les fonctions $f \in F$ peuvent leur assigner les 2^h combinaisons de labels possibles. Notons que la dimension d'une famille peut être infinie.

Pour concrétiser ce concept, considérons trois points représentés dans $\Re^2$. Supposons que la famille des fonctions f correspond aux droites de $\Re^2$:

$$y = ax + b \tag{2.4}$$

La dimension VC de F est 3 car on peut trouver une configuration de trois points séparables de toutes les façons possibles, par contre on ne peut trouver aucune configuration de 4 points (ou plus) rendant une telle discrimination possible. Dans le figure 2.5, Les ronds vides et pleins représentent respectivement les points assignés positivement et négativement. La flèche représente le coté de la droite où les points seront classés positivement [BUR 98].

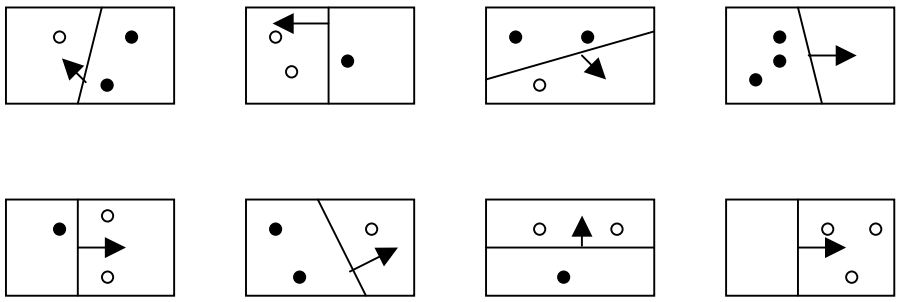


FIG. 2.5 - *Exemple de configuration de trois points séparables de toutes les manières possibles par les droites de* $\Re^2$ [BUR 98].

2.4. La Théorie de Vapnik Chervonenkis

Une manière de contrôler la complexité d'une classe de fonction est donnée par la théorie de Vapnik Chervonenkis et le principe de la minimisation du risque structurel, qui a été mis au point en 1979 [MAH 03]. Le risque empirique, par ailleurs, constitue une estimation assez optimiste du risque réel dont la minimisation ne garantit pas la convergence vers la solution acceptable [AYA 04].

Cette théorie fournie des bornes sur l'erreur de généralisation. Ainsi, Vapnik et Chervonenkis ont démontré que $\forall \alpha \in \Lambda$ et pour $l > h$ où est la dimension VC de la classe F et l est la taille de l'ensemble d'exemples d'apprentissage, l'inégalité suivante sera vérifiée avec une probabilité d'au moins égale à $1-\eta$:

$$R[\alpha] \leq R_{emp}[\alpha] + \phi(\frac{h}{l}, \frac{\log(\eta)}{l}) \tag{2.5}$$

Le membre droite de l'inégalité (2.3) est appelé *Risque garanti,* il est composé de deux termes : le risque empirique et un terme appelé *terme de confidence* donné par :

$$\phi\left(\frac{h}{l}, \frac{\log(\eta)}{l}\right) = \sqrt{\frac{h\left(\log\frac{2l}{h}+1\right) - \log\left(\frac{\eta}{4}\right)}{l}} \tag{2.6}$$

Le terme de confidence dépend du rapport $\frac{l}{h}$; si $\frac{l}{h}$ est suffisamment grand, le terme prédominant du risque garanti est le risque empirique. Ainsi, la minimisation du risque

empirique suffit pour garantir une faible valeur du risque réel R(α). En revanche, lorsque la taille *l* de l'échantillon d'apprentissage est petite, le rapport $\frac{l}{h}$ n'est pas suffisamment grand (Vapnik considère comme petite la taille de l'échantillon dés lors que $\frac{l}{h}$<20) et le terme de confidence peut prendre une valeur importante. Il s'ensuit que la simple minimisation du risque empirique ne garantit plus une faible valeur de risque effectif R(α). La seule façon pour garantir la minimisation du risque réel R(α) consiste à contrôler la dimension VC [VAP 98] ; puisque la taille maximum de l'échantillon d'apprentissage est généralement fixée par les conditions de l'expérience ou du problème à traiter. Vapnik et Chervonenkis proposent d'appliquer un nouveau principe inductif qu'ils nomment « *Principe de Minimisation du Risque Structurel* » [VAP 98, EVE 00].Ce principe est basé sur la minimisation conjointe des deux causes de l'erreur : le risque empirique et le terme de confidence ϕ.

2.5. Minimisation du Risque Structurel (SRM)

La limite supérieure du risque réel (erreur de généralisation R(α)) définie dans l'inégalité (2.3) constitue un principe essentiel dans la théorie des *Machines à Vecteurs de Support* (SVM). Vapnik a introduit le principe de minimisation du risque structurel pour contrôler l'erreur de généralisation. La méthode qu'il propose consiste à définir des structures sur l'ensemble *F* des fonctions $\{f_\alpha : \alpha \in \Lambda\}$ [VAP 98]. On définit une structure sur *F* comme étant une suite de sous ensembles emboîtés $Fi = \left\{ f_i^\alpha / \alpha \in \Lambda_i, \Lambda_i \subset \Lambda \right\}$ Ces structures sont de plus en plus complexe, telles que :

$$F_1 \subset F_2 \subset F_3 \subset ... \subset F_l \subset ....$$

et la capacité *h* dépend de l'ensemble des fonctions que la machine peut inférer. La dimension VC *(h)* de ces structures F_i forment une suite croissante et vérifiant l'inégalité suivante : $h_1 < h_2 < h_3 < ... < h_l < ....$

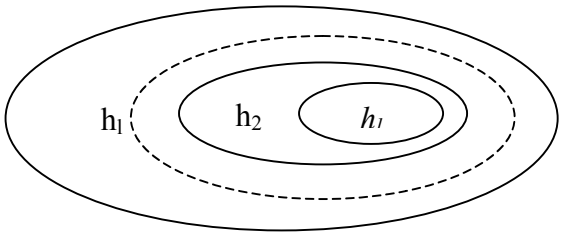


FIG. 2.6- *Minimisation du risque structurel. Les sous ensembles sont ordonnés selon leur Dimension VC*

En pratique, lorsque la dimension VC du modèle augmente le risque empirique décroît, tandis que le terme de confidence croit. La courbe de la figure 2.7 montre que le terme de confidence ϕ varie en fonction croissante en fonction de h. le terme ϕ est une fonction monotone croissante en fonction de h pour n'importe quelle valeur de l.

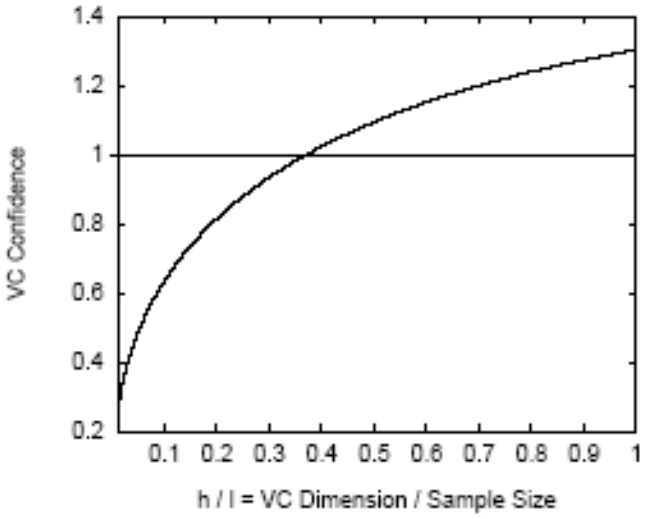


FIG. 2.7 – *Variation du terme de confidence en fonction de h. [BUR 98]*

La borne sur le risque réel R(α) (le risque garanti), qui est la somme de deux quantités, atteint son minimum pour une valeur optimale de la dimension VC. Ainsi, la minimisation du risque structurel vise à choisir une fonction f_α parmi l'ensemble de fonctions $Fi = \left\{ f_i^\alpha / \alpha \in \Lambda_i, \Lambda_i \subset \Lambda \right\}$ possibles, qui minimise le risque empirique sur le sous ensemble F_k, pour lequel le risque garanti est minimal.

Le processus de choisir le bon sous-ensemble de fonctions solutions revient à contrôler la complexité du classifieur en cherchant le meilleur compromis entre une faible erreur empirique et une complexité moindre [VAP 98].
La figure (2.8) représente le comportement du risque empirique, le terme de confidence et le risque garanti en fonction de la dimension VC.

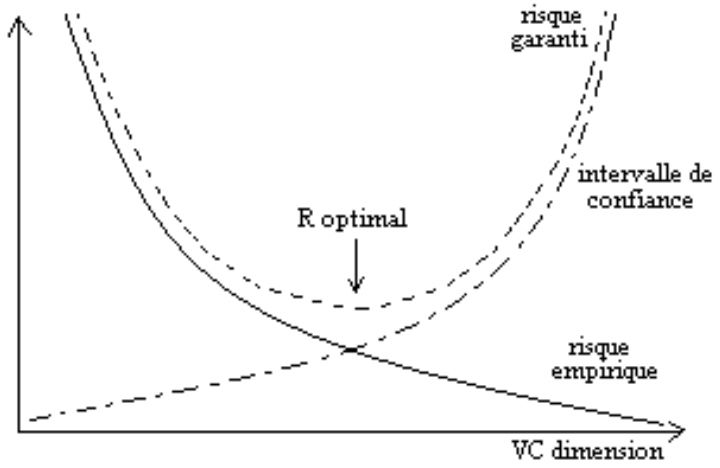


FIG. 2.8 - *Comportement du risque empirique, le terme de confidence et le risque garanti en fonction de la Dimension VC.* [MUL 01]

3. Conclusion

Différentes approches de classification ont été proposées dans la littérature. Dans ce chapitre, nous avons commencé par la présentation de ces approches. Puis, nous nous sommes intéressés de plus près à l'apprentissage statistique en évoquant les fondements de base de la théorie de Vapnik-Chervonenkis. Cette théorie se base sur la minimisation du risque structurel, contrairement aux classifieurs classiques qui se basent sur la minimisation du risque empirique. La minimisation du risque structurel évite le problème de sur-apprentissage et implique la minimisation de la complexité du classifieur. Les concepts présentés dans ce chapitre, constituent un point essentiel dans la théorie des machines à vecteurs de support. Dans le troisième chapitre, nous montrons l'utilisation de ces concepts par les SVM et leur relation avec l'hyperplan de séparation recherchée par le SVM.

Chapitre 3

Les Machines à Vecteur de Support (SVM)

Dans ce chapitre, nous abordons les aspects théoriques des machines à vecteurs de support. Après un bref historique des SVM, la formulation mathématique est présentée. Nous décrivons l'utilisation des machines à vecteurs de support pour la classification et ainsi que la formulation du SVM linéaire. 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 multiclasses.

1. Introduction

L'origine des machines à vecteurs de support (SVM) remonte à 1975 lorsque Vapnik et Chervonenkis ont proposé le principe du risque structurel et la dimension VC pour caractériser la capacité d'une machine d'apprentissage. A cette époque, ce principe n'a pas trouvé place et il n'existait pas encore un modèle de classification solidement appréhendé pour être utilisable. Il a fallu attendre jusqu'à l'an 1982 pour que Vapnik propose le SVM un premier classifier basé sur la minimisation du risque structurel [VAP 82].Ce modèle

était toutefois linéaire et l'on ne connaissait pas encore le moyen d'induire des frontières de décision non linéaires. En 1992, Boser et al. ont proposé d'introduire des noyaux non linéaires pour étendre le SVM au cas non linéaire [BOS 92]. En 1995, Cortes et al. ont proposé une version régularisée du SVM qui tolère les erreurs d'apprentissage tout en les pénalisant [COR 95].

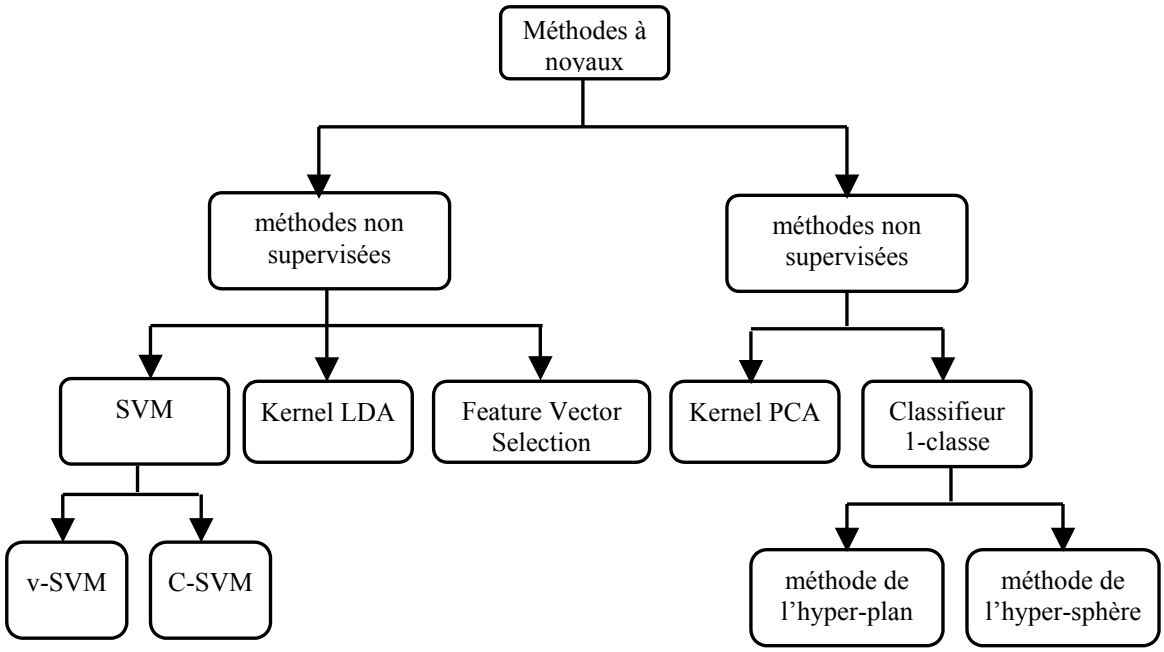


FIG. 3.1- *Arbre de classification des méthodes d'apprentissage à base de noyaux*

Depuis, les SVMs (le pluriel est utilisé pour désigner les différents variantes du SVM) n'ont cessé de susciter l'intérêt de plusieurs communautés de chercheurs de différents domaines d'expertise. Par exemple, Cortes et al. dans [COR 95], et Burges et al.[CHR 97] ont appliqué les SVM à la reconnaissance des chiffres manuscrits isolés, Blanz et al. [BLA 96] ont expérimenté le SVM sur des objets 2D de vues différentes. Osuna et al. ont traité la détection d'images de visages [OSU 97]. Dans la plupart des cas, la performance du SVM égale ou dépasse celle des modèles classiques.

Smola er al. [SMO 98] ainsi que Wahba [WAH 98] ont mis en évidence la ressemblance entre le SVM et la théorie de régularisation. Ils ont démontré qu'associer un noyau particulier à un SVM revient à considérer une pénalisation différente de l'erreur d'apprentissage en maximisant la marge. Ce qui permet de dire que la maximisation de la

marge dans l'espace augmenté est une régularisation de l'apprentissage. Dès lors, le SVM permet de répondre à deux problèmes centraux de la théorie de l'apprentissage statistique :

- Le contrôle de la capacité du classifieur ;
- Le sur-apprentissage.

2. Formulation

Nous considérons la formulation suivante :
Soit l'ensemble D tel que :

$$D_l = \{(x_i, y_i) \in X \times Y \text{ où } i = 1,...,l \text{ et } Y = \{-1, +1\}\}$$

Soit un problème de classification binaire défini par les observations $(x_1,y_1)...(x_l,y_l)$ tirées de manières indépendantes d'une même distribution (iid)[1]. Chacune des données $x_i \ \forall \ i=1...l$ représente un vecteur de caractéristique dans R^n. les variables $y_i = \{+1, -1\} \ \forall \ i=1...l$ représente les classes d'appartenance correspondantes aux données x_i.
Il s'agit d'estimer la fonction de décision $f(x)$ qui approxime au mieux les échantillons de l'ensemble d'apprentissage, telle que $y_i = f(x_i)$ pour toute donnée x_i

3. Le SVM linéaire

Dans cette section, nous décrivons l'utilisation des machines à vecteurs de support pour un problème de classification. Nous présentons la méthode générale de construction de l'hyperplan qui sépare des données appartenant à deux classes. Nous traitons le cas des données séparables, par la suite nous présentons le cas des données non séparables.

