

N° d'ordre : 07/2017-D/INF

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université des Sciences et de la Technologie Houari Boumediene  
Faculté d'Electronique et d'Informatique



**THÈSE**  
Pour l'obtention du grade de Docteur En Sciences  
en : **INFORMATIQUE**  
Spécialité : Intelligence Artificielle et Bases de Données Avancées

Par :

**Zahia TAMEN**

**Vers une reconnaissance efficace  
du manuscrit Arabe**

Soutenue publiquement, le : 24/ 09 / 2017, devant le jury composé de :

<b>M. S. LARABI</b>	<b>Professeur</b>	à l' <b>USTHB</b>	<b>Président</b>
<b>Mme. H. DRIAS</b>	<b>Professeur</b>	à l' <b>USTHB</b>	<b>Directrice de Thèse</b>
<b>Mme. A. SERIR</b>	<b>Professeur</b>	à l' <b>USTHB</b>	<b>Examineur</b>
<b>M. M. E. Y. BOUDAREN</b>	<b>MCA</b>	à l' <b>EMP</b>	<b>Examineur</b>
<b>M. M. DIAF</b>	<b>Professeur</b>	à l' <b>UMMTO</b>	<b>Examineur</b>
<b>M. A. DJEFFAL</b>	<b>MCA</b>	à l' <b>UMKB</b>	<b>Examineur</b>



Université des Sciences et de la Technologie Houari Boumediene  
Faculté d'Electronique et Informatique

---

# Vers une reconnaissance efficace du manuscrit Arabe

---

Zahia TAMEN

Sous la direction du Pr. H. Drias

## ملخص

تقدم هذه الوثيقة حوصلة لنتائج سنوات من العمل في مجال التعرف على الكتابة اليدوية العربية وهي تتعلق بتصميم وتنفيذ نظام التعرف على الكتابة اليدوية العربية وتولي تركيزاً معيناً للوحدات الأساسية الثلاث التي تشكل النظام وهي التجزئة، واستخراج الميزات والتصنيف. أثناء عملية التجزئة، تم تطبيق خوارزميتين اثنتين. الأولى متعلقة بالتعامل مع الكتابة المقيدة و توفر نقاط تجزئة داخل منطقة وسيطة من المقاطع ذات حد أدنى من الوحدات المضيئة. الخوارزمية الثانية، تتعلق بالكتابة الحرة و تمر من خلال عملية تنحيف تتلوها مرحلة كشف وتعديل للخط الأساس للكتابة. يتم اختيار نقاط التجزئة على طول هذا الخط في المناطق ذات ارتفاع بكسل واحد. في كلتا الحالتين نقترح إمكانية وصل أجزاء مجزأة لمحاولة استعادة الأحرف التي تم الإفراط في تجزئتها. نظراً لعدم وجود قاعدة بيانات أحرف مجزأة من قبل اضطررنا لإنشاء واحدة. الميزات المستخرجة من صور الكلمات أو الأحرف هي من النوع الإحصائي والهيكلية. وتشمل الخصائص الإحصائية الإسقاطات المتعامدة على أساس كثير الحدود، وتحويل المويجات، والتحويل تجب العددي والتحليل الكسورية. إنها تتعلق كذلك بتوزيع الوحدات المضيئة على الصورة ككثافة المساحة المكتوبة، وعدد الإنتقالات أسود / أبيض على مختلف الاتجاهات الخ. تركز السمات الهيكلية على المعلومات الطوبوغرافية للكتابة بالتحديد، الدوائر، المقعرات، والتضاعفات و التنازليات الخ. عملية التصنيف ينطوي على استخدام مختلف المصنفات مثل الشبكات العصبية، نماذج ماركوف المخفية، آلة التعلم الأقصى، آلة بناقلات الدعم.

ونقترح طريقة للجمع بين نتائج المصنفات من أجل تحسين النتائج التي تحصلت عليها المصنفات المختلفة بشكل فردي.

## Résumé

Ce document synthétise les résultats de plusieurs années de travail dans le domaine de la reconnaissance du manuscrit arabe. Il touche à la conception et à la réalisation d'un système de reconnaissance de l'écriture manuscrite arabe. Une attention particulière est accordée aux modules de base constituant le système, à savoir, les prétraitements, la segmentation, l'extraction de caractéristiques et la classification.

Durant le processus de segmentation, deux algorithmes ont été mis en œuvre. Le premier, destiné à l'écriture contrainte propose des points de segmentation le long de la ligne de base d'écriture avec comme critère la zone médiane de sections avec le minimum de pixels allumés sur les colonnes. Le deuxième algorithme, englobant l'écriture libre passe par un processus de squelettisation suivi de la détection et redressement de la ligne de base d'écriture. Le choix des points de segmentation est fait le long de cette ligne sur les zones à un pixel de hauteur. Une possibilité de concaténation des parties segmentées est proposée dans les deux cas pour essayer de reconstituer les caractères éventuellement sur-segmentés. L'absence d'une base de données de caractères pré segmentés nous a amenée à en créer une.

Les caractéristiques extraites des images de mots ou de caractères vont du type statistique au type structurel. Les caractéristiques statistiques englobent les moments orthogonaux, les transformées en ondelettes, la DCT et l'analyse fractale. Elles concernent aussi la répartition des pixels allumés sur l'image comme la densité de la surface écrite, le nombre de transitions noir/blanc sur les différentes directions etc. Les caractéristiques structurelles se focalisent sur la donnée d'informations topologiques de l'écriture à savoir, les boucles, les concavités, les ascendants et descendants etc.

Le processus de classification passe par l'utilisation de différents classifieurs tels que les réseaux de neurones, les SVM, les ELM et les HMM. Un processus de combinaison des résultats des classifieurs est proposé dans le but d'améliorer les résultats obtenus par les différents classifieurs individuellement.

**Mots-clés :**

reconnaissance du manuscrit arabe, moments orthogonaux, DCT, filtres de Gabor, IFN/ENIT, AHDB, IESK-arDB, réseaux de neurones, SVM, ELM, HMM

## Abstract

This document synthesizes the results of years of work in the field of Arabic handwriting recognition. It concerns the design and implementation of a system dedicated to Arabic handwriting recognition. Particular attention is given to three basic modules constituting the system, namely, segmentation, feature extraction and classification.