[1] Indépendantes et identiquement distribuées

3.1. Cas des données linéairement séparables

3.1.1. Hyperplan de séparation

Supposons que les échantillons de l'ensemble d'apprentissage (les exemples positifs et les exemples négatifs) sont séparables par un hyperplan, càd la fonction de décision peut être exprimée comme suit :

$$f(x) = sign(w.x + b) \quad w \in R^n, b \in R, x \in R^n \tag{3.1}$$

Pour décider à quelle catégorie un exemple *x* appartient, il suffit de prendre le signe de la fonction de décision *y = sign (f(x))*. Géométriquement, cela revient à considérer un hyperplan de séparation qui s'écrit :

$$x_i.w + b = 0 \quad avec \quad w \in R^n, b \in R, x_i \in R \tag{3.2}$$

Tous les exemples de l'ensemble d'apprentissage satisfaisant les contraintes suivantes :

$$\begin{cases} w.x_i + b \geq 1 & si \quad y_i = 1 \\ w.x_i + b \leq -1 & si \quad y_i = -1 \end{cases} \tag{3.3}$$

Ce qui est équivalent à :

$$y_i(w.x_i + b) \geq 1 \quad pour \quad i = 1,...,l \tag{3.4}$$

L'algorithme de construction du SVM cherche un hyperplan de séparation qui maximise la marge *M* qui représente la plus petite distance entre les échantillons de l'ensemble d'apprentissage des deux classes et l'hyperplan séparateur [MUL 01].
La marge *M* peut être mesurée grâce au vecteur *w*. Donc, il est d'usage de considérer les échantillons satisfaisants :

$$|w.x_i + b| = 1 \tag{3.5}$$

Cela garantit que les points les plus proches de l'hyperplan de séparation ont une sortie de valeur absolue égale à 1. Ces points sont appelés *Vecteurs de Supports* [AYA 04]. Il est facile de démontrer, dans ce cas, que la marge de séparation est égale à :

$$M = \frac{2}{\|w\|} \tag{3.6}$$

Dans la figure 3.2, L'hyperplan (ligne continue) séparent les exemples positifs et négatifs (cercles et losanges) de l'ensemble d'apprentissage. Les données sont séparables, donc il existe un vecteur *w* et un biais b tels que $y_i \ (x_i \ . \ w + b) \geq 1$ (i=1..l).la frontière de décision est : {x| w.x + b =0}. La distance mesurée perpendiculairement à l'hyperplan est égal à $2/\|w\|$

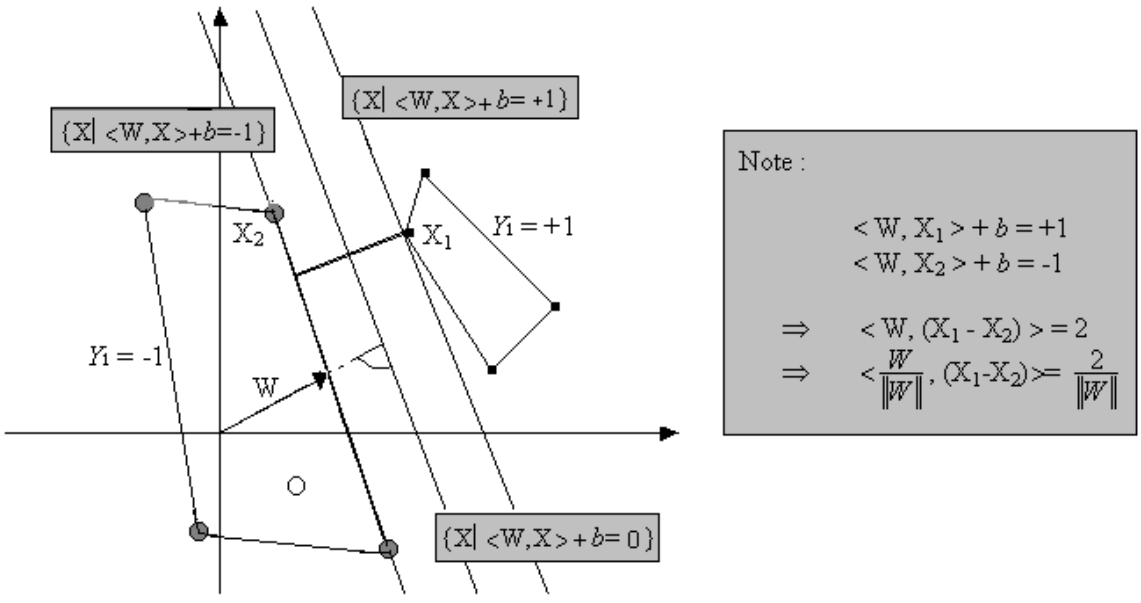


FIG 3.2- *Illustration de l'hyperplan de séparation : Problème de classification binaire [VAP 98].*

3.1.2. Relation avec l'apprentissage statistique

Intuitivement, le fait d'avoir une marge plus large procure plus de "sécurité" lorsque l'on classe un exemple inconnu. Dans la figure 3.3, le même jeu de données est classé par un hyperplan à marge maximale (à gauche) et un hyperplan quelconque ne commettant pas d'erreur sur l'ensemble d'apprentissage (à droite). La boule grisée est un exemple de l'ensemble de test. La partie droite montre qu'avec l'hyperplan optimal, l'exemple de test reste bien classé alors qu'il tombe dans la marge. On constate sur la partie droite qu'avec une plus petite marge, l'exemple est mal classé.

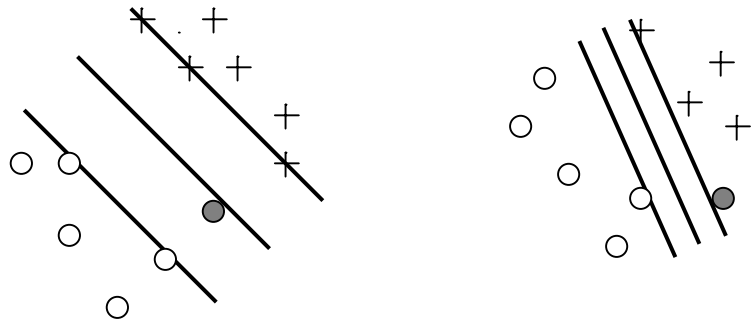


FIG 3.3 – *Comparaison de classification par un hyperplan à marge maximale et un hyperplan quelconque.*

Cette intuition est plus précisément exprimée dans un théorème introduit par Vapnik [VAP 95] faisant intervenir la dimension VC *(h)*. Le principe de minimisation du risque structurel dans la théorie des machines à vecteurs de support découle du fait qu'il est démontré que la dimension VC associée à des fonctions de décision linéaire, vérifie l'inégalité suivante :

$$h < R^2 A^2 + 1 \tag{3.7}$$

Où R est le rayon de la plus petite sphère englobant les données d'apprentissage $x_1, x_2, .., x_l$ et A est un scalaire tel que $A \geq 2/M$.

Maximiser la limite inférieure de la marge *M* permet de minimiser la dimension VC de la machine d'apprentissage. Donc, nous cherchons à minimiser à la fois le nombre d'exemples d'apprentissage mal classifiés et la dimension VC en maximisant la marge *M*.
Un point intéressant est le fait que ce résultat est indépendant de la dimension des données de l'ensemble d'apprentissage. On peut donc obtenir une dimension VC contrôlable dans un espace à très haute dimension.

3.1.3. Hyperplan optimal

On appelle hyperplan optimal l'hyperplan séparateur qui est situé à la distance maximale des vecteurs *x* les plus proches parmi l'ensemble des exemples, cet hyperplan maximise la marge.
Nous cherchons à trouver la fonction de décision :

$$f(x) = \mathrm{sgn}(w.x + b) \quad w \in R^n, b \in R, x \in R^n \tag{3.8}$$

Si cette fonction existe, l'inégalité suivante est vérifiée :

$$y_i(w.x_i + b) \geq 1 \quad pour \quad i = 1,...,l \tag{3.9}$$

Le but de l'apprentissage est de déterminer $w \in R^n$ et $b \in R$ tel que le risque soit minimal. Il s'agit donc de minimiser la borne de l'inégalité constituée du risque empirique et du terme de confidence. Pour un classifieur linéaire et d'après (3.7) la dimension VC est bornée telle que $A \geq 2/M$. il est établi que $||w|| \leq A$. Donc, pour minimiser *h*, il suffit de minimiser $||w||$.
En résumé, maximiser la limite inférieure de la marge *M* permet de minimiser la dimension VC. Cela peut être formalisé par un problème d'optimisation quadratique comme suit [BUR 98] :

$$\begin{cases} Min_{w,b} \ \dfrac{1}{2}\|w\|^2 \\ sous\ les\ contraintes \quad y_i(w.x_i + b) \geq 1 \quad pour \quad i = 1...l \end{cases} \tag{3.10}$$

Dans cette formulation les variables à fixer sont les composantes du vecteur *w* et le biais *b*. Pour résoudre un problème d'optimisation sous contraintes, il suffit de chercher un point stationnaire Z_o du lagrangien $L(Z,\alpha)$ de la fonction g à optimiser et les fonctions C_i^g exprimant les contraintes [EVE 00].

$$L(Z,\alpha) = g(Z) + \sum_{i=1}^{l} \alpha_i C_i^g(Z) \tag{3.11}$$

Où les $\alpha(\alpha_1, \alpha_2, ..., \alpha_l)$ sont les contraintes appelés coefficients ou *Multiplicateurs de Lagrange.*

Avec des fonctions *g* et C_i^g *convexe*, il est toujours possible de trouver un point (Z_o,α^*) qui vérifie :

$$\min_z L(Z,\alpha^*) = L(Z_0,\alpha^*) = \max_{\alpha\geq 0} L(Z_0,\alpha) \tag{3.12}$$

L'introduction des multiplicateurs de Lagrange dans la fonction objectif (3.10) donne le *Lagrangien primal*[2] qui s'écrit :

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i (y_i((x_i.w)+b)-1) \tag{3.13}$$

Les variables α_i correspondent aux facteurs de Lagrange des contraintes définies dans (3.10). Le Lagrangien $L(w,b,\alpha)$ doit être minimal par rapport à (w,b) et maximal par rapport à $\alpha\geq 0$. Le point optimal vérifié[3] [BUR 98] :

$$\frac{\partial}{\partial b} L(w,b,\alpha) = 0 \quad \Leftrightarrow \quad \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{3.14}$$

$$\frac{\partial}{\partial w} L(w,b,\alpha) = 0 \quad \Leftrightarrow \quad w = \sum_{i=1}^{l} \alpha_i x_i y_i \tag{3.15}$$

Avec :

[2] Le primal est un problème initial
[3] les points qui minimisent ou maximisent une fonction dérivable annulent sa dérivée.

$$\alpha_i\left[y_i\left((x_i.w)+b\right)-1\right]= \quad 0 \quad i=1...l \tag{3.16}$$

En remplaçant w par son expression de l'équation (3.15) dans l'équation (3.13), et en prenant en compte la contrainte (3.16), la fonction objectif $L(w,b,\alpha)$ prend la forme d'un dual à maximiser en fonction de α_i seulement. Il s'écrit :

$$W(w)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\alpha_i\alpha_j y_i y_j (x_i.x_j) \tag{3.17}$$

sous les contraintes :

$$\begin{cases}\alpha_i \geq 0 \qquad i=1...l \\ \sum_{i=1}^{l}\alpha_i y_i = 0\end{cases} \tag{3.18}$$

En résolvant le problème d'optimisation dual, on obtient les coefficients α_i, $(i=1,...,l)$ permettant d'exprimer le vecteur w.

$$w=\sum_{i=1}^{l}\alpha_i y_i x_i \tag{3.19}$$

Les données d'apprentissage associées à des paramètres α_i non nuls sont appelés *Vecteurs de Support* [AYA 04].
Selon les contraintes (3.16) (les contraintes d'optimalité) on a :

$$\alpha_i\left[y_i\left((x_i.w)+b\right)-1\right]= \quad 0 \quad i=1...l$$

Les α_I associées aux vecteurs de support ne sont pas nuls donc l'équation :

$$\left[y_i\left((x_i.w)+b\right)-1\right]= \quad 0 \tag{3.20}$$

est vérifiée par ces points.

De ce fait, on peut facilement calculer le paramètre *b*. la fonction de décision finale du SVM est la suivante :

$$f(w) = sign\left(\sum_{i=1}^{l} \alpha_i y_i (x.x_i) + b\right) \tag{3.21}$$

3.2. Cas des données non séparable (Hyperplan à marge molle)

Le problème d'optimisation quadratique énoncé dans l'équation (3.17) a une solution uniquement dans le cas de données séparables [MUL 01]. Cependant, les données d'apprentissage peuvent être bruitées et non séparables (voir figure 3.4). L'hyperplan optimal est celui qui satisfait les conditions suivantes :

- La distance entre les vecteurs bien classés et l'hyperplan optimal doit être maximale ;
- La distance entre les vecteurs mal classés et l'hyperplan optimal doit être minimale.

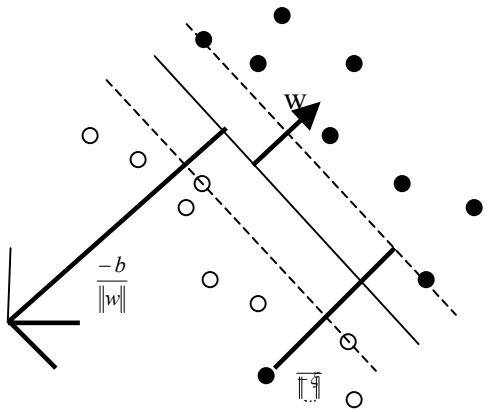


FIG 3.4 – *Hyperplan de séparation linéaire pour des données non séparables [BUR 98]*

Une technique qui consiste à accepter les erreurs d'apprentissage a été utilisée pour les données non séparables. Des variables ξ_i *(Slack variables)* de relaxation sont introduites pour relâcher les contraintes sur la marge [COR 95].
Ces variables transforment les contraintes (3.3) comme suit:

$$\begin{cases} (w.x_i + b) \geq +1 - \xi_i & pour \quad y_i = +1 \\ (w.x_i + b) \leq -1 + \xi_i & pour \quad y_i = -1 \\ \xi_i \geq 0 & pour \quad i = 1...l \end{cases} \tag{3.22}$$

Ce qui est équivalent à :

$$\begin{cases} y_i\,(w.x_i + b) \geq +1 - \xi_i & pour \quad i = 1...l \\ \xi_i \geq 0 & pour \quad i = 1...l \end{cases} \tag{3.23}$$

Dans ce cas, la fonction objectif à optimiser s'écrit :

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \tag{3.24}$$

Où *C* est un paramètre de régularisation qui permet de concéder moins d'importance aux erreurs.

Cortes et Vapnik [COR 95] démontrent que cela mène à un problème dual légèrement différent de celui des données séparables. Le Lagrangien dual à maximiser s'écrit :

$$W(w) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\alpha_i\alpha_j y_i y_j (x_i.x_j) \tag{3.25}$$

sous les contraintes :

$$\begin{cases} 0 \leq \alpha_i \leq C & pour \quad i = 1...l \\ \sum_{i=1}^{l}\alpha_i y_i = 0 \end{cases} \tag{3.26}$$

Ce problème est le même que celui du cas strictement séparables avec une contrainte supplémentaire qui consiste à borner les coefficients α_i. le calcul de *w* et *b* de la fonction

de décision finale du SVM reste le même que pour le cas des données séparables. Donc, le vecteur *w* est encore :

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i$$

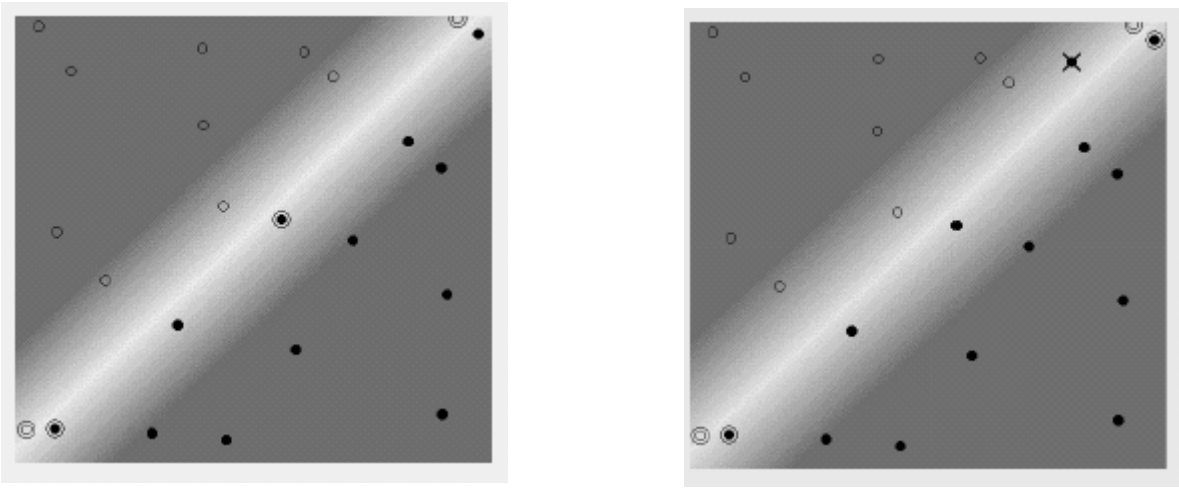


FIG 3.5 – *Un problème de classification binaire, à gauche les données sont séparables, par contre à droite les données ne le sont pas [BUR 98].*

3.3. Les conditions de Karuch-Kuhn-Tucker (KKT)

La plupart des méthodes d'optimisation sont basées sur des conditions d'optimalité du second degré, appelés les conditions de *Karush-Kuhn-Turcker (KKT*). Ces conditions sont nécessaires et parfois suffisantes pour qu'un ensemble de variables soit optimal pour un problème d'optimisation [BUR 98][MAH 03] . La résolution des problèmes d'optimisation quadratiques pour les SVM est basée sur les conditions de convergences de (KKT) qui établissent des critères nécessaires (et souvent suffisants) de convergence de la fonction objectif. Ces conditions sont particulièrement simples [BUR 98] :

$$\begin{cases} \alpha = 0 & \Rightarrow \quad y_i f(x_i) \geq 1 \quad et \quad \xi_i = 0 \\ 0 \leq \alpha_i \leq C & \Rightarrow \quad y_i f(x_i) = 1 \quad et \quad \xi_i = 0 \\ \alpha_i = C & \Rightarrow \quad y_i f(x_i) \leq 1 \quad et \quad \xi_i \geq 0 \end{cases} \tag{3.27}$$

Les équations (3.27) reflètent une propriété importante du SVM stipulant qu'une grande proportion des exemples d'apprentissage sont situés en dehors de la marge et ne sont pas retenus par le modèle. Par conséquent, leurs multiplicateurs α_i sont nuls.

Les conditions de KKT traduisent le fait que seulement les variables αi des points situés sur la frontière de la marge ($0 < \alpha i < C$) ou à l'intérieur de celle-ci ($\alpha_i = C$) sont non nulles. Ces points sont les vecteurs de support du classifieur.

Le SVM produit alors une solution clairsemée n'utilisant qu'un sous ensemble réduit des données d'apprentissage. Sans cette propriété, l'entraînement du SVM sur de grands ensembles de données ainsi que son stockage deviennent extrêmement difficiles [AYA 04].

4. Le SVM non linéaire

4.1. Espace augmenté (Feature Space)

Le classifieur à marge maximale que nous venons de présenter permet d'obtenir de bons résultats lorsque les données sont linéairement séparables. Naturellement, un grand nombre de jeux de données sont non linéairement séparables. Donc, choisir des frontières de décisions linéaires semble être un facteur limitant. Cependant, de tels modèles peuvent être considérablement enrichis en projetant les données dans un *Espace Augmenté (Feature Space),* éventuellement de plus grande dimension que l'espace des entrées, afin de rendre linéairement séparables le jeu de données. La dimension de l'espace augmenté est généralement très élevée. Cela ne pose pas de problème pour le SVM, vu que sa formulation duale fixe le nombre de variables à déterminer en fonction de la taille de l'ensemble d'apprentissage.

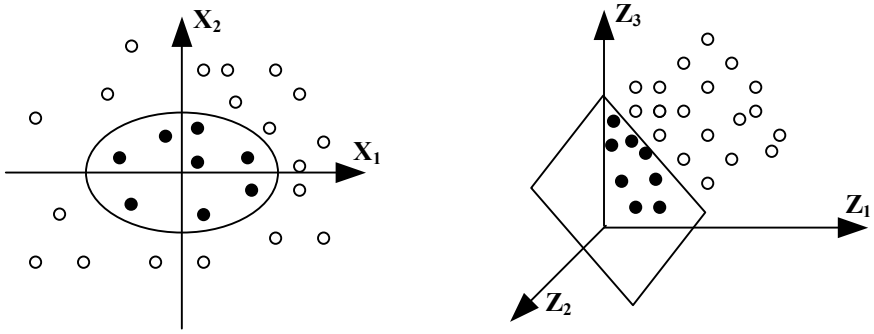


FIG 3.6 – *Illustration de l'effet du changement d'espace par une fonction noyau : Les données non linéaires séparables dans l'espace de départ X le sont à présent dans F [MUL 01].*

Soit l'observation $x = (x_1, x_2, \ldots, x_n) \in R^n$, et supposant que son information utile soit contenue dans l'ensemble des monômes d'ordre d des composants x_j de x, c'est-à-dire $x_{j1}, x_{j2}, \ldots, x_{jd}$ où $j_1, \ldots, j_d \in \{1, \ldots, N\}$. Dans ce cas, il est possible de considérer l'espace F des monômes d'ordre d comme espace de caractérisation (feature space). En reconnaissance d'images, ceci est équivalent à calculer des produits de valeurs de pixels [AYA 04].
Par exemple, dans l'espace d'entrée R^2, l'ensemble des monômes d'ordre 2 constitue un vecteur de caractérisation de dimension 3, comme montré ci-dessous :

$$\phi : \quad \Re^2 \rightarrow F = \Re^3 \tag{3.28}$$

Il associe pour chaque observation $x = (x_1, x_2)$, une image de la forme :

$$(x_1, x_2) \rightarrow (x_1^2, x_2^2, x_1 x_2) \tag{3.29}$$

Cette approche permet de considérer un degré de non linéarité allant jusqu'à :

$$N_F = \frac{(N + d - 1)!}{d!(N-1)!} \tag{3.30}$$

Où N_F représente la dimension de l'espace des monômes de degré d correspondant à l'espace d'entrée R.

Une image de 16 × 16 (pixels en reconnaissance des caractères), par exemple, produit 10^{10} monômes d'ordre 5. Le calcul dans cet espace peut s'avérer fastidieux et très coûteux en complexité de calcul et en espace mémoire. Cependant, il est possible de considérer un espace augmenté *F* de très haute dimension sans calculer explicitement les données dans cette espace en utilisant des noyaux non linéaires *(Kernel function)*. Ces derniers permettent de projeter les données de l'espace d'entrée R^n vers un espace augmenté et manipuler uniquement les produits scalaires des points. La construction de l'hyperplan optimal dans l'espace augmenté *F* et l'évaluation de la fonction de décision correspondante nécessitent seulement le calcul du produit scalaire $(\phi(x).\phi(y))$ et n'exigent jamais le calcul explicite de $\phi(x)$ [VAP 98].

4.2. Exemples de Kernels

Pour pouvoir représenter des produits scalaires de la forme $\phi(x).\phi(y)$ la notation suivante est utilisée :

$$K(x,y) = \phi(x).\phi(y) \tag{3.31}$$

qui décrit la valeur du produit scalaire dans l'espace augmenté *F*.

Ce calcul ne nécessite pas le calcul des images $\phi(x)$ *et* $\phi(y)$ explicitement. Cette propriété a été utilisée par Boser et al.pour étendre l'algorithme de l'hyperplan généralisé au SVM non linéaire [BOS 92].

Donc tout algorithme de classification linéaire pouvant se formaliser sous forme de produits peut être étendu à la classification non linéaire grâce à une fonction noyau choisie à priori [AYA 04].

Par exemple, la forme générique du noyau polynomial est :

$$K(x,y) = \left(a.x.y + b\right)^d \tag{3.32}$$

Prenons une instance simple de cette fonction :

$$K(x, y) = (x.y)^2 \tag{3.33}$$

Si nous considérons l'exemple du paragraphe précédent avec $N = d = 2$, nous aurons alors une transformation de la forme :

$$x\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} : C_2(x) : (x_1, x_2) \rightarrow (x_1^2, x_2^2, x_1 x_2)$$

dont le produit scalaire $(C_2(x).C_2(y))$ peut s'écrire :

$$(C_2(x).C_2(y)) = x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 x_2 y_1 y_2 = (x.y)^2$$

Cet exemple simple montre qu'un noyau quadratique produit un espace augmenté implicite dans R^3. Plus généralement, il est démontré qu'un noyau polynomial d'ordre d projette les données dans l'espace des monômes d'ordre d [AYA 04].
Une autre fonction ϕ possible pour ce même noyau aurait pu être :

$$\phi(x) = (x_1^2, x_2^2, \sqrt{2}.x_1 x_2) \tag{3.34}$$

On peut déduire de ce qui précède, que le mapping et l'espace augmenté que le kernel induit ne sont pas uniques.
Toute fonction symétrique n'est pas nécessairement une fonction noyau dans la mesure où elle peut ne pas correspondre à un produit scalaire dans un espace [COR 02]. Il faut pour cela qu'une certaine condition mathématique appelée condition de *Mercer* soit vérifiée. En pratique, quelques familles de fonctions paramétrables sont connues pour satisfaire à ces conditions.

Le tableau suivant montre quelques noyaux de *Mercer* classiques utilisés pour le SVM :

NOYAU	FORMULE
Linéaire	$K(x,y) = x.y$
Sigmoide	$K(x,y) = \tanh(ax.y + b)$
Polynomial	$K(x,y) = (ax.y + b)^d$
RBF	$K(x,y) = \exp(-\|x-y\|^2/\sigma^2)$

TAB. 3.1 – *Quelques noyaux de Mercer*

4.3. Condition de Mercer

Pour être sûr qu'une fonction symétrique *K(x,y)* admet un développement de la forme suivante :

$$K(x,y) = \sum_i \phi(x)_i \phi(y)_i \tag{3.35}$$

(K(x,y) décrit un produit scalaire) il est nécessaire que la condition suivante soit satisfaite :

$$\iint K(x,y)g(x)g(y)dxdy \geq 0 \tag{3.36}$$

pour toute fonction $g \neq 0$ avec :

$$\int g^2(z)dz \geq 0 \tag{3.37}$$

4.3. Frontière de décision non linéaire

Puisque la classe des fonctions linéaires n'est pas suffisamment discriminante, et pour construire les machines à vecteurs de support, il faut chercher un hyperplan dans l'espace augmenté *F*.

L'algorithme des SVM est uniquement basé sur l'utilisation des produits scalaires. Boser et al.[BOS 92] ont su produire des frontières de décision non linéaire avec le SVM (Figure3.7). L'idée est d'utiliser un noyau de *Mercer* qui permet de projeter les données dans un espace éventuellement plus grand dans lequel une séparation linéaire des classes est possible. Ceci permet d'obtenir des frontières de décision non linéaires dans l'espace initial, tout en appliquant l'algorithme initial de séparation linéaire dans l'espace augmenté

F. Il est alors important que la formulation du produit scalaire dans l'équation 3.17 reste intacte après l'introduction du noyau. Boser et al. démontrent qu'un noyau de Mercer ayant la forme (3.35)
ne change pas la nature de la fonction objectif à optimiser qui est toujours équivalente à un problème d'optimisation quadratique.
Le Lagrangien dual de la fonction objectif à maximiser est alors [VAP 98] :

$$W(w) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j k(x_i . x_j) \tag{3.38}$$

La fonction de décision s'écrit encore :

$$f(w) = sign\left(\sum_{i=1}^{l} \alpha_i y_i k(x . x_i) + b) \right) \tag{3.39}$$

Notons que *f(x)* est une combinaison linéaire de terme non linéaire $k(x, x_i)$ qui représente la similarité entre les images des points de x et x_i. Pour une observation de test *x* quelconque, la valeur de *f(x)* indique la classe d'appartenance inférée par le SVM.

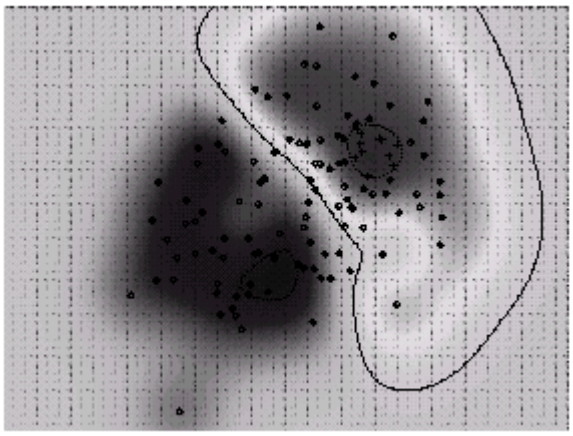


FIG 3.7 – *Frontière de décision non linéaire [AYA 04]*

4.4. Unicité et globalité de la solution

Afin de trouver les paramètres du SVM, il est nécessaire de résoudre le problème d'optimisation quadratique convexe donné par l'équation (3.36) dont la formulation matricielle est la suivante :

$$-\frac{1}{2}\alpha^T Q\alpha + 1^T\alpha \tag{3.40}$$

où Q est une matrice semi- définie positive dont les composantes $Q_{i,j} = y_i y_j k(x_i, x_j)$ et 1 est le vecteur unité de même taille que α. Comme la fonction objectif est convexe, tout *maximum local* est aussi un *maximum global*. Toutefois, il peut y avoir des solutions optimales différentes en terme de α_i donnant lieu à des performances différentes.

5. Algorithmes d'apprentissage du SVM

Il existe une grande variété de méthodes et de logiciels traitant de la résolution de problèmes quadratiques. Cependant, quelques méthodes seulement sont capables de traiter un grand nombre d'exemples souvent sous hypothèse que la matrice de *Graha. Schmidt Q* soit creuse [VAN 94]. Dans le cas contraire, l'apprentissage d'un SVM de quelques centaines d'exemples prendrait énormément de temps de calcul et beaucoup de ressources mémoire. Il est possible, de dériver des algorithmes qui exploitent la forme particulière de la fonction objectif duale du SVM. Dans cette section nous allons présenter trois approches différentes pour la résolution du problème quadratique du SVM.

5.1. Méthode de chunking

La résolution de la fonction objectif duale de l'équation (3.27) avec un très grand nombre d'exemples donne lieu à un vecteur α creux. Selon les données, plusieurs des paramètres α_i sont soit nuls ou égaux à C. s'il y a un moyen de savoir a priori lesquels α_i seront nuls, il est possible de déduire la taille de la matrice K sans altérer la valeur de la fonction objectif. Aussi, une solution α est valide si et seulement si elle respecte les conditions de KKT.

Vapnik [VAP 82] était le premier à décrire une méthode qui exploite cette propriété en prenant en compte seulement les α_i nuls ou ceux violant les conditions de *Karush Kunh Tucker*. La taille de ce sous ensemble dépend du nombre de vecteurs de support, de la taille des données et de la complexité du problème de classification. Cette méthode se comporte assez bien sur des problèmes de quelques centaines de vecteurs de support. Des tâches plus complexes requièrent un schéma de décomposition de l'objectif en sous problèmes plus facile à résoudre. Cette technique est décrite ci-dessous.

5.2. Méthode de décomposition successive

Cette méthode est similaire à celle du « *Chunking* » dans la mesure où elle considère aussi une succession de sous problèmes quadratiques à résoudre. La différence est que la taille des sous problèmes retenus est fixe. Cette méthode est basée sur la constatation qu'une succession de sous problèmes quadratiques ayant au moins un exemple qui ne vérifie pas les conditions de *KKT* converge toujours vers une solution optimale [OSU 97]. Osuna et al. suggèrent de conserver la taille du sous problème fixe et d'ajouter ou d'enlever un exemple à la fois. Ceci permet d'entraîner de gros ensembles de données. En pratique, cette stratégie peut converger lentement si diverses heuristiques ne sont pas prises en compte. En effet, il est possible d'adopter des stratégies sophistiquées afin d'inclure ou d'exclure quelques exemples de la fonction objectif. L'algorithme de SVM_{light} est une implémentation de cette méthode [JOA 99].

5.3. Méthode de minimisation séquentielle : SMO

La méthode d'optimisation par minimisation séquentielle *(SMO : Sequential Minimal Optimization)* proposée par Platt peut être perçue comme le cas extrême des méthodes de décomposition successive [PLA 99]. A chaque itération, elle résoud un problème quadratique de taille égale à deux. La résolution de ce dernier est analytique et donc nul besoin de recourir à un module d'optimisation quadratique. Encore faut-il choisir le bon couple de variables $(\alpha_{i.}, \alpha_j)$ à optimiser durant chaque itération. Les heuristiques utilisées sont basées sur les conditions de *KKT*. L'implémentation de cet algorithme est relativement simple et un pseudo code de la méthode est aussi fourni [PLA 99].

6. Systèmes de reconnaissance d'écriture à base de SVM

6.1. Système de Ayat, 2004

Le travail présenté dans [AYA 04] présente une nouvelle méthodologie de sélection de modèle automatique des machines à vecteurs de support pour la reconnaissance de chiffres manuscrits. L'approche permet d'optimiser les paramètres de noyaux et de réduire efficacement la complexité du classifieur en minimisant le nombre de vecteurs de support. Ceci s'accompagne d'une réduction de l'erreur de généralisation. L'algorithme proposé a été testé sur la base USPS (US Postal Service), et la base INDCENPARMI.

La base de données USPS est un benchmark bien connu au sein de la communauté de reconnaissance de formes. Cet ensemble contient 9298 images de chiffres manuscrits dont 7291 images de test. Ces images ont été saisies à partir d'images d'enveloppes collectées au centre CEDAR à Buffalo (Etats-Unis). Chaque image de chiffre est représentée par 16*16 pixels de niveau de gris allant de 0 à 255. La base de données de chiffres indiens de CENPARMI a été collectée à partir d'images de chèques saoudiens fournis par la compagnie "Al Rajhi Banking Corporation". La base de chiffres indiens a été constituée à partir des montants numériques et contient deux corpus de données. La taille du corpus d'apprentissage est 4682, celle de l'ensemble de test est 1939.

Pour décrire les images de chiffres indiens, Ayat a considéré un premier ensemble de caractéristiques structurelles calculé par rapport aux régions pouvant exister dans les images de chiffres. Les régions détectées sont : région en vallée, région en montagne, région en boucle. Ces primitives structurelles sont combinées avec des primitives statistiques. Les primitives statistiques considérées sont basées sur la technique de zonage et la distribution des directions de Freeman et les courbure des pixels composant les contours du tracé. Le taux de reconnaissance obtenu est 85%.

Le taux de reconnaissance obtenu sur la base USPS après l'entraînement du système avec les valeurs des hyper-paramètres est 95,8%, 59,6% et 94,7% respectivement avec les noyaux KMOD(noyau proposé par Ayat), RBF et polynomial (d=2). Le taux de reconnaissance est améliorer 0,2% par rapport à la sélection manuelle.

6.2. Système de Bellili, 2001

Dans [BEL 01], les auteurs proposent un système de reconnaissance de chiffres manuscrits par combinaison hybride de réseau de neurones de type MLP et de machines à vecteurs de support. Les SVM sont utilisés pour améliorer significativement les performances de reconnaissance d'un réseau MLP dans le voisinage des hyperplans de séparation entre chaque paire de classes numériques, dans l'espace augmenté des formes à reconnaître. Cette architecture de combinaison est fondée sur la constatation que les deux sorties maximales de la couche de sortie du MLP contiennent presque systématiquement la bonne classe de la forme à classifier et que certaines paires de classes constituent la majorité des confusions générées par le MLP (le 3 et 9 par exemple). Des SVM locales spécialisés sont introduit pour détecter la bonne classe parmi les deux meilleures hypothèses de classification fournies par le réseau MLP. Le système hybride MLP-SVM réalise un taux de reconnaissance de 98,01% sur des chiffres manuscrits issus de codes postaux.

6.3 Système de Bahlman, 2002

Le travail présenté dans [BAH 02] décrit une nouvelle approche de classification pour la reconnaissance de l'écriture manuscrite "en ligne" des caractères latins. L'approche proposée combine les DTW *(Dynamic Time Warping)* et les SVM pour établir une nouvelle fonction noyau appelée : *"Gaussian Dynamic Warping Kernel"*. Le système proposé a été testé sur la base "UNIPEN" de caractères manuscrits latins en utilisant la méthode de minimisation séquentielle (SMO). D'après les auteurs, cette approche a donné des résultats meilleurs que ceux obtenus par les HMM.

7. Conclusion

Les SVM sont des nouvelles méthodes d'apprentissage statistiques proposées par Vapnik en 1995. Le succès de cette méthode est justifié par les solides bases théoriques qui la soutiennent. La plupart des techniques de machine learning possèdent un grand nombre de paramètres d'apprentissage à fixer par l'utilisateur (structure d'un réseau de neurones, coefficient de mise à jour du gradient…). De plus, le nombre de paramètres à calculer par l'algorithme d'apprentissage est en relation linéaire, voire exponentielle, avec la dimension

de l'espace d'entrée. La formulation élégante des SVM laisse très peu de place aux paramètres et leur nombre est linéaire en fonction de l'ensemble d'apprentissage. Le SVM est donc une méthode de classification particulièrement bien adaptée pour traiter des données de très haute dimension telles que les textes et les images. Dans la section 3 du chapitre suivant, nous définissons les paramètres fixés par l'utilisateur pour l'algorithme d'apprentissage des SVM.

Chapitre 4

Un Système à Base des SVM pour la Reconnaissance des Caractères Arabes

Dans le présent chapitre, nous proposons un système de reconnaissance de caractères arabes manuscrits en utilisant des classifieurs statistiques de type SMV, dont les fonctions noyaux et leurs paramètres sont déterminés de façon expérimentale

1. Introduction

Le domaine de recherche devient très actif grâce aux nombreuses applications pratiques qui intègrent des systèmes de reconnaissance de l'écrit (lectures d'adresses postales, traitement de chèque,…etc.) [FIL 98, AYA 00, ALO 03].

Il est d'usage de considérer sur ce thème de recherche les travaux portant sur les caractères qui sont déjà isolés (comme les caractères imprimés ou les caractères manuscrits issus de documents pré-casés par exemple) de travaux concernant les caractères issus d'un processus de segmentation automatique d'un mot ou d'une chaîne de caractères numériques (chiffres). Si dans le cas des caractères imprimés, on considère que les difficultés majeures ont été surmontées, il en va autrement en ce qui concerne les caractères manuscrits et surtout ceux issus d'un processus de segmentation. Toutes ces

recherches ont permis l'accumulation de nombreux et divers algorithmes de classification qui utilisent soit la représentation pure -en pixels- du caractère, soit une représentation vectorielle des caractéristiques (primitives).

Souvent, les erreurs de classification sont très coûteuses à détecter et à corriger. Deux situations sont à l'origine d'une confiance faible dans la décision de classification fournie par un système statistique à apprentissage supervisé :

1. les formes des caractères à classifier peuvent être ambiguës ;
2. ou elles peuvent n'avoir aucun lien avec les données d'apprentissage utilisées pour entraîner et optimiser le classifieur.

En 1979, V.Vapnik a mis au point le principe de la minimisation du risque structurel qui évite le sur-apprentissage des données, contrairement au risque empirique qui représente une estimation assez optimiste du risque réel dont la minimisation ne garantit pas la convergence vers une solution acceptable. Ce principe constitue un point essentiel dans la théorie des machines à vecteurs de support (SVM) (cahpitre2).

Les SVM ont été utilisés avec succès dans divers domaines d'applications (chapitre 3), mais, à notre connaissance, ils n'ont jamais été utilisés pour la reconnaissance de l'écriture arabe. De ce fait, le présent travail porte sur une contribution au domaine de la reconnaissance de l'écriture arabe manuscrite, plus précisément, la reconnaissance des caractères arabes manuscrits par l'utilisation des machines à vecteurs de support.

L'efficacité de cette méthode réside dans sa capacité à générer un hyperplan optimal de séparation entre les exemples de deux classes différentes. Cet hyperplan définit la marge de séparation la plus large entre les exemples de ces classes.