During the segmentation process, two algorithms have been implemented. The first dealing with constraint writing offers segmentation points within the median zone of sections with a minimum of illuminated pixels on the columns. The second algorithm, dealing with free writing passes through a skeletonization process followed by detection and adjustment of the writing baseline. The choice of segmentation points is made along this line in areas with one pixel height. A possibility of concatenating the segmented portions is proposed in both cases to try to restore the characters that are over-segmented. The absence of a pre segmented characters database has led us to create one.

The features extracted from the images of words or characters are of statistical and structural types. The statistical characteristics include orthogonal moments (OM), the wavelet transform, the DCT and fractal analysis. They also relate to the distribution of the foreground pixels on the image as the density of the written area, the number of transitions black/white on the different directions etc. The structural features focus on the data of topological information of the writing namely, holes, concave, ascending and descending etc.

The classification process involves the use of different classifiers such as neural networks, SVM, ELM and HMM. A method of combination of classifier results is proposed in order to improve the results obtained by different classifiers individually.

### Keywords

Arabic Handwriting recognition, orthogonal moments, DCT, Gabor filters, IFN/ENIT, AHDB, IESK-arDB, neural networks, SVM, ELM, HMM

Je tiens à remercier vivement Mme le Professeur Habiba Drias pour m'avoir supportée durant toutes ces années.

Je remercie les membres de jury qui m'ont fait l'honneur de juger ce travail.

Mes vifs remerciements vont au Professeur Dalila Boughaci pour sa précieuse aide.

À tous ceux qui m'ont aidée, qui ont essayé de m'aider et à tous ceux qui m'ont soutenue, Merci.

Je dédie ce modeste travail :

À mes très chers parents. Que Dieu les garde et les protège.  
À mon cher mari. Que Dieu le garde et le protège. À mes frères et mes  
sœurs. Que Dieu les garde et les protège. À mes très chers enfants. Que  
Dieu les garde et les protège. À tous ceux que j'aime et qui m'aiment.  
Que Dieu les garde et les protège.

À la mémoire de mon amie et collègue : Ahlam Benchennaf. Que Dieu  
lui accorde son pardon et l'accueille dans son vaste paradis.

# Sommaire

# Introduction

La parole et l'écriture manuscrite sont parmi les moyens essentiels de la communication entre les personnes humaines. La communication entre l'homme et la machine avec ces deux moyens exige de réaliser des systèmes pour faciliter cette interaction. Malgré les prédictions faites sur sa disparition à long terme, l'écriture est un moyen de communication encore essentiel et apprécié de nos jours. Elle peut être sous différents styles, imprimée ou manuscrite. Ces styles d'écriture sont différents aussi bien par le support sur lequel on trace les signes graphiques (pierre, papier, tablette digitale, ordinateur, etc.) et le moyen utilisé pour les tracer (poinçon, stylet, stylo, plume, clavier, machine à écrire, etc.). Même si l'imprimé prend une part grandissante dans diverses applications pour des raisons de lisibilité, de facilité de stockage et sécurité, facilité et rapidité de communication, le manuscrit reste encore très convoité que ce soit pour sa production naturelle qui nous a été enseignée, ou pour son utilité dans le diagnostic de certaines maladies (Parkinson, dyslexie, dyspraxie) ou d'autres tâches comme en psychologie expérimentale, dans l'éducation, l'expression des sentiments, etc.

Le clavier et la souris n'ont pas pu devenir comme on l'avait prévu longtemps auparavant, les moyens de communication naturels entre l'humain et l'ordinateur. Avec le nombre croissant d'utilisateurs de ce dernier, le manuscrit demande de plus en plus d'attention de la part des chercheurs dans la conception d'interfaces homme-machine conviviales. Les systèmes de reconnaissance du manuscrit peuvent contribuer énormément au développement de processus automatiques et peuvent amé-



Le but de ce travail, a été de concevoir et de réaliser un système efficace pour la reconnaissance du manuscrit arabe. Pour ce faire, un ensemble varié de techniques de modélisation a été utilisé, englobant les deux grands aspects concernant les types de reconnaissance à savoir :

1. la reconnaissance globale utilisant l'analyse en ondelettes, l'analyse fractale ou même les moments orthogonaux pour décrire les entités à reconnaître qui sont à ce moment-là les mots.
2. la segmentation en caractères avec une description topologique de ces derniers.

La première approche est applicable lorsque le dictionnaire ou le lexique utilisé est de taille relativement petite. Elle s'adapte parfaitement aux problèmes d'interaction homme-machine puisqu'il est prouvé que dans ce cas, la taille ne dépasse généralement pas quelques centaines de mots. En effet, on s'imagine mal, avec l'état actuel des choses, en train de discuter avec une machine de nos sentiments ou de nos idées politiques par exemple. Dans cette approche, on utilise souvent des descripteurs globaux qui décrivent l'image du mot en entier. Parmi ces descripteurs, nous pouvons citer :

1. Les moments géométriques ou les moments orthogonaux ;
2. Les transformations telles que :
  - (a) La transformée de Fourier,
  - (b) L'analyse en ondelettes,
  - (c) L'analyse fractale,
  - (d) La DCT, la SVD, les Curvelets etc.

Il est aussi souvent nécessaire d'appliquer ces mêmes descripteurs dans des zones d'intérêt de l'image : il s'agira à ce moment-là de la technique de « zonage » ou « fenêtrage ». La technique de zonage (zoning method) permet de focaliser l'attention sur des zones particulières en utilisant par exemple des fenêtres glissant à travers l'image. Ces zones peuvent être de simples fenêtres rectangulaires souvent de même hauteur que l'image, se déplaçant horizontalement sur l'image, avec ou sans

chevauchement. Elles peuvent être sous formes de blocs se déplaçant horizontalement et verticalement sur l'image comme elles peuvent être circulaires, concentriques sur le centre de gravité de l'image ou se déplaçant le long de la ligne d'écriture suivant le sens de lecture. Cette technique permet d'extraire des informations locales de l'image sans avoir à faire une vraie segmentation. Ces zones (fenêtres) peuvent être de dimensions et de formes quelconques. Celles que nous avons utilisées dans ce travail concernent les fenêtres glissant de droite à gauches sur les mots ayant :

1. La hauteur de l'image avec une forme rectangulaire ;
2. Une forme circulaire se déplaçant le long de la ligne de base d'écriture. Ces fenêtres peuvent aussi subir des subdivisions en blocs pour récolter plus d'informations et aussi pour permettre d'enlever l'ambiguïté entre certains caractères proches en forme globale et différents dans les détails (voir figure ??).