Dans ce qui suit, nous décrivons la méthodologie développée pour la conception du système proposé. Le schéma de combinaison utilisé dans notre système est un contre tous. Nous construisons autant de classifieurs SVM que de classes. Chaque SVM construit un hyperplan de séparation à marge optimal entre la classe i et les classes restantes.

2. Architecture du système proposé

L'objectif que nous nous sommes assignés s'articule autour du développement d'un système pour la reconnaissance des caractères arabes manuscrits omni-scripteurs. La figure 4.6 montre la méthodologie adoptée pour la reconnaissance.

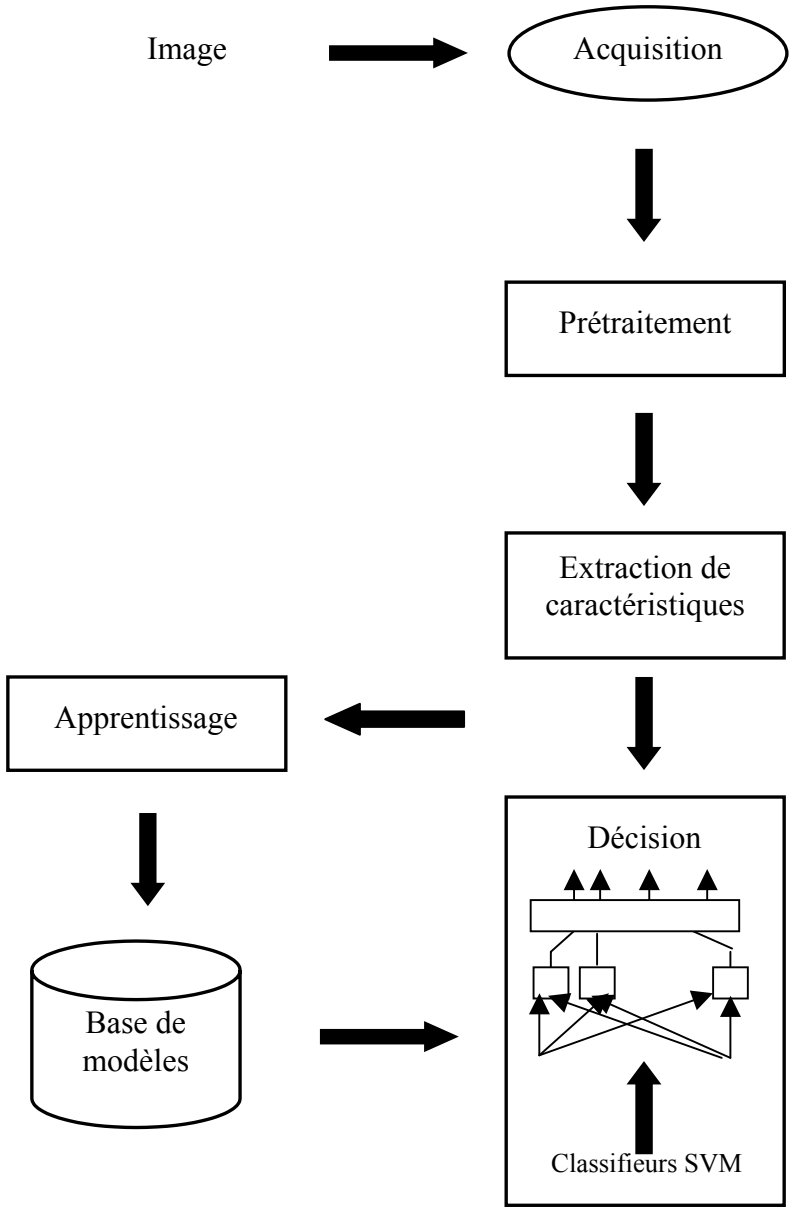


FIG. 4.1 – *Architecture du système proposé*

L'analyse du caractère à reconnaître passe par les étapes de prétraitement, extraction de caractéristique et apprentissage. L'étape de classification proprement dite (décision) déterminera la classe d'appartenance du caractère en entrée.

2.1. Prétraitement

Le rôle du prétraitement est de préparer l'image du caractère aux traitements ultérieurs. Il s'agit essentiellement de réduire le bruit superposé aux données et ne garder que l'information significative de la forme représentée. Dans notre système, le prétraitement inclut : la binarisation, le lissage, la normalisation et l'extraction du contour.

2.1.1. Binarisation

Initialement, les images des caractères à reconnaître sont en niveaux de gris, elles sont obtenues avec un scanner, codées au format BMP de Windows avec une résolution de 300 dpi (Dot par Inch) et une échelle de 256 niveaux de gris.
La binarisation transforme cette image à plusieurs niveaux de gris en une image bimodale (contenant seulement du noir et du blanc). Le principal problème pour cette étape réside dans le choix d'un seuil adéquat.
Le seuil de binarisation est déterminé à partir de l'histogramme de niveaux de gris de tous les pixels de l'image du caractère.
Le seuil choisi est calculé comme suit :

$$H = \sum_{i=0}^{255} \frac{H(i).n_i}{H(i)}$$

Où H(G) représente le niveau de gris, et n_i est le nombre de pixels associé à ce niveau. Tout pixel ayant un niveau de gris inférieur à ce seuil sera considéré comme un point noir et sera représenté par la valeur « 1 », les autres pixels ayant une valeur supérieure au seuil seront considérés comme des pixels blancs (appartenant au fond).

Le pseudo code de l'algorithme utilisé est le suivant :

Algorithme de Binarisation

Entrée : Image I en niveaux de gris sous format BMP

Sortie: Image I' bimodale (1 et 0)

Début

1. H = Histogramme de niveau de gris de I;

$$\overline{X_{Histo}} = \sum_{i=0}^{255} \frac{H(i).n_i}{H(i)}$$

2. Pour chaque pixel P de I faire

   Si P ≤ $\overline{X_{Histo}}$ alors P := 0 // rendre le pixel noir//

   Sinon P := 1 // rendre le pixel blanc//

   Fin Pour

Fin.

FIG. 4.2 –*l'image du caractère Tad en entrée*

FIG. 4.3 –*l'image du caractère Tad binarisée*

2.1.2. Lissage

Durant le processus d'acquisition et binarisation, quelques faux pixels viennent s'ajouter à l'image du caractère. Ces pixels constituent un bruit qui altèrent la qualité de l'image en créant des irrégularités sur le tracé du caractère.

Pour pallier à ce bruit, nous procédons à un lissage par nettoyage et bouchage, pour simplifier les traitements ultérieurs.

A cet effet, nous utilisons un algorithme adapté de [MAH 94]. Cet algorithme réduit le bruit d'une image binaire en éliminant les pixels isolés d'une part et en bouchant les trous vides d'autres part.

Cette technique simple et efficace est basée sur une décision statistique. En donnant une image binaire d'un caractère arabe, l'algorithme modifie chaque pixel selon sa valeur initiale et celles des pixels voisins (N-voisinages[4]).

[4] P_1, P_3, P_5, P_7 sont les voisins directs (d-voisins) de P_0
P_2, P_4, P_6, P_8 sont des voisins indirects (I-voisins) de P_0

P_4	P_3	P_2
P_5	P_0	P_1
P_6	P_7	P_8

FIG. 4.4 – *Le pixel courant P_0 et ses voisinages*

Algorithme de Lissage

Entrée : Image binaire I

Sortie : Image lissée I'

Début

// la règle de décision est la suivante//

Si $P_0 = 0$ alors

$$p'_{0=}\begin{cases} 0 & si \quad \sum_{i=1}^{8} P_i < T \\ 1 & \text{sinon} \end{cases}$$

Sinon

$$p'_{0=}\begin{cases} 1 & si \quad P_i + P_{i+1} = 2 \ pour\ au\ moins\ un\ i = 1,...,8 \\ 0 & \text{sinon} \end{cases}$$

Fin

P_o est la valeur du pixel courant, P_o' est la nouvelle valeur du pixel et T est un seuil choisi. Des expérimentations ont montré que la valeur 5 pour le seuil T donne de bons résultats.

FIG. 4.5 – *Image du caractère Tad lissée*

2.1.3. La normalisation

Puisque la taille des caractères est fortement variable, sa normalisation est souvent utilisée pour échelonner les caractères à une taille fixe. L'acquisition des images se à une dimension variable mais le processus de normalisation permet de rendre chaque image à une taille de 32*32 pixels.

2.1.4. Extraction du contour

Le contour est fortement sollicité dans la phase d'extraction de caractéristiques qui se base sur le suivi de contours extraits. L'opération de traçage du contour sert à décrire l'image du caractère en termes de codage de Freeman, qui représente tous les contours de l'image et sa topologie.

La chaîne de Freeman est la méthode la plus utilisée de description des contours dans les images. C'est une technique de représentation des directions de contour (on code la direction le long du contour dans un repère absolu lors du parcours du contour à partir d'une origine donnée). Les directions peuvent se présenter en 4 connexités (codage sur deux bits) ou en 8 connexités (codage sur 3 bits). Les codes des contours sont donnés par la figure suivante.

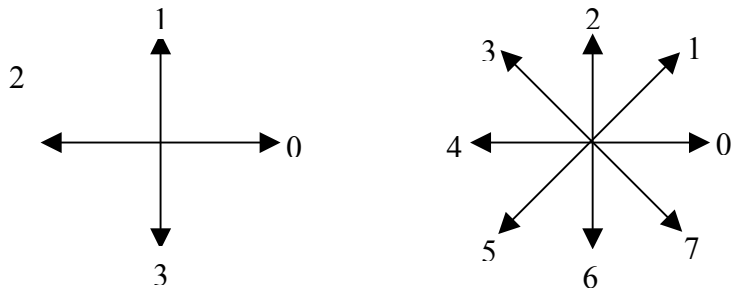
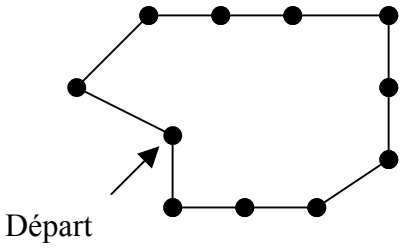


FIG. 4.6 – *Le code de Freeman en 4-connexités (à gauche) et en 8 connexités (à droite)*



Code 31000665442

FIG. 4.7 – *Codage à l'aide du code de Freeman*

L'étiquetage des contours permet d'attribuer à chaque contour de l'image une étiquette (couleur), permettant ainsi de faire sortir les différents types de portion connexe ie, les tracés principaux, les occlusions, les tracés secondaires (Hamza, les points diacritiques) existant dans l'image du caractère.

Chaque contour est codé en spécifiant un point de départ suivi par une chaîne ou une séquence de codes de la chaîne de Freeman, d'autres informations définissent le type de contour, et ses propriétés topologiques et géométriques. Suivant ce codage, chaque image de caractère est représentée sous forme d'une liste de contours. Les types de contours distingués sont les contours externes des tracés principaux des caractères, les contours internes (occlusions), et les contours externes correspondant aux tracés secondaires. Le début du contour spécifie le pixel (point) à partir duquel le premier code, dans sa chaîne de Freeman, commence.

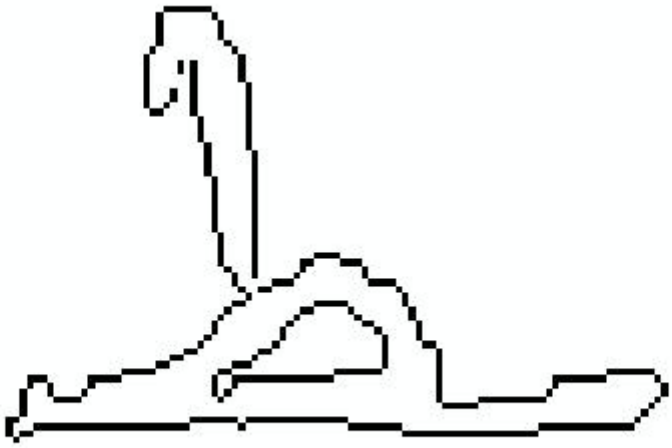


FIG. 4.8 – *Contour du caractère Tad*

2.2. Extraction de caractéristiques

Les systèmes de reconnaissance de l'écriture nécessitent généralement deux étapes importantes : une étape d'extraction de caractéristiques et une étape de classification dans laquelle des règles de décision, pour séparer les classes, sont définies.

Dans la littérature, plusieurs travaux ont porté sur l'élaboration de nouvelles caractéristiques à même de procurer un pouvoir discriminant inter-classes accru tout en minimisant la variabilité intra-classes.

Un des problèmes fondamentaux de la reconnaissance des formes est de déterminer quelles caractéristiques à employer pour avoir le bon résultat de la classification. L'objectif de l'extraction des caractéristiques est d'identifier les caractéristiques qui sont importantes dans la discrimination de classes de formes. L'extraction des primitives descriptives de l'image est décisive en reconnaissance. Un compromis doit être respecté lors de l'extraction des caractéristiques : l'extracteur de caractéristiques doit fournir des primitives uniformes pour différents types d'écriture, en reflétant avec précision tous les détails nécessaires au processus de reconnaissance [SNO 02]. Il doit, en outre, être peu gourmand en temps d'exécution et en espace mémoire [BOU 97]. Divers types de méthodes d'extraction des caractéristiques sont connus en littérature [CAE 93, CHH 93]. Ils se basent sur trois descriptions principales : la représentation Bitmap de l'image [KIM 99], la squeletisation [HAM 93], et la fonction du contours [CAE 97, MAH 94, MAD 97].

Durant notre étude bibliographique sur les caractéristiques structurelles globales de l'écriture arabe, les primitives qui ont été prises en considérations sont les caractéristiques de haut niveau issues de la perception humaine [SOU 06, FAH 05, MIL 98, AMI 98, CHE 98, ALB 95]. Ces caractéristiques seront détaillées dans les prochaines sections.

Le problème avec les caractéristiques structurelles globales, est que les mots qui ne contiennent pas ces caractéristiques risquent de ne pas être traités par le système de reconnaissance. La solution apportée par les chercheurs est l'ajout de caractéristiques locales secondaires telles que : les vallées [MIL 98, COT 98], le type de courbure du contour ou du squelette [MAH 94], et le nombre de concavités [MAH 94]. Le problème avec ces caractéristiques secondaires est la difficulté de leurs extractions, particulièrement pour l'écriture manuscrite.

Puisque le SVM est un classifieur statistique, nous utilisons, dans notre système, des caractéristiques statistiques dérivées à partir de la distribution des pixels comme : les caractéristiques de projections, de transitions et les coefficients de Fourier.

Cependant, les caractéristiques statistiques ne sont pas suffisantes pour représenter les motifs géométriques des caractères. Aussi, l'aspect morphologique constitue un facteur très important dans l'interprétation humaine des images des caractères. Donc, nous avons décidé de considérer un deuxième jeu de données basé sur des caractéristiques structurelles de l'alphabet arabe.

La combinaison de ces deux catégories donne une meilleure et complète description du caractère.

2.2.1. Les caractéristiques statistiques

2.2.1.1. Les caractéristiques de projections

Les primitives de projections sont bien utilisées dans la reconnaissance des caractères. Il existe différentes formes de projections incluant la projection verticale et horizontale. Ces projections sont dérivées à partir des histogrammes représentant la projection des pixels noirs de l'image du caractère. Elles sont extraites à partir des images normalisées. Dans notre cas, nous avons pris des images de tailles 32*32 pour une meilleure représentation

[MEH 05]. Les figures 4.9 et 4.10 illustrent les histogrammes de projections sur l'axe X et l'axe Y du caractère « Tad »

La projection horizontale est définie par :

$$H(i) = \sum_{j} P(i, j) \tag{4.1}$$

et la projection verticale est définie par :

$$V(i) = \sum_{i} P(i, j) \tag{4.2}$$

où

$$P(i, j) = \begin{cases} 1 \quad si \quad pixel \quad noir \\ 1 \quad si \quad pixel \quad blanc \end{cases} \tag{4.3}$$

En plus de la projection horizontale et verticale, nous effectuons une projection de la diagonale et diagonale oblique.

Le résultat des projections sera un vecteur de taille égale à 190 (32 horizontales + 32 verticales + 63 diagonales + 63 diagonales obliques). Ce vecteur sera utilisé lors de la classification. Ces projections sont faites pour toutes les images de la base d'apprentissage et de test.

FIG. 4.9 – *Projection horizontale du caractère Tad*

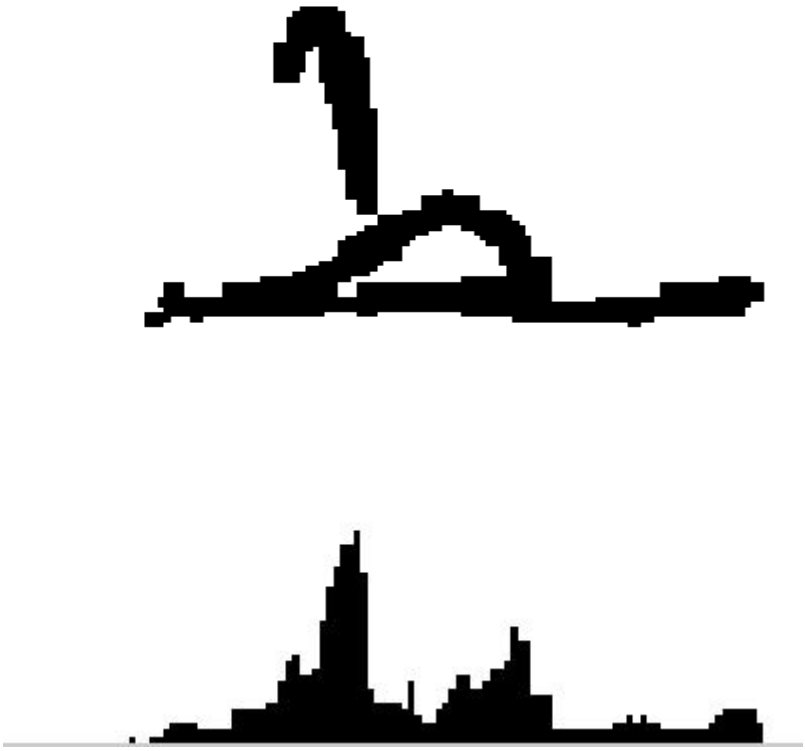


FIG. 4.10 – *Projection verticale du caractère Tad*

2.2.1.2. Les caractéristiques de transitions

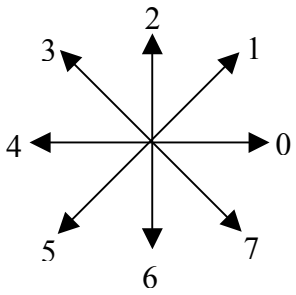
C'est le passage d'une zone libre à une zone occupée par les pixels noirs. Dans notre cas, on calcule le nombre de passage de 0 à 1 de chaque ligne, colonne, diagonal et diagonal oblique [MEH 05].

2.2.1.3. Suivi du contour

L'opération de suivi de contour permet de déterminer :

- La chaîne de code de Freeman (code de Freeman) de chaque tracé ;
- Le nombre de tracé ;
- Le nombre P d'éléments dans la chaîne de codes, eg. le périmètre du contour ;
- La surface A du contour, eg. nombre de pixels ;
- Les cordonnées (X_{min}, Y_{min} , X_{max} , Y_{max}) du rectangle (Bounding Box) du contour. Ce rectangle est défini comme le plus petit cadre emboîtant le contour ;
- Le calcul des coefficients de Fourier est effectuée sur la base des chaînes de code.

L'algorithme utilisé pour le traçage du contour est basé sur celui de [PAV 81]. Il emploie la technique de *Left-Most-Looking (LML),* cette technique peut être décrite comme étant un observateur qui traverse l'ensemble des pixels de l'objet (pixel égal à 1) et sélectionne le pixel disponible le plus à gauche selon la direction d'entrée dans le pixel courant. Le point de départ peut être trouvé en parcourant l'image du caractère de droite à gauche et de haut en bat.



0 3,2,1,0,7,6,5,4	4 7,6,5,4,3,2,1,0
1 4,3,2,1,0,7,6,5	5 0,7,6,5,4,3,2,1
2 5,4,3,2,1,0,7,6	6 1,0,7,6,5,4,3,2
3 6,5,4,3,2,1,0,7	7 2,1,0,7,6,5,4,3

FIG. 4.11 – *Les séquences du mouvement dans le suivi du contour*

2.2.1.4. Les descripteurs de Fourier

La reconnaissance de l'écriture par transformée de Fourier est opérée sur les cordonnées des pixels du contour du caractère. Le contour d'un caractère est décrit par le code de Freeman sur lequel opère le calcul des descripteurs de Fourier. La figure 4.12 présente les étapes de calcul des coefficients de Fourier.

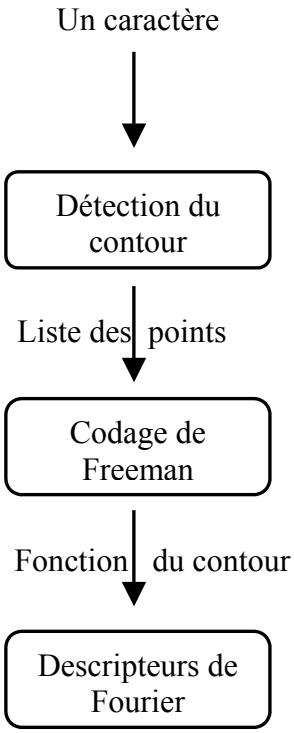


FIG. 4.12 – *Calcul de descripteurs de Fourier*

Le contour fermé d'un caractère (Boundary) peut être représenté par une fonction périodique de deux dimensions *x* et *y* sur laquelle peut être calculée une transformée de Fourier.

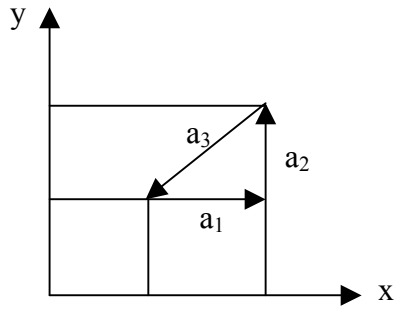
Le contour est représenté par des séries discrètes : x(m) qui représente les cordonnées des points x et y(m) qui représente les cordonnées des points y (ou m=0.1,…,L-1).

Dans ce qui suit, les formules discrètes pour calculer les descripteurs de Fourier du contour du caractère sont présentées.

Soit Δt_i la longueur d'une portion du contour à partir d'un point de départ, $\Delta t_i = 1$ si le code est pair, et $\sqrt{2}$ s'il est impair. Dans le domaine discret, les distances parcourues jusqu'àu point q sont décrites par x_q (projection du contour sur l'axe des x) et y_q (projection du contour sur l'axe des y) avec :

$$x_q = \sum_{i=1}^{q} \Delta x_i \qquad et \qquad y_q = \sum_{i=1}^{q} \Delta y_i \tag{4.4}$$

Δx_i et Δy_i représente les changements dans les valeurs des cordonnées x et y lorsque les points du contour sont traversés. Δx_i et Δy_i peuvent être 1, 0 ou -1 suivant la valeur et la direction du code de Freeman qui relie deux points du contour. La figure 4.13 présente ce codage.



$a_{1x}=1$, $a_{1y}=0$; $a_{2x}=0$, $a_{2y}=1$; $a_{3x}=-1$, $a_{3y}=-1$

FIG. 4.13 – *Projection des codes de Freeman sur l'axe des X et l'axe des Y*

La longueur cumulative du contour à partir d'un point de départ est donnée par :

$$t_q = \sum_{i=1}^{q} \Delta t_i \tag{4.5}$$

Soit :

$$a_{x,n} = \frac{T}{2\pi^2 n^2} \sum_{q=1}^{k} \frac{x_q}{t_q} \left(\cos\frac{2\pi n}{T} t_q - \cos\frac{2\pi n}{T} t_{q-1}\right) \tag{4.6}$$

$$b_{x,n} = \frac{T}{2\pi^2 n^2} \sum_{q=1}^{k} \frac{x_q}{t_q} \left(\cos\frac{2\pi n}{T} t_q - \cos\frac{2\pi n}{T} t_{q-1}\right) \tag{4.7}$$

Les descripteurs de Fourier sont calculés comme suit :

$$F_n[n] = \sqrt{a_{x,n}^2 + a_{y,n}^2 + b_{x,n}^2 + b_{y,n}^2} \tag{4.8}$$

Un nombre réduit de descripteurs de Fourier normalisés suffit pour reconnaître les objets. Cependant, certains caractères arabes ont des formes très proches. Les recherches ont montré que les dix premiers descripteurs de Fourier donnent de bons résultats dans le cas de la reconnaissance des caractères arabes. Les coefficients de Fourier varient suivant le point de départ de parcours du contour. Une étape de normalisation est nécessaire à ce stade.

Pour obtenir des coefficients de Fourier normalisés, le point de départ de chaque caractère doit être normalisé. Dans notre système, la normalisation est assurée en adoptant une séquence de parcours fixe pour les caractères. Chaque caractère est parcouru de droite à gauche et de haut en bas. Le premier pixel non trouvé est pris comme un point de départ. De plus, pour avoir un système invariant au changement d'échelle, les coefficients de Fourier sont divisés par le plus grand coefficient (e.g. le premier). Les coefficients de Fourier obtenus sont invariants à la rotation, translation et changements d'échelle.

2.2.2. Les caractéristiques structurelles

Les primitives visuelles de l'écriture arabe font partie des descripteurs structurels qui sont reliés à la forme de l'écriture. Divers descripteurs peuvent être extraits à partir de l'écriture dont certains sont spécifiques à l'alphabet arabe. Ces primitives semblent être bien adaptées pour nombreux de classifieurs., les caractéristiques structurelles les plus utilisées dans le cas de l'écriture arabe sont :

- **les ascendants (H) :** connues sous le non de « *hampes H* », et correspondant aux extensions hautes pouvant exister dans les caractères ;
- **les descendants (J) :** connues sous le nom « *jambages J* », et correspondant aux extensions basses pouvant exister dans les caractères ;
- **les diacritiques (P : haut, Q : bas) :** qui correspondent aux parties secondaires des caractères;
- **les boucles (B) :** correspondant aux occlusions pouvant exister dans les caractères.

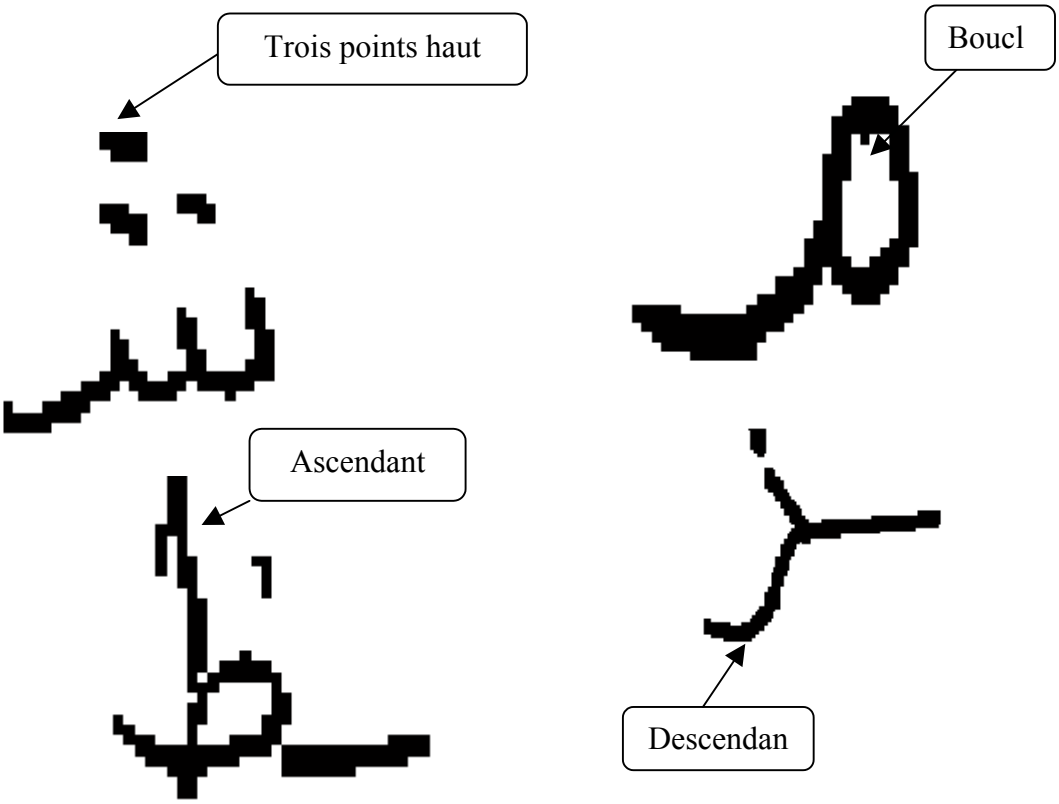


FIG. 4.14 – *Caractéristiques structurelles de quelques caractères*

Le tableau 4.1 classe les lettres qui n'ont ni hampes ni jambages ni boucles mais des points diacritiques, ou des caractères sans ces quatre caractéristiques **(R)**.

Lettres Arabes	Début	Milieu	Fin	Isolé
Ba	: Q	: Q	: Q	: Q
Ta	: P	: P	: P	: P
Tha	: P	: P	: P	: P
Del			: R	: R
Dhel			: P	: P

TAB. 4.1 : *Caractères classifiées par points diacritiques*

Dans le tableau 4.2, la classification est effectuée en tenant compte de la caractéristique jambage, nous remarquons que cette caractéristique ne peut être que dans les positions finale ou isolée.

Lettres Arabes	Début	Milieu	Fin	Isolé
Noun	: P	: P	: P, J	: P, J
Yae	: Q	: Q	: Q, J	: Q, J
Jim	: Q	: Q	: Q, J	: Q, J
Hae	: R	: R	: J	: J
Khae	: P	: P	: P, J	: P, J
Ra			: J	: J
Zain			: P, J	: P, J
Waw			: B, J	: B, J
Ssin	: R	: R	: J	: J
Chin	: P	: P	: P, J	: P, J

TAB. 4.2: *Caractères classifiées par Jambages*

Le tableau 4.3 présente une classification par présence de boucle.

Lettres Arabes	Début	Milieu	Fin	Isolé
Sad	: B	: B	:B, J	: B, J
Dhad	: B, P	: B, P	: B, P, J	: B, P, J
Tta	: B, H	: B, H	: B, H	: B, H
Dha	: B, P, H	: B, P, H	: B, P, H	: B, P, H
Ain	: R	: B	: B, J	: J
Ghain	: P	: B, P	: B, P, J	: P, J
Fa	: B, P	: B, P	: B, P	: B, P
Quaf	: B, P	: B, P	:J,B, P, J	:J, B, P, J
Mim	: B	: B	: B, J	: B, J
Ha	: B	: B	: B	: B
Ta marbouta			: B, P	: B, P

TAB. 4.3: *Caractères classifiées par boucles*

De cette analyse structurelle, nous remarquons que :

- la description structurelle d'une lettre varie suivant sa position dans le mot ;
- pour certaines lettres, des caractéristiques visuelles sont éliminées, d'autres sont ajoutées et d'autres sont totalement modifiées en passant d'une position à une autre ;
- différentes lettres peuvent avoir une description structurelle identique ;
- certaines combinaisons de caractéristiques ne sont jamais possibles ;
- les extensions hautes et basses des caractères sont toujours en dehors de la zone médiane, et pour les détecter, il faut toujours commencer par la détection de cette zone. La détermination des lignes qui délimitent cette zone n'est pas évidente dans le cas des caractères, surtout pour certains caractères comme : ح ,ر.

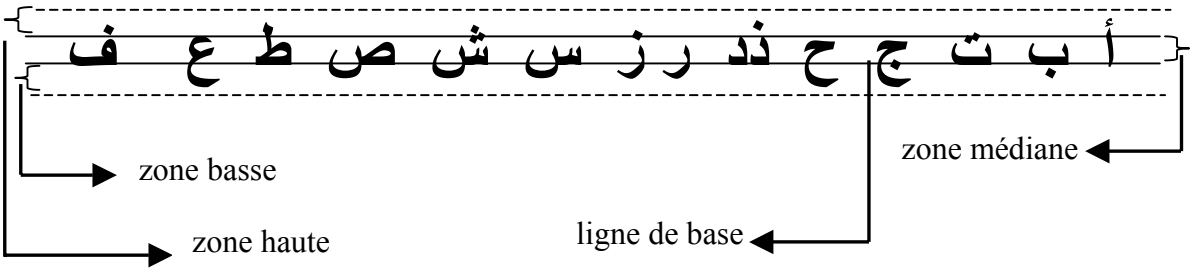


FIG. 4.15 – *Positionnement des caractères arabes isolés par rapport à la ligne principale d'écriture*

De ce fait, pour représenter nos images de caractères, les caractéristiques structurelles retenues sont : les boucles, les points diacritiques [MEH a06]

2.2.2.1. Extraction des diacritiques

L'écriture arabe est riche en points diacritiques. L'écriture de ces points (simple ou multiple) est généralement respectée par les scripteurs, car ces points permettent de distinguer entre les caractères ayant le même corps principal. Les points diacritiques sont en dehors de la zone d'information principale (partie primaire du caractère), ils constituent les parties secondaires des caractères. Les points diacritiques sont de taille assez faible, ce

que les rend sensible aux bruits d'acquisition. Un point simple est défini comme étant la plus petite entité connexe qui peut être considérée comme information significative et non pas comme bruit. Les points multiples, par contre, sont des formes plus complexes, car ils sont un regroupement des points d'un même caractère. Ce regroupement est dû essentiellement au style d'écriture. Même si ces derniers ont des formes plus simples dans le cas de l'imprimé, ils restent parfois indéchiffrables et inséparables dans le cas du manuscrit. Les points multiples sont de deux types : les points doubles et les points triples.

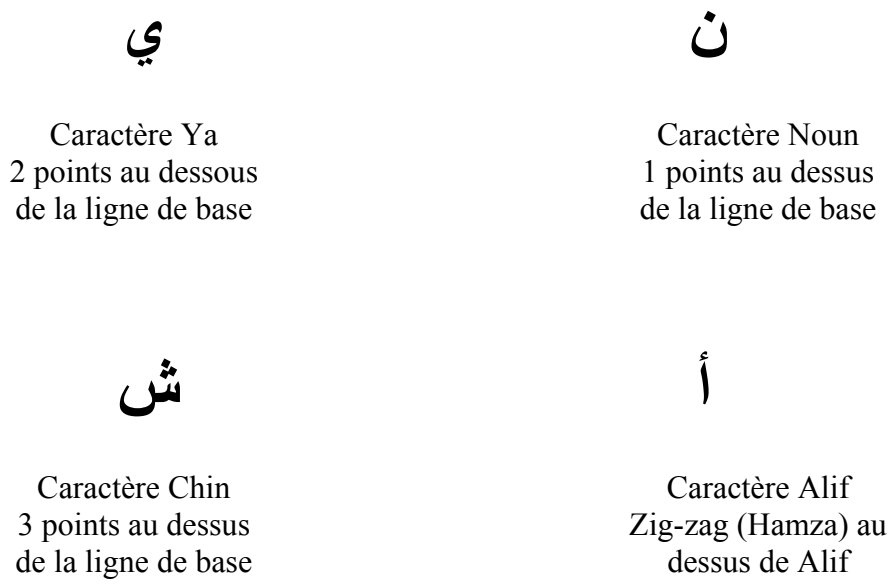


FIG. 4.16 – *Exemple des points diacritiques avec leurs positions*

Pour l'extraction des diacritiques, nous avons utilisé des heuristiques basées sur l'épaisseur du trait et proposées par Ameur et all [AME 94]. L'algorithme utilisé est présenté dans la figure 4.17, sachant que S est l'épaisseur du trait, Xmin,Xmax,Ymin,Ymax sont les cordonnées délimitant la composante considérée. La position des diacritiques est déterminée par rapport à la ligne de base.