La deuxième approche, basée sur la segmentation est nécessaire lorsque l'on s'attaque à un vocabulaire ouvert. La segmentation permet de réduire les entités à reconnaître aux graphèmes issus de la segmentation qu'ils soient constitués de :

1. Caractères entiers dans le cas idéal,
2. Ou de parties contenant :
  - (a) Plus d'un caractère et dans ce cas on parle de sous-segmentation. Cette situation se présente en général lorsque l'on a affaire à des ligatures particulières du script telles que la composition « lam-alif » qui donne naissance à une entité impossible à segmenter (voir figure ??). Dans le cas du manuscrit, ces combinaisons peuvent se multiplier et se diversifier puisque l'être humain peut se donner le plaisir d'écrire dans les styles de son choix. La figure ?? montre un même mot écrit avec deux styles différents.
  - (b) Le deuxième cas concerne la sur-segmentation et à ce moment-là, la partie segmentée contient moins d'un caractère. Il s'agit d'un problème inévitable sans l'utilisation de connaissances a priori sur le script et découle de la segmentation de caractères comme « sin », « sad » etc.



FIGURE 2 – Deux mots proches en forme globale et différents en contenu

محمد → ح + م ; لحم → ح + ل ; لماذا → م + ل ; لازال → ا + ل

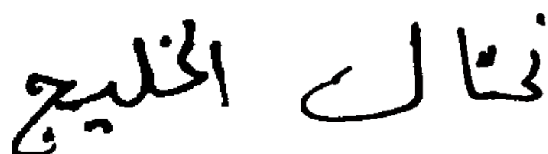


FIGURE 3 – Quelques combinaisons de caractères impossibles à segmenter, en haut cas de l'imprimé ; en bas, cas du manuscrit

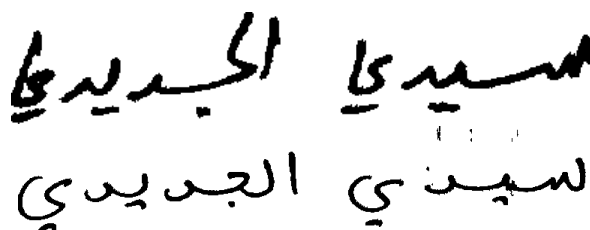


FIGURE 4 – Un même mot écrit de deux manières différentes

Nous allons décrire dans ce document, l'essentiel de notre contribution dans le domaine qui a touché aux deux aspects de la reconnaissance.

Dans la première approche, l'essentiel de notre contribution peut être synthétisée en ces points :

1. D'abord, nous avons implémenté la plupart des techniques globales disponibles en termes de transformations, de moments, d'analyse fractale, avec les filtres de Gabor etc. Nous avons ensuite testé plusieurs combinaisons et hybridations pour améliorer les performances du système.
2. Pour les caractéristiques analytiques, nous avons utilisé une panoplie des techniques existantes et nous avons mis au point deux approches propres à nous qui sont :
  - une technique d'extraction de caractéristiques basées sur le contour de l'écriture et les statistiques,

- des variantes utilisant les fenêtres glissantes et la technique de zonage.
3. Beaucoup de caractéristiques prises dans ce travail, se basent sur la détection et le redressement de la ligne de base d'écriture. De ce fait, plusieurs techniques de détection ont été implémentées et parallélisées en vue de les optimiser en termes de temps d'extraction et d'efficacité.

La seconde approche a été touchée selon ses deux aspects :

1. Sans squelettisation de l'écriture donnant naissance à un algorithme de segmentation suivant un certain nombre de règles.
2. Avec squelettisation, et là nous avons implémenté les algorithmes les plus connus dans le domaine avec leur parallélisation. Dans ce cadre, nous avons proposé une amélioration de l'algorithme de squelettisation qui a été le plus efficace dans le domaine en proposant une étape supplémentaire de nettoyage des restes inévitables de ce dernier. Une étude comparative des résultats obtenus est faite en utilisant les métriques usuelles connues dans le domaine.
3. Pour pouvoir assurer un apprentissage du système basé segmentation (selon les deux aspects suscités) et en l'absence d'une base de données gratuite disponible pour la recherche, on s'est vue obligée d'en réaliser une qu'on a nommée TBK.

Plusieurs techniques de classification ont été testées sur les deux approches allant des réseaux de neurones de type perceptron multicouches aux SVM (Support Vector machines), aux ELM (Extreme Learning Machine). Les réseaux LVQ (Learning Vector Quantization) et les réseaux RBF (Radial Basis Function) ont aussi été utilisés. Plusieurs techniques de combinaison des classifieurs en vue de l'amélioration des performances du système ont été testées et ont donné naissance à une méthode de combinaison qui a concurrencé les résultats de l'état.

Le reste de la thèse s'articulera comme suit :

- Le premier chapitre va être consacré à l'étude de l'état de l'art. Dans ce chapitre, on mettra l'accent sur les difficultés innées du traitement

du manuscrit en général, puis on détaillera les particularités du manuscrit arabe qui en font un script plus compliqué à traiter que le Latin. Dans ce même chapitre, on décrira les quelques bases de données disponibles pour le manuscrit arabe et le problème qui nous a mené à en créer une sur les caractères arabes manuscrits pré segmentés. On terminera ce chapitre par les résultats les plus récents atteints dans la reconnaissance du manuscrit arabe.

- Le second chapitre sera dédié aux différentes composantes d'un système de reconnaissance du manuscrit arabe à savoir :
  - Prétraitements ;
  - Extraction de caractéristiques ;
  - Classification.
- Les trois chapitres qui suivent seront consacrés à l'essentiel de notre contribution dans le domaine :
  - Le chapitre trois décrit ce que nous avons apporté au module de prétraitements que ce soit pour :
    - la détection et le redressement la ligne de base d'écriture,
    - la squelettisation du manuscrit arabe.
  - Le module d'extraction de caractéristiques est pris en charge par le quatrième chapitre. On y décrit les différentes caractéristiques utilisées et comment elles ont été testées et combinées. Un type de caractéristiques de notre propre design y est aussi décrit. C'est aussi dans ce chapitre que notre contribution concernant les techniques de classification est décrite. Les classifieurs utilisés ainsi que les méthodes de combinaison de plusieurs classifieurs pour améliorer les performances individuelles de ces derniers y sont explicitées.
  - Le cinquième chapitre est dédié à notre apport dans la partie segmentation du manuscrit arabe. Il décrit la segmentation de l'écriture contrainte et non contrainte et explicite les différentes techniques de concaténation des parties segmentées pour remédier aux problèmes inévitables de sur-segmentation.
- Le sixième et dernier chapitre est consacré aux résultats des différentes expérimentations entreprises explicitant les différentes configu-

rations « caractéristiques-classifieurs » utilisées. Dans ce cadre, une distinction est faite pour séparer les deux approches utilisées dans notre travail à savoir, l'approche globale et celle basée sur la segmentation en caractères. Dans l'approche globale, une combinaison de plusieurs classifieurs est proposée améliorant sensiblement les résultats des classifieurs individuels et notamment ceux de l'état de l'art actuel. Quelques résultats obtenus concernant la segmentation y sont aussi donnés.

- Enfin, une conclusion générale terminera ce document proposant quelques perspectives à ce travail.
- Toujours dans les perspectives de ce travail, la nouvelle approche sur laquelle nous travaillons et qui concerne la normalisation de l'écriture arabe manuscrite avant son traitement est aussi succinctement décrite.

# Partie I : Entrée introductive au domaine de reconnaissance du manuscrit arabe

Cette partie constituera une entrée dans le domaine de la reconnaissance du manuscrit arabe. Elle est subdivisée en deux chapitres comme suit.

Dans le premier chapitre :

- Il s'agira, en premier lieu de définir les caractéristiques spécifiques à l'écriture manuscrite et la manière dont on peut approcher son traitement automatique.
- Le manuscrit arabe et les difficultés qui lui sont liées sont ensuite explicités.
- Les différentes bases de données utilisées dans le traitement du manuscrit arabe y sont aussi décrites succinctement.

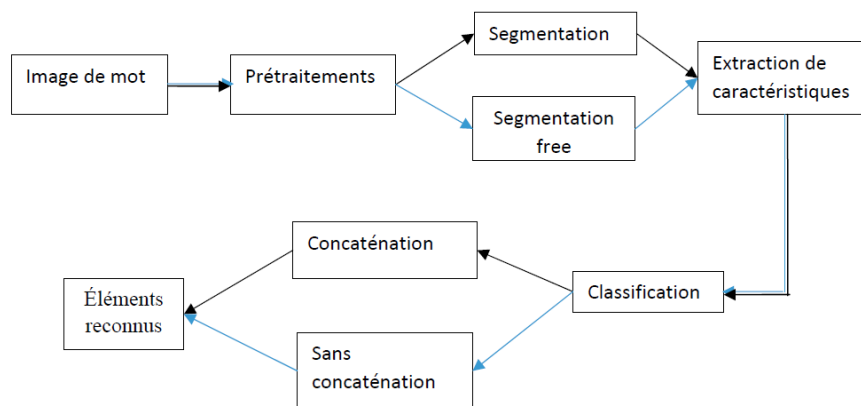
Le second chapitre décrira brièvement les modules constituant un système de reconnaissance du manuscrit arabe. Il détaillera surtout les modules :

- Prétraitements.
- Extraction de caractéristiques.
- Classification.

# Reconnaissance de l'écriture arabe manuscrite : Aperçu et État de l'art

## 1.1 Introduction

Avec la parole, le manuscrit reste encore de nos jours l'un des deux moyens de communication homme-machine les plus désirés parce que les plus naturels. Bien que le manuscrit exige une certaine connaissance intellectuelle, il est encore le moyen le plus convoité vu qu'il permet contrairement à la parole, le plus de discrétion. De plus, le patrimoine humain qui est stocké dans des documents manuscrits est énorme. De ce fait, réaliser des systèmes capables de transformer ces trésors en des documents numérisés faciles à manipuler et permettant une recherche aisée dans ces derniers, est d'une importance énorme. En fait, la reconnaissance de l'écriture manuscrite présente de gros problèmes même pour l'être humain. Il est bien connu que, contrairement à l'imprimé, le manuscrit présente le problème de styles d'écriture, d'humeur du scripteur, du niveau intellectuel de ce dernier etc. En fait, le caractère subjectif de l'écriture manuscrite résulte du fait que le premier lecteur est souvent le scripteur lui-même. Il est fréquent que ce dernier ne se rende pas compte à quel point son écriture peut-être illisible. Une limite à prendre en considération pour ce problème est que si l'être humain ne peut pas lire un texte, on ne peut logiquement pas le demander à un ordinateur. Ceci permet de fixer une limite basse admissible à la qualité de l'écriture proposée comme entrée d'un système de reconnaissance



Deux types de traitements du manuscrit : chemin noir, avec segmentation et sans segmentation chemin bleu.

FIGURE 1.1 – Schéma Bloc d'un système de reconnaissance de l'écriture

[Dargenton1994].

Nous allons dans ce qui suit faire une entrée dans le domaine de la reconnaissance du manuscrit en général et le manuscrit arabe en particulier. Nous parlerons des particularités qui font de l'arabe, un script pas facile à traiter automatiquement. Nous terminerons par donner un aperçu de quelques bases de données disponibles pour le but de la validation et la comparaison des travaux de recherche dans le domaine.

## 1.2 Système de reconnaissance du manuscrit

Un schéma global de la reconnaissance de l'écriture manuscrite peut-être illustré en figure ???. Cette figure illustre les deux chemins possibles pour tout travail de reconnaissance du manuscrit. En effet, la reconnaissance de l'écriture peut-être classifiée en ces différentes catégories [Amin1998] :

- L'approche holistique ou globale qui utilise des primitives extraites des images de mots dans le but de reconnaître le mot entier. Il est généralement admis que cette approche globale est acceptable lorsque le vocabulaire à apprendre est réduit. Ceci est souvent le cas dans les applications interactives homme-machine.