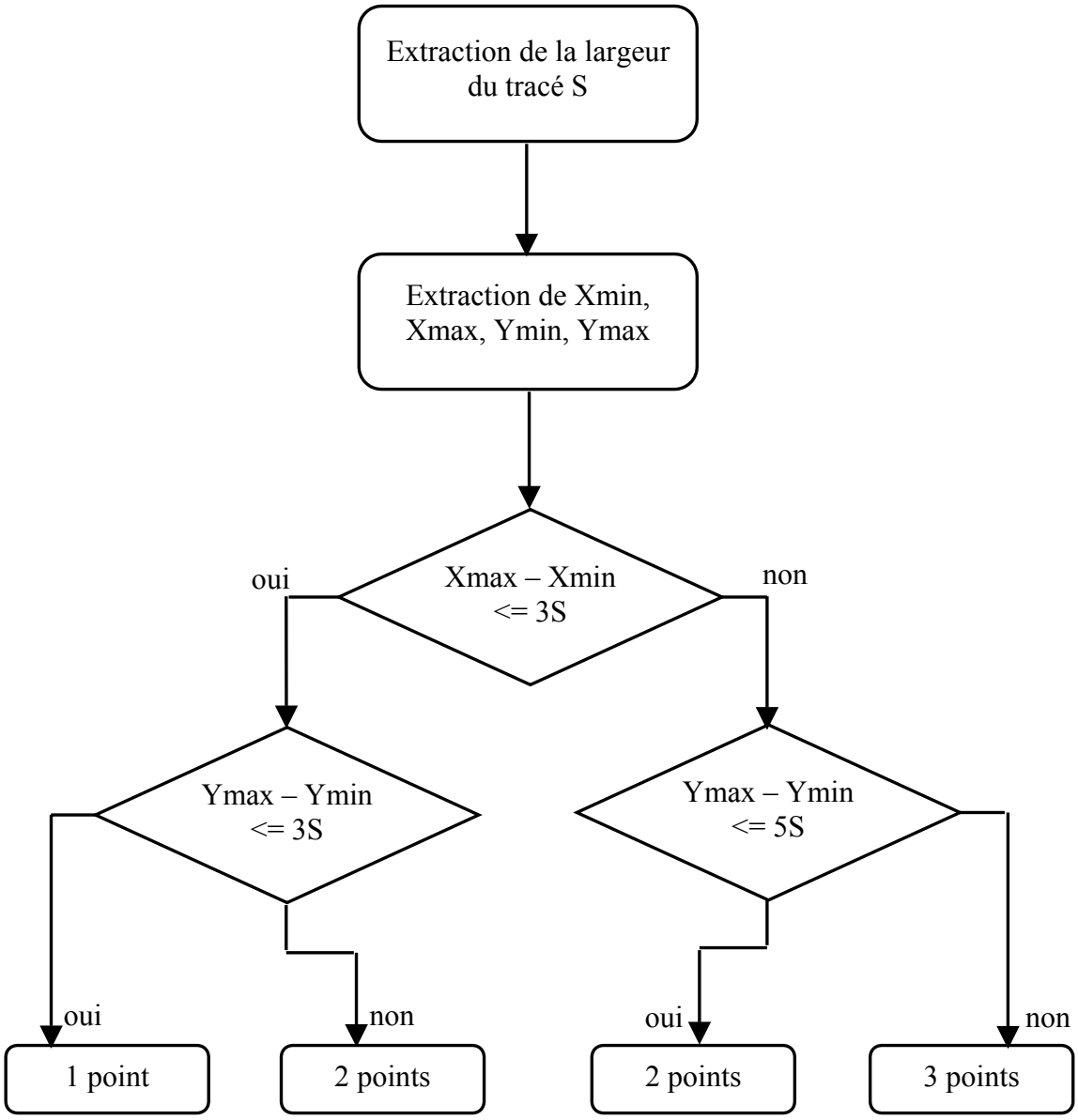


FIG. 4.17 – *Extraction des points diacritiques*

2.2.2.2. Extraction des boucles

Elles correspondent aux contours internes dans les tracés primaires des caractères. Les occlusions dans le cas de l'écriture arabe sont dans la zone médiane et à proximité de la ligne de base du caractère. L'algorithme d'extraction de contour utilisé produit deux types de contours :

- *les contours externes* : correspondant aux tracés primaires des caractères et aux points diacritiques ;
- *les contours internes* : correspondant aux boucles.

Les boucles sont utilisées pour identifier certains caractères. La présence ou non d'une boucle, le nombre de boucles sont des caractéristiques à déterminer. Par exemple, le caractère Haa (ھ) est le seul caractère arabe qui contient deux boucles.
Pour détecter les boucles, il faut parcourir la liste de tous les contours du caractère et comparer chaque contour avec les autres. Si toutes les cordonnées d'un contour englobent toutes les coordonnées correspondantes d'un autre, alors il existe une boucle.

2.3. Apprentissage

Après l'extraction de caractéristiques, on crée un fichier qui contient les données d'apprentissage (data file), ce fichier est l'un des paramètres passés au programme d'apprentissage des SVM (SVM-training) pour la création du fichier modèles.

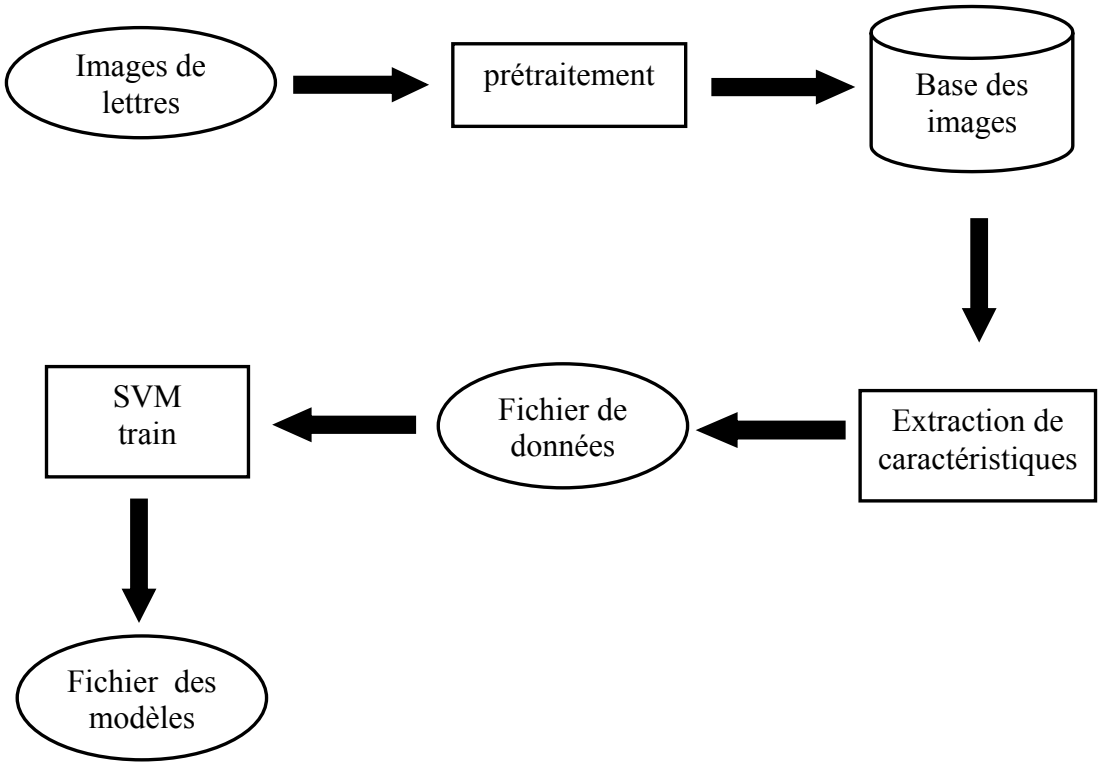


FIG. 4.18 – *La phase d'apprentissage*

2.4. Décision

Architecture du système en stratégie un contre tous

L'alphabet arabe comprend 28 caractères et à chaque caractère peut correspondre jusqu'à quatre formes différentes, ce qui lève à environ 96, le nombre de formes à reconnaître. Cependant, le SVM est un classifieur binaire qui ne traite habituellement que des données étiquetées par rapport à deux classes d'appartenance. Donc, traiter des données multiclasses comme c'est le cas en reconnaissance des caractères, requiert la combinaison d'un ensemble de SVM, chacun se spécialise en une partie du problème.

Nous cherchons à modéliser une fonction $G : \Omega \rightarrow 1,\ldots,K$, qui définit K partitions dans l'espace des caractéristiques Ω. Connaissant G, les partitions des classes sont définies par $G^{-1}(c)$, où $c \in [1\ldots K]$ (K est le nombre de classes).

Pour un problème à deux classes, l'hyperplan *(w,b)* du SVM délimite les deux partitions selon *sign f(x)* ou $f(x) = w.x + b$. Par ailleurs, bien que le SVM soit un classifieur binaire, il peut être étendu pour décider de l'appartenance des données multiclasses.

Comme notre problème de classification de caractères est un problème multiclasses, on a pensé à deux possibilités. La première consiste à entraîner un ensemble de SVM, chacun se séparant un couple de classes *(i,j)* parmi ceux existant. La sortie de chaque de SVM fournit un vote partiel concernant uniquement le couple de classes (w_i,w_j). Cela nécessite, dans notre cas, la construction de 4560 SVM (1/2. 96.(96-1)), et chaque SVM nécessite le calcul des hyper-paramètres de ses fonctions noyaux.

Vu ce nombre important de SVM, on a opté pour une deuxième méthode qui consiste à entraîner autant de SVM que de classes (96 SVM) séparant chaque classe des 95 classes restantes (Un-Contre-Tous) [MEH b06]. Par exemple, le premier SVM sépare la classe du « Alif » des classes restantes, la classe du caractère « Alif » est considérée comme la classe positive et tous les caractères restant forment la classe négative.

Durant l'apprentissage, tous les exemples appartenant à la classe considérée sont étiquetés positivement (+1) et tous les exemples n'appartenant pas à cette classe sont étiquetés négativement (-1). A la fin de l'apprentissage, nous disposerons de 96 modèles correspondant aux hyperplans (w_i, b_i) tels que $i = 1,...,96$.

Durant le test, le caractère en entrée est associé à la classe dont la sortie est positive selon la règle $x \in C_k$ *si* $w_i .x+b_i > 0$ *pour* $i=k$. Or, il est possible que plusieurs sorties soient positives pour un exemple de test donné. C'est particulièrement le cas des données ambiguës situées près des frontières de séparation des classes. Dans ce cas, un vote majoritaire pour attribuer l'exemple x à la classe C_k selon la règle de décision suivante :

$$c = arg\ max\ (w_i.x + b_i) \tag{4.9}$$

Les 96 SVM sont entraînés indépendamment les uns contre les autres. Ils produisent des vecteurs de support différents. La figure 4.19 illustre l'architecture du système en stratégie un contre tous. Notons, que l'étage étiqueté « *fusion* » désigne le schéma de vote utilisé et aussi toute sorte d'expert capable de fusionner les sorties pour décider de la classe d'appartenance.

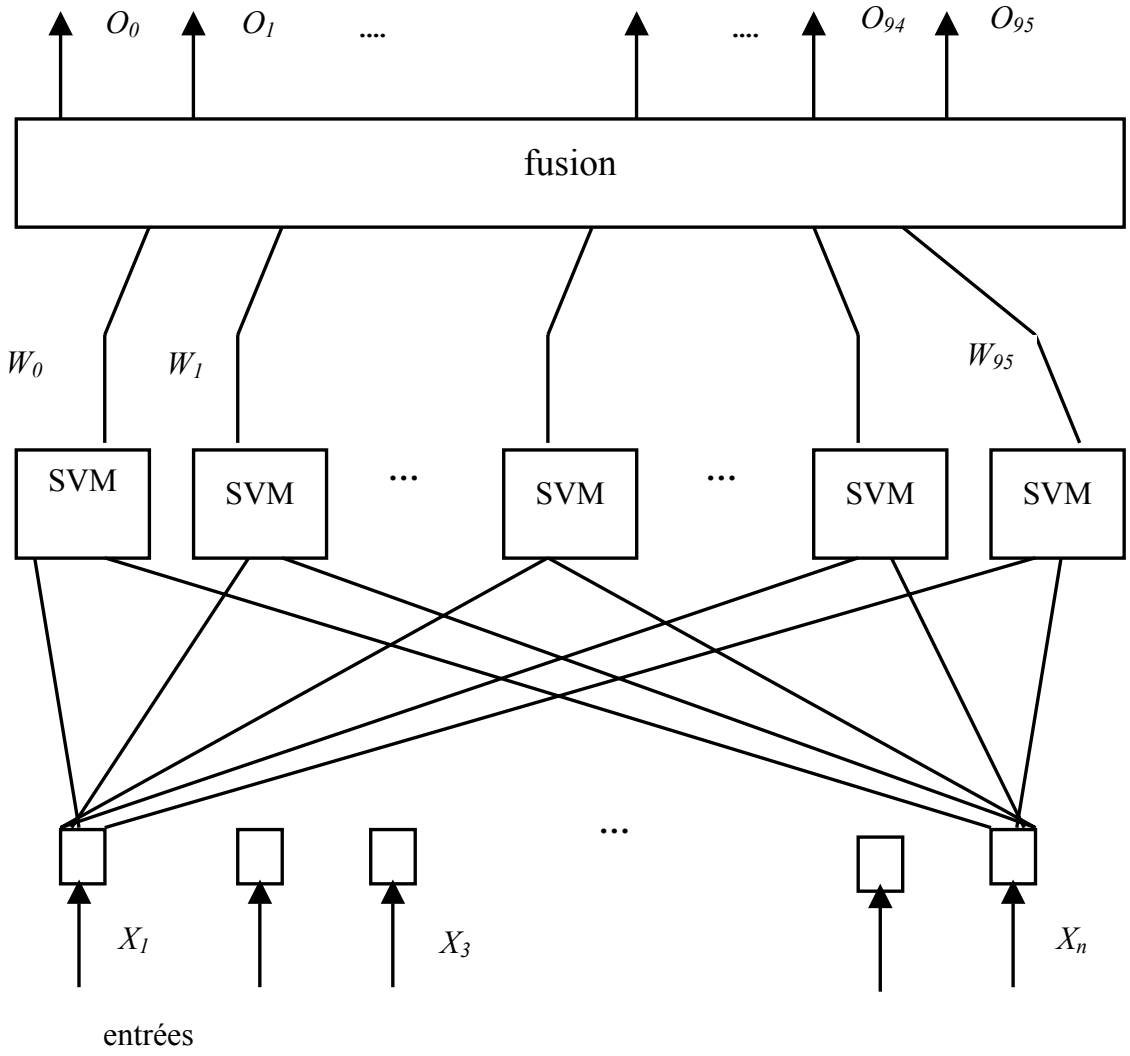


FIG. 4.19 – *Architecture du système en stratégie Un-Contre-Tous*

3. Sélection du modèle par validation d'erreur

Parmi les difficultés majeures liées à l'utilisation de classifieurs figure la nécessité d'adapter les variables conditionnant le processus d'apprentissage et ne figurant pas dans la fonction de décision finale. Ces variables sont appelés hyper-paramètres. Si ces paramètres

sont adéquatement choisis, ils permettront d'éviter les situations de sur-apprentissage fréquentes lorsque les données sont bruitées.
Plusieurs critères de sélection sont utilisables pour le choix des valeurs des hyper-paramètres. Si le modèle utilise un seul hyper-paramètre, il est possible d'essayer un nombre fini de valeurs et choisir celle qui minimise le critère. Cette technique devient toutefois difficile à implémenter pour deux hyper-paramètres ou plus. Parmi les hyper-paramètres du SVM, on trouve les paramètres des noyaux et la variable de compromis C.

Soit K une fonction noyau d'un ou plusieurs paramètres encodés dans un même vecteur $\theta(\theta_1, \theta_2, ..., \theta_N)$. Le SVM modélise la classe de fonctions conditionnées par α, b et θ selon l'équation :

$$f_{\alpha,b,\theta} = \sum \alpha_i y_i K_\theta(x, x_i) + b \tag{4.10}$$

α représente le vecteur des multiplicateurs α_i associés aux couplets de données (x_i, y_i) ; ou x_i et y_i représentent respectivement le vecteur d'entrée et son étiquette. Sa taille est égale au nombre de vecteurs de support.
Le vecteur de paramètre θ est composé de paramètres du noyau choisi pour le SVM. Ces paramètres influencent la performance du SVM, on citera le paramètre σ du RBF, le triplet (a,b,d) du noyau polynomial, le couplet (a,b) de la sigmoïde.

NOYAU	FORMULE
Linéaire	$K(x,y) = x.y$
Polynomial	$K(x,y) = (ax.y + b)^d$
RBF	$K(x,y) = \exp(-\|x-y\|^2/\sigma^2)$

TAB 4.4 – *Noyaux de Mercer utilisés*

La variable C du SVM, est un hyper paramètre additionnel qui influence la généralisation du classifieur lorsque les classes sont bruitées.

En présence de données multiclasses, comme c'est le cas en reconnaissance de caractères, nous avons vu, dans la section précédente, qu'il est nécessaire de partager la tâche de classification entre plusieurs SVM et de combiner les réponses produites par les différents SVM en vue de prédire l'appartenance d'une observation donnée. Cette combinaison des SVM rend plus complexe la tâche d'optimisation, d'autant plus que le nombre des hyper-paramètres dépend non plus du noyau, mais aussi des dichotomies associées aux classifieurs. Donc, pour notre système, il faut considérer 96n (nK) variables à optimiser (en considérant que le même type de noyau est utilisé avec chaque classifieur et le nombre d'hyper-paramètres du noyau est égal à n et que le nombre de classes est égal à 96).

Afin de choisir les valeurs des hyper-paramètres, il nécessaire de spécifier un critère permettant de sélectionner un modèle parmi plusieurs possibles.

Dans le but de trouver le SVM qui a une capacité de généralisation meilleure, nous utilisons un ensemble de données indépendant de celui de l'apprentissage pour l'évaluation du classifieur. Nous entraînons plusieurs SVM sur un même ensemble d'apprentissage avec toutefois différentes valeurs d'hyper-paramètres. Leurs performances sont par la suite comparées en estimant l'erreur de chacun d'eux sur un ensemble de validation indépendant de celui de l'apprentissage. Le SVM qui fournit la plus faible erreur est le plus approprié à choisir. La figure 4.2 schématise l'algorithme de l'optimisation de l'ensemble de SVM dans l'approche un contre tous. Après chaque correction des valeurs des hyper-paramètres, le SVM est ré-entrainé et ainsi de suite jusqu'à la convergence de l'algorithme.

Afin d'éviter le sur apprentissage des exemples de validation, le classifieur choisi est à son tour évalué sur un ensemble de tests indépendants.

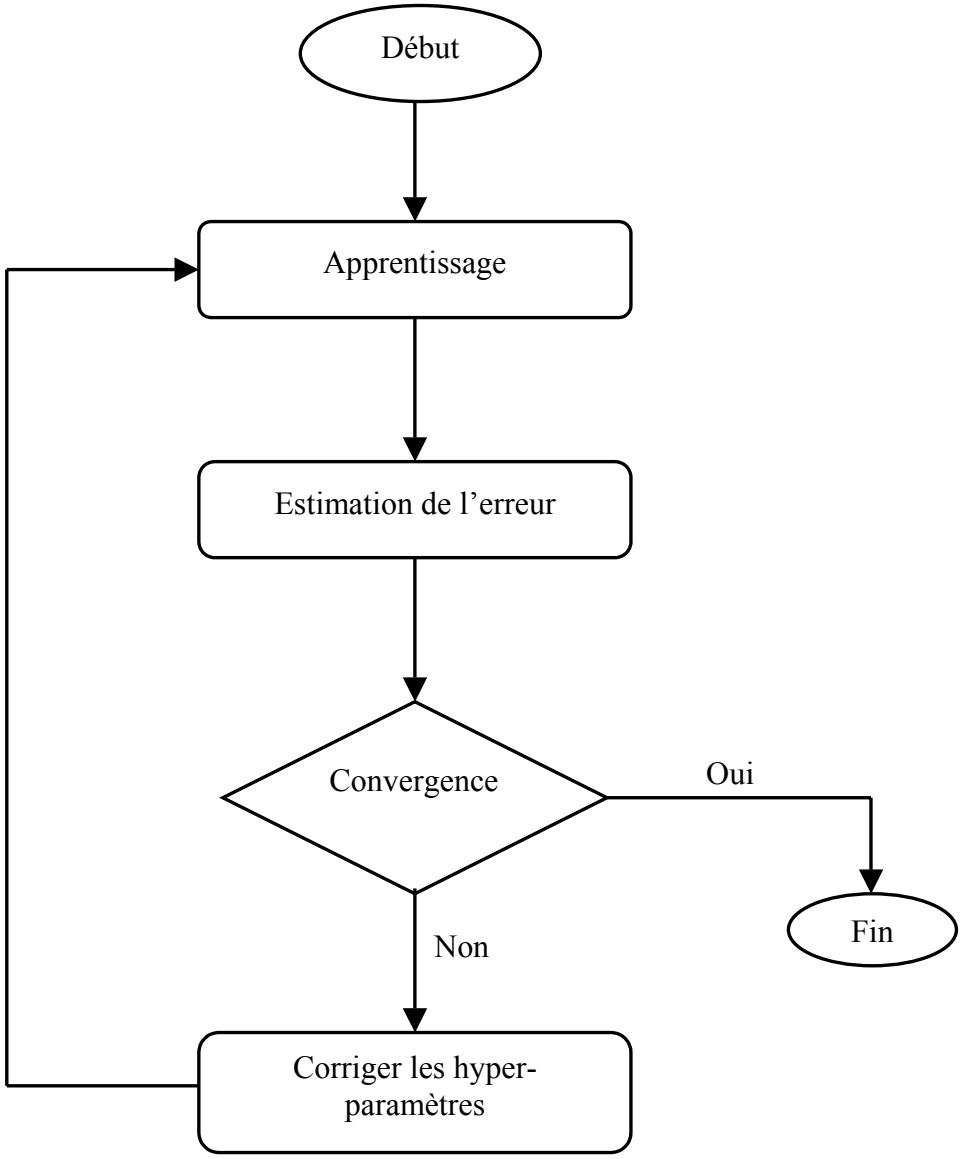


FIG. 4.20 – *Les étapes d'optimisation des hyper-paramètres avec réduction de l'erreur*

4. Base de données

L'estimation des performances nécessite des exemple, donc, il est nécessaire d'avoir une base de données afin d'effectuer tous les traitements. Vu l'absence d'une base standard de lettres arabes manuscrites, une base locale de 4224 images de caractère, a été construite.

Cette base est composée de 96 formes de caractères (à chaque caractère peur correspondre jusqu'à quatre formes différentes, ce qui lève à environ 96 le nombre de formes à reconnaître) .

Pour la constitution de la base d'images, nous avons sollicité 16 scripteurs (des universitaires). Les documents écris par ces scripteurs, sont scannés et leurs contenus sont traites. Une première étape d'étiquetage des lettres de la base est générée. La figure suivante illustre quelques échantillons de caractères arabes, issus de quatre scripteurs différents.

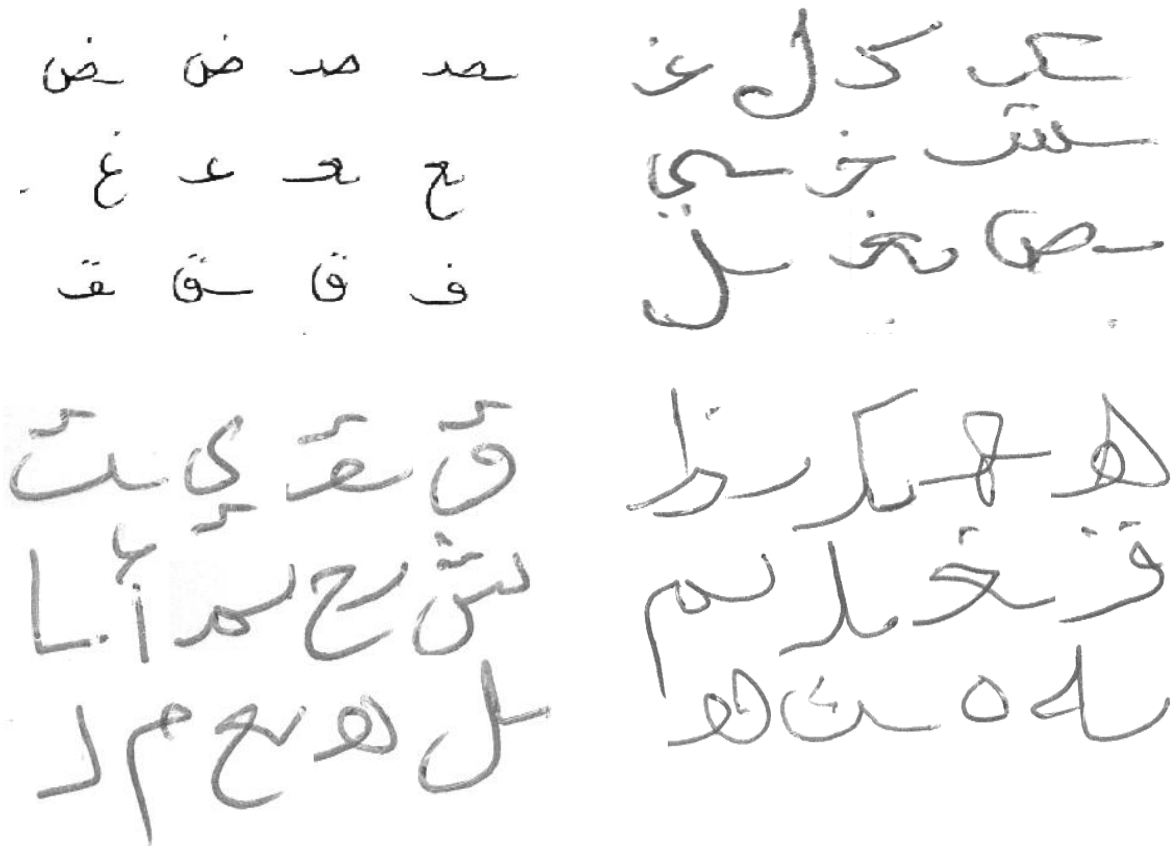


FIG. 4.21 – *Echantillons de lettres arabes manuscrites*

Les résultats de test décrits ci après ont été réalisés sur un ensemble d'apprentissage contenant 2880 caractères et un ensemble de test de 1344 caractères. L'ensemble de test est divisé en deux : l'un est dépendant (ses échantillons sont issus de même scripteurs produisant la base d'apprentissage) et contenant 576 images, et l'autre est indépendant de

l'ensemble d'apprentissage, elle contient 768 images de caractères. La base de test constitue 36% de la base d'apprentissage des caractères. La figure 4.22 montre la décomposition de la base construite.

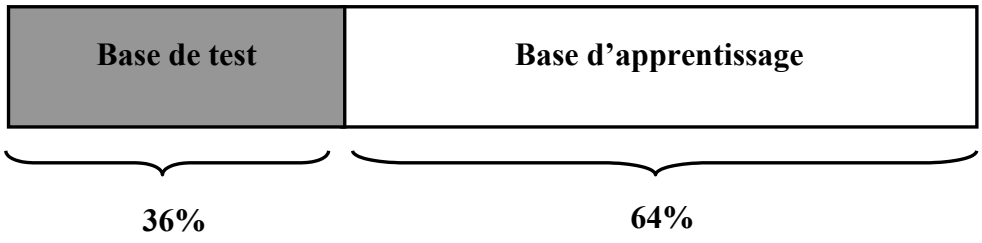


FIG. 4.22 - *Décomposition de la base des caractères*

Le tableau 4.5 indique quelques détails sur la base d'apprentissage. Le nombre d'échantillons par classe est 30 provenant de 10 scripteurs, chacun écris le caractère 3 fois.

Classe	Caractère	Nbr de Script	Echant /Scrip	Echant/classe
0	أ	10	3	30
1	ا	10	3	30
.	.	.	.	.
.	.	.	.	.
.				.
94	ي	10	3	30
95	ﻱ	10	3	30
Totale	2880 caractères			

Nbr : nombre de scripteurs, Echant/Scrip : nombre d'échantillons écris par chaque scripteur, Echant/classe : nombre d'échantillons par classe

TAB 4.5 - *Description de la base d'apprentissage*

Le tableau 4.6 décrit la base de test indépendante.

Classe	Caractère	Nbr de Script	Echant /Scrip	Echant/classe
0	أ	8	1	8
1	ا	8	1	8
.	.	.	.	.
.	.	.	.	.
.				.
94	ي	8	1	8
95	ﻳ	8	1	8
Totale	768 caractères			

Nbr : nombre de scripteurs, Echant/Scrip : nombre d'échantillons écris par chaque scripteur, Echant/classe : nombre d'échantillons par classe

TAB 4.6 – *Description de la base de test indépendante*

La base de test dépendante est constituée de la même façon que la base indépendante, sauf que le nombre de scripteurs est six.

5. Tests, résultats et discussions

Durant les expérimentations effectuées, nous avons utilisé la stratégie un contre tous, les modèles sont optimisés de façon expérimentale l'un après l'autre.

Il évident que le choix des hyper-paramètres de façon expérimentale est très gourmand en temps de calcul. De plus, il est nécessaire d'initialiser les paramètres des noyaux à des valeurs raisonnables qui assurent que la fonction objective est convexe, et donc une solution exacte existe

Le but des expériences adressées dans cette partie est de présenter les performances des SVM avec différents noyaux. D'après les résultats, on a remarqué le taux d'erreur est très élevé en utilisant un noyau polynomial de degré 3, par contre, le taux de reconnaissance

est significativement augmenté avec un noyau linéaire, et un noyau polynomial de degré 1 et 2 . Le noyau RBF a donné des résultats meilleurs que les autres noyaux, particulièrement avec σ =35 (81,64%).

Avec la base de test issue de même scripteurs produisant la base d'apprentissage, les meilleurs taux de reconnaissance sont obtenus avec un noyau RBF de σ égal 35 (98%), un noyau linéaire, et un noyau polynomial de degré d=1 (97,4%). 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 RBF.

Les résultats obtenus sont rapportés dans les tableaux 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, notons que ces résultats sont obtenus en testant seulement les caractéristiques statistiques.

- **La base indépendante**

Noyau linéaire	
Taux de reconnaissance	78,79%

TAB 4.7: *Le taux de reconnaissance obtenu avec le noyau linéaire*

Noyau polynomial	
Degré (d)	Taux de reconnaissance
1	78,78 %
2	79,04 %
3	1.82 %

TAB 4.8 - *Les taux de reconnaissance obtenus avec le noyau polynomial*

Noyau RBF	
σ	Taux de reconnaissance
8	72,01 %
10	72,01 %
15	72,01 %
20	80,99 %
25	81,51 %
35	81,64 %

TAB 4.9 - *Les taux de reconnaissance obtenus avec le noyau RBF*

- **La Base dépendante**

Noyau linéaire	
Taux de reconnaissance	98 %

TAB 4.10 - *Le taux de reconnaissance obtenu avec le noyau Linéaire*

Noyau polynomial	
Degré (d)	Taux de reconnaissance
1	97,40 %
2	97,05 %
3	2,08 %

TAB 4.11 – *Les taux de reconnaissance obtenus avec le noyau Polynomial*

Noyau RBF	
σ	Taux de reconnaissance
8	95,32 %
15	95,32 %
20	97,4 %
25	97,75 %
35	98 %

TAB 4.12 - *Les taux de reconnaissance obtenus avec le noyau RBF*

6. Conclusion

Dans ce chapitre, nous avons présenté un système à base des SVM pour la reconnaissance des caractères arabes manuscrits, en nous basant sur plusieurs traitements: binarisation, lissage, extraction du contour, extraction de caractéristiques et finalement la décision. La classification des caractères arabes est un problème multiclasses, bien que le SVM soit un classifieur binaire. De ce fait, nous avons combiné plusieurs SVM pour décider la classe d'appartenance de chaque caractère. Le système a été testé sur une base contenant 4224 images. Cette base nous l'avons construire en l'absence d'une base standard pour les lettres arabes manuscrites, qui nous permet de comparer notre système avec ceux de la littérature. Pour la résolution du problème quadratique, nous avons utilisé le SVM-TORCH. Le système est à son premier stade d'expérimentation, les résultats obtenus avec les caractéristiques statistiques sont encourageants, La méthode des SVM est aisée d'emploi et les paramètres à régler ne sont pas nombreux. Le prototype du système complet est en cours de contruction.

CONCLUSION GENERALE

Le présent travail s'inscrit dans le cadre générale de la reconnaissance automatique de l'écriture arabe, plus précisément, il exploite une approche basée sur les machines à vecteurs de support appliquée à la reconnaissance des lettres arabes manuscrits.

Cette technique fournie un cadre théorique, statistique et non connexionniste à l'apprentissage. Le pouvoir de cette méthode réside dans la construction d'un hyperplan de séparation à marge maximal entre les données d'entrées. La maximisation de marge est une méthode de sélection du modèle implicite à l'apprentissage qui minimise les erreurs d'apprentissage tout en réduisant la complexité du classifieur.

Le SVM représente un modèle dont la complexité est en fonction du nombre de vecteurs de support et des valeurs des hyper-paramètres, mais elle ne dépend pas du nombre de caractéristiques. Cette propriété rend le classifieur plus robuste au phénomène de sur-apprentissage lorsque la quantité des données disponibles est modeste.

En outre, l'utilisation des noyaux de *Mercer* permet de projeter les données d'entrées dans un espace augmenté de grande dimension, dans lequel une séparation linéaire des classes est possible.

Dans notre système, nous avons opté pour une stratégie *Un-Contre Tous* pour décider de la classe d'appartenance de chaque caractère. 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 ouvre 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. Un choix adéquat des hyper-

paramètres améliore le taux de reconnaissance et permet d'éviter les situations de sur-apprentissage lorsque les données sont bruitées.
L'absence d'une base de lettres arabes manuscrites qui est standard sur laquelle nous pouvons tester le système proposé, nous a contraint à travailler sur une base locale composée de 4224 caractères.
Les résultats préliminaires obtenus pour le premier jeu de caractéristiques employé sont encourageants, car la qualité de la base de données est très réaliste avec des erreurs et des différences d'écriture d'un scripteur à un autre. Afin d'améliorer les taux de reconnaissance obtenus, nous avons pensé à utiliser un deuxième type de caractéristiques basé sur les primitives visuelles de l'alphabet arabe, ces caractéristiques ne sont pas encore testées.

Ce système est en premier stade d'expérimentation et l'expérience s'est avérée intéressante mais des extensions restent envisageables :

- Elargir la base de données en introduisant un plus grand nombre des scripteurs, pour inclure plus de variations dans les styles d'écritures, ainsi la généralisation pourra être plus performante.
- Tester la totalité des caractéristiques choisies, et introduire de nouvelles caractéristiques des caractères arabes telles que les ouvertures et leurs directions.
- Etendre le système pour l'appliquer à la reconnaissance des mots arabes moyennant une approche analytique ou globale. Dans le cas de l'approche analytique, il faut procéder par la segmentation des mots en caractères et effectuer la reconnaissance par le système proposé.
- Tirer profils des algorithmes de la sélection automatique du modèle des machines à vecteurs de support pour optimiser les paramètres des noyaux et réduire la complexité des classifieurs.
- Utiliser autre critère, que la validation d'erreur, pour le choix des valeurs des hyper-paramètres.
- Penser à la possibilité de combiner les SVM avec d'autres classifieurs. A l'origine, le SVM est un classifieur binaire, donc, il sera intéressant de construire un

classifieur SVM-PMC pour la reconnaissance des caractères arabes. L'utilisation du SVM sera uniquement pour les classes qui sont sources de la majorité des confusions du réseau PMC. Pour classifier une forme de caractère inconnu, le système procède, dans le pire des cas, à une décision de réseau PMC et une décision du classifieur SVM.

Bibliographie

[ABD 90] H.Y.Abdelazim, M.A.Hashish, " Arabic typeset : an OCR approach", *Proc 5th EUSIPCO-90, European Signal Processing Conference*, pp.1019-1022, Barcelona, Spain, 1990.

[ABU 94] I.S.I.Abuhaiba, S.A.Mahmoud, R.J.Green, "Recognition of Handwritten Cursive Arabic Character", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 16, pp.664-672, June 1994.

[ALB a95] B.Al Badr, S.A.Mahmoud, "Survey and Bibliography of Arabic Optical Text Recognition", *Signal processing*, vol 41, pp.49-77, 1995.

[ALB b95] B.Al Badr, R.M.Haralik, "Segmentation Free Word Recognition with Application to Arabic", *IEEE Proceedings of ICDAR'95*, pp.355-359, 1995.

[ALM 87] H. Almuallim, S. Yamaguchi, "A Method of Recognition of Arabic Cursive Handwriting", *IEEE*, T*rans*. *Pattern Anal*. *Mach*. *Intell*. PAMI-9, pp.715–722, 1987.

[ALY 92] H.Al Yousfi, S.S.Udpa, "Recognition of Arabic Characters", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol 14, pp.853-857, Aug 1992.

[AME 94] A.Ameur, K.Romeo-Pakker,H.Miled, M.Chriet, "Approche Globale pour la Reconnaissance des Mots Manuscrits arabes", *Actes CNED94, 3ème Colloque National sur l'écrit et le document*, pp.151-156, Juillet 1994.

[AMI 80] A.Amin, A.Kaced, J.P.Haton, R.Mohr, "Handwritten Arabic Character Recognition by the IRAC System", *Proceeding of ICPR'80, 5th International Conference on Pattern Recognition*, pp 729-731, October 1980.

[AMI 82] A.Amin, "Machine Recognition of Handwritten Arabic Words by IRACII System", *Proceedings of ICPR'82, 6th International Conference on Pattern Recognition*, vol 1, pp.34-36, October 1982.

[AMI 85] A.Amin, G.Masini, "Deux Méthodes de Reconnaissance de Mots pour l'Ecriture Arabe Manuscrites", *Actes RFIA'85, 5ème congrès Reconnaissance des formes et Intelligence*

[AMI 86] A.Amin, G.Masini, "Machine Recognition of Multifont Printed Arabic Texts", *Proceedings of ICPR'89, 8th International Conference on Pattern Recognition,* vol 1, pp.392-395, October 1986.

[AMI 89] A. Amin,J. F. Mari, "Machine Recognition and Correction of Printed Arabic Text", *IEEE* [1]*rans*. *Man Cybernet*. 9, pp.1300-1306,1989.

[AMI 91] A.Amin, "Recognition of Arabic Handprinted Mathematical Formulate", *The Arabian Journal for Science an Enginneering*, Vol 16, N[0]: 4B, pp.531-542, October 1991.

[AMI 92] A.Amin, H.B.Al Sadoun, "A new Segmentation Techniques of Arabic text", *IEEE Proceedings 11th IAPR International Conference on Patten Recognition, Patten Recognition, Conf B: Pattern Recognition Methodology and Systems",* pp.441-445, 1992.

[AMI a96] A.Amin, H.Al Sadoun, S.Fisher, "Hand-Printed Arabic Character Recognition System Using an Artificial Network", *Pattern Recognition*, vol 29, N[o] 4, pp.663- 675, 1996.

[AMI b96] A.Amin, S.Fisher, T.Parkinson, R.Shiu, "Fast Algorithm for Skew Detection", *SPIE Proceeding*, vol 2661, pp.29-30 Janvier, 1996.

[AMI 97] A. Amin ,W. Mansoor, "Recognition of Printed Arabic Text using Neural Networks", *Proc. 4th Int. Conf. on Document Analysis Recognition,* Ulm, Germany, 1997.