- L'approche analytique segmente le mot en des graphèmes particuliers typiquement des caractères. La segmentation en caractères avant le processus de reconnaissance s'appelle segmentation en caractères externes. La segmentation interne en caractères se produit en concurrence avec le processus de reconnaissance.
- Appariement de séquences de primitives : Dans cette approche, on extrait de façon séquentielle des primitives et on déduit l'identité du mot à partir de cette séquence. Il s'agit de la reconnaissance statistique des formes où les HMM (modèles de Markov cachés) ont été largement utilisés. Il est à noter que la reconnaissance à base de HMM est aussi souvent classifiée dans la catégorie "approche globale".

Le chemin en bleu dans le schéma illustre les méthodes globales qui consistent à considérer le mot comme unité de base à reconnaître. Celui qui contient des bifurcations noires délimitent les méthodes de reconnaissance basées-segmentation. La concaténation fait référence au "collage" éventuel des parties segmentées dans le but de reconstituer les caractères qui auraient subi une sur-segmentation.

### 1.2.1 Difficultés liées aux caractéristiques du manuscrit

Contrairement à l'imprimé, où les caractères respectent des règles d'écriture strictes, le manuscrit présente des difficultés dues au fait que le scripteur se voit avoir beaucoup de libertés :

- Il peut, selon son humeur et son entrain, écrire de façon :
  - artistique, allant jusqu'à dessiner son manuscrit ne respectant aucune règle d'écriture.
  - gribouillée, rendant l'écriture illisible même pour l'être humain.
  - soignée ou même très soignée, ce qui représente le meilleur des cas mais qui n'est pas acquis d'office.
- Il peut s'appliquer à ne bien tracer que les lettres permettant de distinguer les mots. Cette attitude a une conséquence importante sur la stratégie de lecture à adopter pour la reconnaissance. En effet, elle implique d'avoir la connaissance du vocabulaire parmi lequel l'auteur a estimé pouvoir distinguer entre les mots [**Dargent**1994].
- La variabilité des styles d'écriture est telle que la connaissance du

vocabulaire, et les connaissances a priori du domaine sont essentielles pour guider le déchiffrement de l'écriture, même pour un humain [Menasri2008].

- Nous ajoutons à cela, le fait que, alors que pour un être humain, donner son appréciation sur la qualité de l'écriture qui lui est présentée est une chose aisée, faire calculer à la machine un degré de lisibilité de l'écriture est dans le domaine de la fiction encore à l'état actuel.

### 1.2.2 Catégories de la reconnaissance du manuscrit

On peut aussi classer la reconnaissance automatique du manuscrit suivant la manière dont la donnée est présentée au système. Les approches de reconnaissance hors ligne du manuscrit ne nécessitent pas d'interaction avec l'utilisateur. Une image du texte manuscrit est scannée et est présentée au système sous le format d'une image digitalisée. Dans le cas de la reconnaissance en ligne, on utilise un stylo spécial pour écrire sur un dispositif digital tel qu'une tablette digitale possédant une résolution en dpi (dots per inch) et un échantillonnage en points par seconde. Les échantillons numérisés sont envoyés au système en tant que séquence de points (2D) en temps réel, ainsi le suivi temporel des données est une donnée supplémentaire non présente dans l'entrée hors ligne [Amin1998, BenCheikhetal2008, Khorsheed2002, ParvezMahmoud2013]. La reconnaissance en ligne présente plusieurs caractéristiques intéressantes. D'abord, la reconnaissance est effectuée sur des données unidimensionnelles plutôt que sur des images 2D comme dans la reconnaissance hors ligne. L'écriture est représentée par une séquence de points dont la localisation est fonction du temps. Ceci présente plusieurs conséquences importantes [Amin1998, Khorsheed2002] :

- L'ordre d'écriture est disponible et peut être utilisé par le processus de reconnaissance.
- La ligne d'écriture n'a pas d'épaisseur.
- Des informations temporelles telles que la vitesse d'écriture peuvent être prises en considération.
- Les levers de stylo peuvent aussi être utiles dans le processus de reconnaissance.

La reconnaissance hors ligne est de ce fait, considérée comme plus difficile que la reconnaissance en ligne. Dans le cas en ligne, les caractéristiques peuvent être extraites et de la trajectoire du stylo et de l'image résultante alors que dans le cas hors ligne, seule l'image est disponible [Amin1998, Khorsheed2002].

### 1.3 Le manuscrit arabe

La langue arabe est parlée par plus de quatre cent millions de personnes à travers le monde. Elle est lue par encore plus de monde en tant que langue choisie par le Créateur pour le Coran. La lecture du Coran est un rituel quotidien demandé à tous les Musulmans.

La langue arabe a une forme standard appelée Arabe Standard Moderne qui est utilisée comme moyen de transcription de plusieurs langues telles que le Persan, le Kurde, le Malay et le Urdu. Pour toutes ces raisons, réaliser un système qui reconnaît un tel script est d'une grande importance [AlKhateebetal2009, Amin1998, ParvezMahmoud2013].

#### 1.3.1 Particularités de l'écriture arabe et difficultés supplémentaires

Le script arabe s'écrit de droite à gauche. En arabe, les ligatures se situent toujours au niveau de la ligne d'écriture, c'est-à-dire qu'il n'existe pas de lettre à liaison haute comme le 'o' ou le 'v' dans l'alphabet latin. Mais dès qu'il s'agit du manuscrit, une personne peut très bien se voir opérer des ligatures verticales comme un style spécial d'écriture (Figure ??).

Cette ligne fictive d'écriture est appelée ligne de base et elle peut très souvent se trouver non horizontale nécessitant ainsi une rectification qu'on appelle redressement (Figure ??).

L'alphabet arabe comporte 28 lettres (voir le tableau ??). Certaines des lettres arabes ne peuvent pas être liées aux suivantes rendant ainsi le script arabe semi-cursif même dans sa version imprimée. Ces lettres sont : ر, د, و, ا, ذ and ز.