[AMI 98] A.Amin, " Off Line Arabic Character Recognition: The State of The Art", *Pattern Recognition*, vol 31, N[o] 5, pp.517-530, 1998.

[AMI a00] A.Amin, "Recognition of Printed Arabic text Based on Global Features and Decision Tree Learning Techniques", *Pattern Recognition*, vol 33, pp.13091323, 2000.

[AMI b00] A.Amin, N.Murshed, "Off-line Recognition of Printed Arabic Words Through Global Features and Neural Networks", *4th International Workshop on Document Analysis Systems*, DAS'2000, pp.267-277, Rio de Janeiro, Brazil, 2000.

[AYA 00] N.D. Ayat, M.Cheriet, C.Y.Suen, "Un Système neuro-flou pour la Reconnaissance de Montants Numériques de Chèques Arabes, In Colloque International Francophone sur l'écris et le documents, pp 171-180, France, 2000.

[AYA 04] N.D. Ayat, Sélection Automatique de Modèle Dans Les Machines à Vecteurs de Support: Application à La Reconnaissance d'Image De Chiffres Manuscrits, Thèse de Doctorat, Université de Québec, 2004.

[AZI a02] N.Azizi, Combinaison de Classifieurs neuronaux bésée sur la logique floue: application à la reconnaissance des mots arabes manuscripts, Mémoire de Magistère, Université de Annaba, 2002.

[AZI b02] N.Azizi, L.Souci, M.Sellami, " Une Architecture de Combinaison floue de Classifieurs Neuronaux pour la Reconnaissance de Mots Arabes Manuscrits", *CIFED'2002, Colloque International Francophone sur l'Ecrit et le Document*, Hammemet, pp 89-96, Tunisie, Octobre 2002.

[BAG 97] A.Bagdanov, J.Kanai, "Projection Profile Based Skew Estimation Algorithm for JBIG Compressed Images", *International Conference on Document Analysis and Recognition*, pp 401-405, 1997.

[BAH 02] C.Bahlman, B.Haasdonk, H.Bukhardt, "On-line Handwriting Recognition with Support Vector Machines – A Kernel Approch", *Proc. Of the 8th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR),* pp. 49–54, 2002.

[BAL 70] G.J. Balm, "An introduction to optical character reader considerations. *Pattern Recognition*, 2(3):pp.151–166, 1970.

[BAP 88] G.Baptista, K.M.Kulkarami, "A High Acuracy Algorithm for Recognition of Handwitten Numerals", *Pattern Recognition*, pp.287-291, 1988.

[BEL 92] A.Belaid et Y. Belaid, Reconnaissance des Formes : Méthodes et applications, InterEditions, janvier 1992.

[BEL 01] A.Belaid, " Reconnaissance Automatique de l'Ecriture et du Document". Pour la science, 2001.

.

[ESS 97] N.Ben Amara, "Application des PHMMs pour la Reconnaissance de l'Ecriture Arabe Imprimée", *JST'97-Francil*, pp.389-392, Avignon -France Avril 1997.

[BEN 98] A.Benouareth, M.Sellami, "Proposition d'une Méthode Structurelle pour la Reconnaissance des Mots Arabes Manuscripts par Approche Globale", *JECHM'98-INI*, Alger, pp 144-151, Juillet 1992.

[BEN 00] A. Benouareth, Reconnaissance de l'Ecriture Arabe Manuscrite par une Approche Hybride", Mémoire de Magister, Université de Annaba, 1999.

[BER 98] S.Bergler, S.Khoury, B.C.Y.Suen, B.Waked, " Skew Detection, Page Segmentation and Script Classification of Printed Document images", *IEEE International Conference on Systems Man and Cybernetics*, pp.4470-4475, October 1998.

[BLA 96] V. Blanz, B. Scholkopf, H.H. Bulthoff, C. Burges, V. Vapnik,T.Vetter. "Comparison of View-Based Object Recognition Algorithms Using Realistic 3d Models". In *ICANN*, pp.251–256, 1996.

[BOS 92] B.E.Boser, I.M.Guyon, V.N.Vapnik,"A Training algorithm for Optiml Margin Classifiers", *In fifth Annual Workshop on Computational Learning Theory*, Pittburg, 1992.

[BOU 97] V. Bouletreau, N. Vincent, R. Sabourin, "Synthetic Parameters for Handwriting Classification", *ICDAR*, pp.102-106, 1997.

[BUR 98] C.Burges, "A Tutotial on Support Vector Machine for Pattern Recognition", *Data Mining and Knowledge,*2(2): p.21-167,1998.

[CAL 03] J.Callut, Implémentation Efficace des Support Vector Machines pour la Classification, Master, Université Libre de Bruxelles, 2003.

[CAE 93] T. Caesar, J. M. Gloger, E. Mandler, "Processing and Feature Extraction for a Handwriting Recognition System", *International Conference on Document Analysis a Recognition,ICDAR*, pp.408-411, 1993.

[CHH 93] A. K. Chhabra, Z. An, D. Balick, G. Cerf, K. Loris, P. Sheppard and R. Smith, B. Wittner, "High-Order Statistically Derived Combinations of Geometric Features for Handprinted Character Recognition", *International Conference on Document Analysis a Recognition, ICDAR*, pp.397-401, 1993.

[CHE 98] M.Cheriet, H.Miled, C.Olivier, Y.Lecourtier, "Visual Aspect of Cursive Arabic Handwriting Recognition", *Proceedings Vision Interface*, *VI'98*, pp.262-270, 1998.

[CHI 98] Y.-C. Chim, A.A. Kassim, Y. Ibrahim. "Dual Classifier System for Handprinted Alphanumeric Character Recognition". *Pattern Analysis and Application*, pp 155-162, 1998.

[CHO 65] C.K.Chow, "Statistical Independence and Threshold Functions", *IEEE Transactions of Electrical Computer*", pp 66-68, 1965.

[CHR 97] Chris J.C. Burges, B. Schölkopf. "Improving the Accuracy and Speed of Support Vector Machines". In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, vol 9, pp 375,*The MIT Press*, 1997.

[COR 95] C. Cortes, V.Vapnik, "Support-vector networks", *Machine Learning*, 20(3): pp. 273-297, 1995.

[COR 02] A.Cornuéjols, "Une Nouvelle Méthode d'Apprentissage: Les SVM. Séparateurs à Vaste Marge", *Rapport de stage*, 2002.

[COT 98] M. Côté, E. Lecolinet, M. Cheriet and C.Y. Suen, "Automatic Reading of Cursive Scripts Using a Reading Model and Perceptual Concepts", *International Journal on Document Analysis And Recognition*, *IJDAR,*vol 1, Num 1, pp.3-17, 1998.

[EIK 96] A.K.Eikvil, «Text Page Recognition Using Grey-Level Features and Hidden Markov Models", *Pattern Recognition*, vol 29, Num 6, pp.977-995, 1996.

[ELD 90] S.S.El Dabi, R.Ramsis, A.Kamel, "Arabic Character System: A System Approach for Recognizing cursive Typewritten Text", *Pattern Recognition*, vol 23, Num 5, pp.485-495, 1990.

[ELK 90] F.El Khaly, M.A..Sid Ahmed, "Machine Recognition of Optically Captured Printed Text", *Pattern Recognition*, vol 23, Num 11, pp.1207-1214, 1990.

[ELW 89] M.S. El-Wakil and A. Shoukry, "On-line Recognition of Handwritten Isolated Arabic Characters", *Pattern Recog nition*, vol. 22, Num 2 pp.97-105, 1989.

[ESS 95] N. Essoukri Ben Amara, N.Ellouze, "A Robust Approach for Arabic Printed Character Segmentation", *IEEE. Proc. 3rd International Conference on Document Analysis and Recognition (ICDAR'95),* pp 220-224, Vienne, Autriche, 1996

[ESS 99] 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.

[ESS 02] N.Essoukri Ben Amara, "Problématique et Orientations en Reconnaissance de l'Ecriture Arabe", *CIFED'2002, Colloque International sur l'Ecrit et le Manuscrit, pp 1-10, Hammamet*, Tunisie, Octobre 2002.

[EVE 00] T.Evegeniou, Learning with Kernel Machine Architecture, Departement of Electrical Engineering and Computer Science, 2000.

[FAR 05] N.Farah, L.Souici, M.Sellami, "Arabic Word Recognition by Classifiers and Context", JCST Journal of Computer Science and Technology, Vol 20, Num 3, pp 402-410, May 2005.

[FIL 98] A.Filatov, N. Nikitin, A. Volgunin, P. Zelinsky. "The *Address* Script Recognition System for Handwritten Envelopes". In *International Association for Pattern Recognition Workshop on Document Analysis Systems (DAS'98)*, pp.157–171, Nagano, Japan, November, 1998.

[GOV 90] V.Govindan, A.P.Shivaprasad, " Character Recognition: A Review", *Pattern Recognition*, vol 23, Num 7, pp.671-683,1990.

[GOV 94] V.Govindaraju, R.K.Srihari, S.N. Srihari, "Handwritten Text Recognition", *In Internal Association for Pattern Recognition Workshop on Document Analysis Systems (DAS'94)*, pp 157-171, *Kaiserslautern, Germanny,* September 1994.

[GRA 03] F.Grandidier, "Un Nouvel Algorithme de Sélection de Caractéristiques : Application à la Lecture Manuscrite", Thèse de Doctorat, Montréal, 2003.

[HAL 04] K.Hallouli, Reconnaissance des Caractères par Méthodes Markoviennes et Réseaux Bayésiens, Thèse de Doctorat, Paris, 2004.

[HAM 93] M. Hamanaka, K. Yamada, J. Tsukumo, "On-line Japanese Character Recognition Experiments by an Off-line Method Based on Normalization Co-Operated Feature Extraction", *ICDAR*, pp. 204-207, 1993.

[HU 62] M.K. Hu. "Visual Pattern Recognition by Moments Invariants". *IRE Trans. on Information Theory*, 8: pp 179–187, 1962.

[JOA 99] T. Joachims. "Making large-scale svmlearning practical". In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapitre 11,1999.

[KIM 99] G. Kim, V. Govindaraju and S. N. Srihari, "An architecture for Handwritten Text Recognition Systems, *International Journal on Document Analysis and Recognition*, *IJDAR,* pp. 37-44, 1999.

[MUL 01] KR.Muller, S. Mika, G. Ratsch, K.Tsuda, B. Schôlkopf. "An introduction to kernel-based learning algorithms". *IEEE Transactions on Neural Networks*, 12 :pp.181-201, 2001.

[KNO 69] A.L. Knoll. "Experiments with Characteristic Loci for Recognition of Handprinted Characters". *IEEE Trans. on Computers*, 18: pp.366–372, 1969.

[KWA 02] H.K.Kwag, S.H.Kim, S.H.Jeong, G.S.Lee, "Efficient Skew Estimation and Correction Algorithm for Document Images", *Image and Vision Computing*, pp 25-35, 2002.

[LEE 93] S.W.Lee, J.S.Park, Y.Y.Tang, "Performance Evaluation of Nonlinear Shape Normalization Methods for the Recognition of Large-Sat Handwritten Characters", *ICDAR, IEEE*, pp.402-407, 1993.

[LI 92] Y. Li. "Reforming the Theory of Invariant Moments for Pattern Recognition". *Pattern Recognition*, 25: pp 723–730, 1992.

[LIN 65] N. Lindgren, " Machine Recognition of Human Language: Part iii-Cursive Script Recognition" *IEEE Spectrum*, 2(5) :pp.104–116, 1965.

[LIU 97] Y.Liu, S.Srihari, "Document Image Binarization on Texture Features", *On Pattern Analysis and Machine Intelligence,* vol 19, Num 5, pp.540-544, Mai 1997.

[LOR 92] G.Lorette, Y.Lecourtier, "Reconnaissance et Interprétation de Texts Manuscripts Hors-line: Un Problème d'Analyse de Scène", Bigre Num 80-*CNED Colloque National sur l'Ecrit et le Document, Nancy, CNED,* Juillet 1992.

[LU 96] Y.Lu, M.Shridar, "Character Segmentation in Handwritten Words : An Overview", Pattern Recognition, vol 29, Num 1, pp.77-96, 1996.

[MAD 97] S. Madhvanath, V. Krpàsundar, "Pruning Large Lexicons Using Generalised Word Shape Descriptors", *ICDAR*, vol 2, pp. 552-555, 1997.

[MAH 94] A.S.mahmoud, "Arabic Character Recognition Using Fourier Descriptors and Character Contour Encoding", *Pattren Recognition*, vol 27, Num 6, pp 815-824, 1994.

[MAH 03] P.Mahé,"Noyaux pour Graphes et Support Vector Machines pour le Calibrage des Molécules", *Rapport de Stage*, 2003.

[MAL 97] Y. Mallet, D. Coomans, J. Kautsky, O. De Vel. "Classification Using Adaptive Wavelets for Feature Extraction", *IEEE Trans. on Pattern Analysis and Machine Recognition*, 19(10):1058–1066, 1997.

[MAK 01] G. Mak, The Implementation of Support Vector Machines Using The Sequential Minimal Optimization Algorithm, Master, McGill University, Montreal, Canada, 2001.

[MEH 05] Z.Mehennaoui, A.Benouareth, M.Sellami, "Reconnaissance des Caractères Arabes Manuscrits par Support Vector Machines », *Text Image and SpeechRecognition Workshop*, pp 80-87, Annaba, December, 2005.

[MEH a06] Z.Mehennaoui, A.Benouareth, M.Sellami, "Un Système à base des SVM pour la Reconnaissance d'images de lettres manuscrites", Conférence Internationale sur l'Informatique et ses Applications, CIIA 06, Saida, Algérie, 2006.

[MEH b06] Z.Mehennaoui, A.Benouareth, M.Sellami, "Utilisation des Machines à Vecteurs de Support dans la Reconnaissance Arabes Manuscrits", Conférence Internationale sur le Contrôle, la Modélisation et le Diagnostic, ICCMD'06, Annaba, Algérie, 2006.

[MIL 98] H.Miled, M.Cheriet, C.Olivier, Y.Lecourtier, "Modélisation Markovienne de l'écriture Arabe Manuscrite: une Approche Analytique", *Actes CIFED'98, 1er Colloque International francophone sur l'Ecrit et le Document,* pp.50-59, Québec, Canada, Mai 1998.

[MUL 01] K.R Muller, S. Mika, G. Ratsch, K. Tsuda,B. Scholkopf. "An introduction to kernel-based learning algorithms". *IEEE Transactions on Neural Networks*, 12 :pp.181-201, 2001.

[OLI 96] C.Oliver, H.Miled, K.Romeo, Y.Lecourtier, "Segmentation and Coding of Arabic Handwritten Words", *Proceedings of ICPR'96, 13th International Conference on Pattern Recognition,* vol 3, Track C, pp.264-268, October 1996.

[OSU 97] E. Osuna, R. Freund, F. Girosi. "Training Support Vector Machines: an Application to Face Detection", 1997.

[PET 90] J.C.Pettier, J.Camillerapp, "Reconnaissance statistique de mots manuscrits", *Reconnaissance Automatique de l'Ecrit*,pp 136-147, , Berge, Mai 1990.

[PLA 99] J.C. Platt. "Fast Training of Support Vector Machines Using Sequential Minimal Optimization". In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapitre 12. 1999.

[SAR 99] S.Sari, "Un Système de Reconnaissance de Mots Arabes Manuscrits Basé Segmentation", Thèse de Magister, Université de Annaba, 1999.

[SEH 00] A.Sehah, H.Hocini, M.Sehad, S.Ameur, "Analyse des Documents par la méthode des K plus proches voisins", CVA, Tizi-Ouzou, Algérie, 2000.

[SHE 99] D. Shen, H.H.S. Ip. "Discriminative Wavelet Shape Descriptors for Recognition of 2-D Patterns". *Pattern Recognition*, 32: pp.151–165, 1999.

[SHE 05] A.Sehad, L.Mezai, M.T.Laskr, M.Cheriet, "Méthode Rapide et Fiable pour la Détection de l'Angle d'Inclinaison des Documents imprimés par la Régression et la Transformée en Ondelettes" , *Text Image and Speech Recognition Workshop*, pp 211-217, Annaba, December, 2005.

[SMO 98] A. Smola, Learning with Kernels. PhD thesis, GMD First, Berlin, Germany, 1998.

[SNO 02] S.Snoussi Madouri, Modèle Perceptif Neuronal à Vision Globale-Locale pour la Reconnaissance des Mots Arabes Omni-Scripteurs, Thèse de Doctorat, Ecole National d'Ingénieurs de Tunis, 2002.

[SOU a97] L.Souici, T.Sari, Z.Zemirli, M.Sellami, "Prototype de Reconnaissance de Caractère Arabes Manuscrits à Base de Sous Réseaux Neuronaux", *Séminaire national d'Informatique,* Biskra, Novembre 1997.

[SOU b97] L.Souici, Z.Zemerli, M.Sellami, "Système Connexionniste pour la Reconnaissance de l'Arabe Manuscrits", *1ères Journées Scientifiques et Techniques, JST'97 FRANCIL*, pp.383-388, Avignon, France, 1997.

[SOU 06] L.Souci, Reconnaissance de Mots Arabes Manuscrits par Intégration Neuro-Symbolique, Thèse de Doctorat, Université de Annaba, 2006.

[TAP 90] C.C.Tappert, C.Suen, T.Wakahara, " The State of the Art in On-Line Handwriting Recognition ", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 12, N^0 8, pp: 787-808, 1990.

[TRI 95] O.D.Trier, T.Taxt, «Evaluation of Binarization Methods for Document Images", *On Pattern Analysis and Machine Intelligence,* vol 11, Num 12, pp. 312-314, Decembre, 1995.

[TRI2 96] O.D.Trier, A.K.Jain, T.Taxt, " Feature Extraction Methods Character Recognition: A Survey", *Pattern Recognition*, 29(4):pp. 641-662, 1996.

[VAN 94] R. J. Vanderbei. Interior point methods : Algorithms and formulations. *ORSA Journal of Computing*, 6(1) pp.32–34, 1994.

[VAP 82] V.Vapnik, *Estimation of Dependences Based on Data*, springer Verlag, Berlin, 1982.

[VAP 91] V. Vapnik, A. Chervonenkis "The Necessary and Sufficient Conditions for Consistency of the Method of Empirical Risk Minimisation", *Pattern recognition and Image Analysis*, 1(3), pp.284-305, 1991.

[VAP 95] V. Vapnik. The Nature of Statistical Learning Theory. Springer Verlag, New York, 1995.

[VAP 98] V.N.Vapnik, Statistical Learning Theorry, Wiley, New York, 1998.

[WAH 98] G. Wahba. Support vector machines, reproducing kernel hilbert spaces, gacv, 1998.

[WAK 97] T.Wakahara, K.Odaka, "Adaptative Normalization of Handwritten Characters Using Global/Local Affine Transformation", *ICDAR-IEEE*, pp 28-33, 1997.

[ZAH 90] A.Zahour, " Une Méthode de Reconnaissance de l'Ecriture Arabe Cursive", Thèse de Doctorat, 1990.