Nom caractère	Position début	Position milieu	Position finale	Position isolée
Alif	ا	ا	ا	ا
Ba	ب	ب	ب	ب
Ta	ت	ت	ت	ت
Tha	ث	ث	ث	ث
Djim	ج	ج	ج	ج
Haa	ح	ح	ح	ح
Kha	خ	خ	خ	خ
Dal	د	د	د	د
Dhal	ذ	ذ	ذ	ذ
Ra	ر	ر	ر	ر
Zey	ز	ز	ز	ز
Ta	ط	ط	ط	ط
Dha	ظ	ظ	ظ	ظ
Sad	ص	ص	ص	ص
Dhad	ض	ض	ض	ض
Sin	س	س	س	س
Chin	ش	ش	ش	ش
Ain	ع	ع	ع	ع
Ghain	غ	غ	غ	غ
Fa	ف	ف	ف	ف
Kaf	ك	ك	ك	ك
Kaf	ق	ق	ق	ق
Lam	ل	ل	ل	ل
Mim	م	م	م	م
Noon	ن	ن	ن	ن
Ha	ه	ه	ه	ه
Waw	و	و	و	و
Ya	ي	ي	ي	ي

FIGURE 1.2 – Alphabet arabe

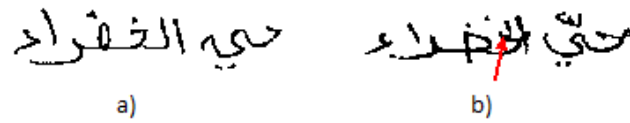


FIGURE 1.3 – Un mot de la base IFN/ENIT a) sans ligature verticale b) avec ligature verticale

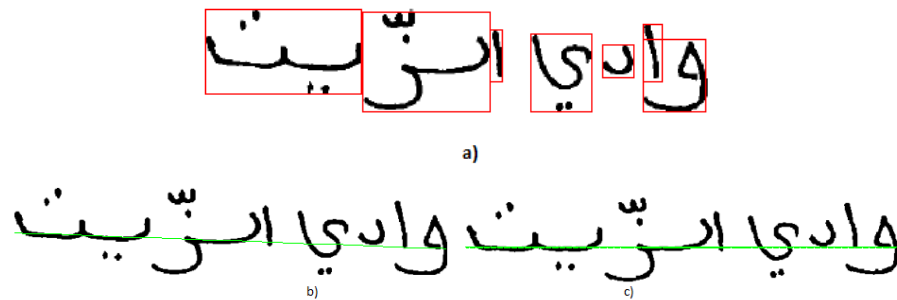


FIGURE 1.4 – a) Deux mots constitués de quatre PAW pour le premier à partir de la droite et trois pour le second b) ligne de base oblique c) ligne de base redressée

Ces dernières font qu'un mot arabe peut être constitué de plusieurs parties connexes appelées PAWs pour (Pieces of Arabic Words). Un exemple est donné en figure ??.

Quelques lettres ont des formes différentes d'écriture selon leurs positions dans la PAW : au début, au milieu ou à la fin de cette dernière ou dans sa forme isolée. La figure ?? montre une exemple de la lettre 'ع' écrite selon quatre différentes formes.

Quand on traite le manuscrit, on est confronté à une multitude de

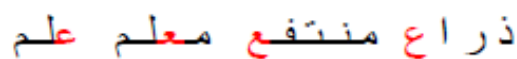


FIGURE 1.5 – Le caractère 'aine' dans les quatre positions possibles dans une PAW (début, fin, milieu, et isolée)

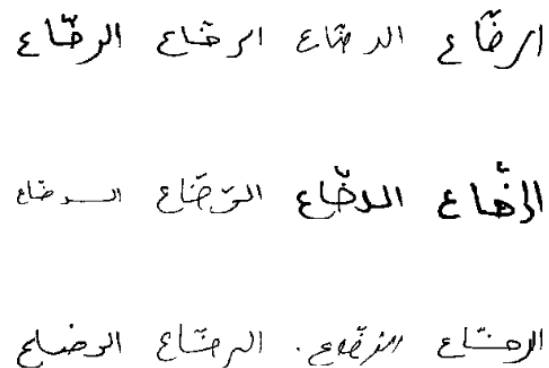


FIGURE 1.6 – Un mot écrit par douze scripteurs



FIGURE 1.7 – Les voyelles en arabe

façons d'écrire le même mot même quand il s'agit de la même personne. Dès qu'on parle de multi-scripteurs, cette multitude frôle l'innombrable. La figure ?? montre un même mot écrit par douze scripteurs différents montrant la variation extrême entre les différentes transcriptions de ce dernier.

Le terme 'signe diacritique' peut porter à confusion : dans certains travaux, seules les voyelles (voir figure ??) sont appelées diacritiques. Dans d'autres travaux, en revanche, tous les signes secondaires sont appelés diacritiques, qu'il s'agisse des voyelles, des points ou des autres signes (chadda, madda, hamza, etc.).

### 1.3.2 Historique et État de l'art

La recherche dans la reconnaissance du manuscrit arabe a débuté par les premiers travaux entrepris par A. Amin. Depuis, beaucoup de tentatives significatives lui ont succédé dans le but d'améliorer les résultats obtenus dans les processus de segmentation et de reconnaissance du manuscrit arabe. Seulement, malgré la disponibilité de plusieurs OCR dédiés à l'imprimé arabe, et à part quelques prototypes de recherche développés pour prouver des concepts théoriques, à notre connaissance, aucun système opérationnel n'est commercialisé jusqu'à présent pour le manuscrit arabe. Nous concentrerons l'examen de l'état de l'art sur trois étapes importantes dans un système de reconnaissance, à savoir les prétraitements, l'extraction de caractéristiques et la classification où nous parlerons de l'historique de la combinaison de classifieurs dans le domaine de la reconnaissance du manuscrit arabe.

#### Prétraitements

Nous limiterons l'étude de l'état de l'art concernant les prétraitements aux deux axes principaux qui sont :

1. La détection et le redressement de la ligne de base d'écriture.
2. La squelettisation.

**Détection et redressement de la ligne de base** Dans le but d'estimer la ligne de base d'écriture de mots de la base IFN/ENIT, les auteurs dans [PechwitzMargner2002], commencent par définir une approximation polygonale du squelette des mots. Les points diacritiques sont ensuite soustraits de l'image traitée et une première estimation de la ligne de base est effectuée sur les parties connexes restantes. L'étape finale est une analyse de régression des points pertinents autour de la zone de cette première estimation pour trouver la position effective de la ligne de base. La figure ?? tirée de [PechwitzMargner2002] résume les résultats de l'approximation polygonale d'un mot de la base IFN/ENIT tandis que la figure ?? montre les résultats de détection de la ligne de base pour le même mot.

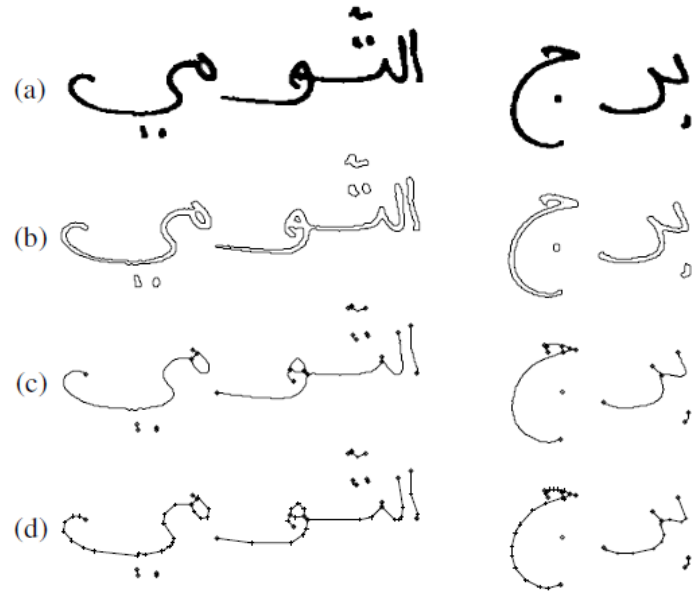


FIGURE 1.8 – Approximation polygonale d'un mot de la base IFN/ENIT (image tirée de [PechwitzMargner2002])

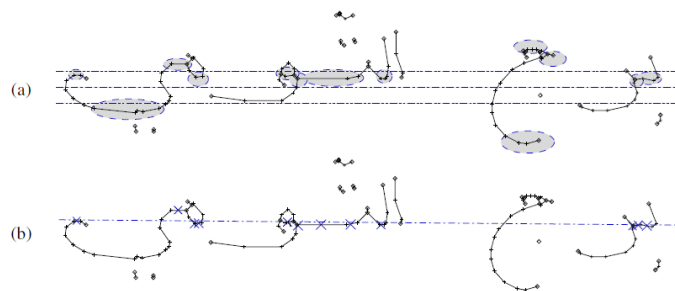


FIGURE 1.9 – Détection de la ligne de base d'un mot de la base IFN/ENIT selon [PechwitzMargner2002]. Les points considérés pour la régression linéaire sont ceux marqués de croix. L'image est tirée de [PechwitzMargner2002]

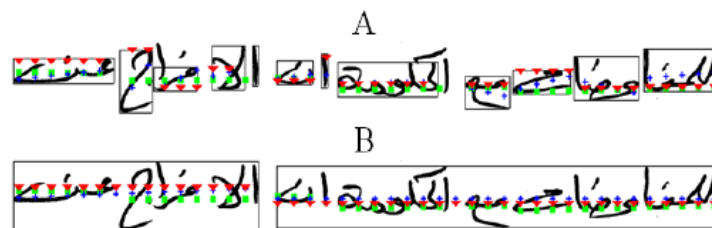


FIGURE 1.10 – Estimation de la ligne de base suivant les formules ?? et ?? de [Natarajanetal2011]. Les triangles, les croix et les carrés représentent les méthodes de projection, basée-centroïde et l'estimation humaine respectivement.

Dans [Natarajanetal2011], les auteurs font une première estimation de la ligne de base sur une ligne de texte complète. Ensuite, ils affinent cette estimation en recalculant la ligne de base pour chaque composante connexe de la ligne de texte. Pour éviter les problèmes engendrés par les points et divers diacritiques lors de la détection de la ligne de base, ils procèdent à l'élimination temporaire de toute partie connexe  $c_1$  s'il existe une autre partie  $c_2$  telle que :

$$c_1.l \geq c_2.l \text{ et } c_1.r \leq c_2.r \text{ et } c_1.pts \leq c_2.pts \quad (1.1)$$

où  $c.l$  et  $c.r$  sont les position horizontales extrêmes gauche et droite.  $c.pts$  représente le nombre de points de la partie connexe.

Pour palier aux cas où les diacritiques ne seraient pas obligatoirement écrits aux bons emplacements, les auteurs utilisent un seuil pour fusionner des parties connexes dont les positions vérifient :

$$\frac{\min(c_1.r, c_2.r) - \max(c_1.l, c_2.l) + 1}{\max(\text{width}(c_1), \text{width}(c_2))} \geq O_{Th} \quad (1.2)$$

Les valeurs positives pour  $O_{Th}$  permettent de fusionner uniquement les composantes qui chevauchent horizontalement, tandis que les valeurs négatives fusionnent toutes les composantes proches. La figure ?? tirée de [Natarajanetal2011] montre les résultats obtenus par les auteurs en utilisant les méthodes de projection, et d'estimation des centroïdes verticaux des parties connexes.

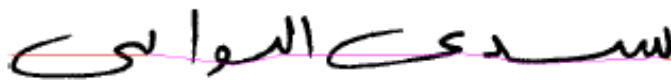


FIGURE 1.11 – Estimation de la ligne de base suivant [Menasrietal2008].

Dans [Snoussi<sup>etal</sup>2008], les auteurs implémentent six méthodes de détection de la ligne de base d'écriture évaluées sur la base IFN/ENIT. Quatre d'entre elles sont les plus connues dans le domaine à savoir, la méthode de projection d'histogramme, la méthode de minimisation d'entropie, la méthode de Hough et celle de la projection d'histogramme appliquée au squelette. Les auteurs ont développé deux méthodes hybrides de la méthode de projection qu'ils ont appelées méthode Min-Max et méthode de combinaison de la méthode Min-Max avec la position des primitives structurales. Nous avons implémenté les quatre méthodes communes et nous avons adapté les deux autres méthodes. Ceci va être décrit en détail dans le chapitre ??.

Les auteurs de [Menasri<sup>etal</sup>2008] présentent une méthode d'estimation dans laquelle la ligne de base n'est pas linéaire mais est constituée d'une succession de segments reliant deux points de support successifs. Ces points de support sont extraits à la fois du squelette et du contour du mot. A partir du maximum de l'histogramme horizontal, les auteurs appliquent un seuil haut et un seuil bas, qui permettent de déterminer une approximation de la bande de base. Certains points singuliers (points de branchement, croisement, points de boucles etc.) servent d'estimateurs locaux plus précis de la position de la ligne d'écriture (ligne de base). Un exemple est donné en figure ??.

Dans [BoukermaFarah2010], les auteurs décrivent un algorithme d'estimation de la ligne de base d'écriture basée sur le traitement des sous-mots. A l'intérieur de chaque sous-mot, ils sélectionnent des points de support pour la ligne de base : les points minima du contour inférieur et les bas points des boucles situées sur le bas de la bande horizontale du sous-mot. Au lieu d'utiliser la régression linéaire pour extraire une ligne de base rectiligne, ils interpolent entre chaque deux points consécutifs

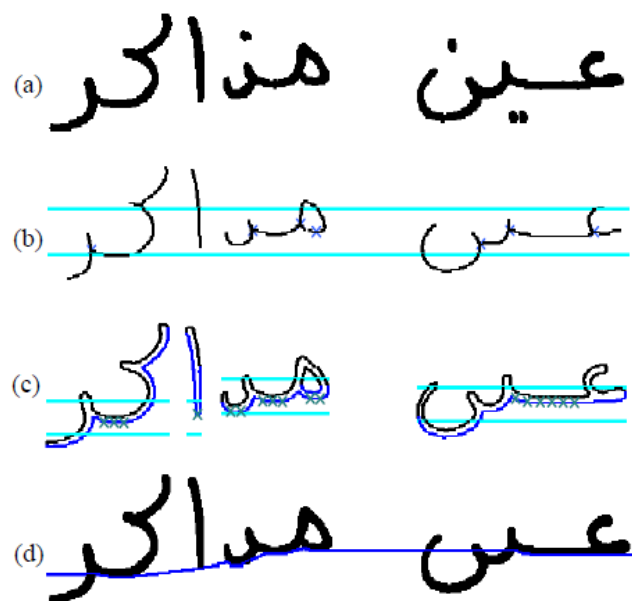


FIGURE 1.12 – Estimation de la ligne de base suivant [BoukermaFarah2010] d'où l'image est tirée.

des points de support sélectionnés. Ils arrivent à extraire en conséquence une ligne qui reflète plus 'réaliste' du mot comme le montre la figure ??.

Dans [Burrow2004], l'auteur utilise la décomposition en valeurs propres (PCA) de l'image de texte pour estimer la position de la ligne de base d'écriture. L'analyse en Composantes Principales est un moyen de trouver les directions selon lesquelles une distribution de points montre le plus de variation. Ces composantes représentent les vecteurs propres les plus significatifs de la matrice de covariance de ces points. Quand on prend comme ensemble de points, les points de premier plan d'une image de mot (texte), ces vecteurs propres décrivent la direction de la ligne de base. L'angle de l'axe central est donné par PCA. La figure ??, tirée de [Burrow2004] montre quelques résultats obtenus par l'auteur.

**Squelettisation** Des dizaines d'algorithmes de squelettisation du manuscrit ont vu le jour qui remontent jusqu'aux années quatre-vingts du siècle passé [Lametal1992]. Nous avons implémenté sept des plus connus d'entre eux. Nous les avons comparés entre eux moyennant des

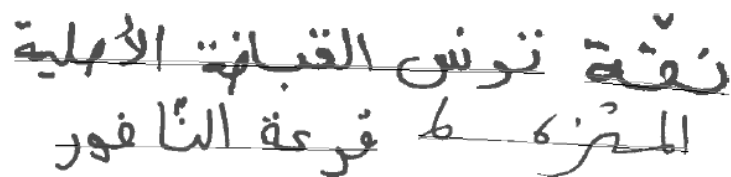


FIGURE 1.13 – Estimation de la ligne de base suivant [Burrow2004] d'où l'image est tirée. La ligne foncée représente la ligne donnée dans la vérité terrain de la base IFN/ENIT, la plus claire est la ligne estimée par l'auteur.

critères visuels et calculables et nous avons proposé une méthode de post-traitement de l'algorithme le plus performant pour le manuscrit arabe. Ceux-ci sont décrits dans le détail dans le chapitre ??.

Ici, nous allons classer les algorithmes de squelettisation des images binaires en trois catégories : séquentiels, parallèles et les algorithmes non-itératifs qui ne se basent sur les configurations des pixels dans leur processus de squelettisation [Lametal1992].

- Dans **les algorithmes séquentiels**, les pixels du contour sont examinés dans un ordre prédéterminé. Ceci peut être obtenu par un balayage avec un masque ou par suivi de contour. Quand un pixel de contour est examiné, il est généralement supprimé ou retenu en accord avec la configuration de ses huit voisins. Pour éviter de supprimer toute une branche en une seule itération, un algorithme séquentiel marquera (flaguera) les pixels à éliminer et tous les pixels marqués sont supprimés à la fois à la fin de l'itération. Les algorithmes de Hilditch [Hilditch1983, Karthikeyan, YuLi2009, Stefanelli1986, Verwer1988], de Marthon [Olleretal2003], Zhou et al [Zhouetal1995] sont des exemples d'algorithmes de squelettisation séquentiels.
- Dans **les algorithmes parallèles**, les pixels examinés pour la suppression sont basés sur les résultats obtenus dans l'itération précédente. Pour éviter tout problème suppression abusive de pixels, ces algorithmes procèdent en sous-itérations où uniquement des directions particulières du contour sont examinées à la fois. Les algorithmes de Zhang & Suen [ZhangSuen1984], de Thomé [Lametal1992], de Datta & Parui [DattaParui1994] et OPTA [ChinWan1987] en sont des exemples.

- **Les algorithmes non itératifs** produisent une certaine médiane ou une ligne centrale directement de la forme en une seule passe sans examiner les pixels individuellement. Ces algorithmes procèdent par calcul de l'axe médian en utilisant des transformations de distance. Ils représentent le plus, la manière dont un humain effectuera un tel traitement [**Lametal1992**].

### Extraction de caractéristiques

Plusieurs études ont utilisé les moments comme les caractéristiques extraites des images et ont montré une bonne performance en reconnaissance des caractères arabes [**AbuBakarShamsuddin2009**, **Alirezaeetal2006**, **Boveiri2010**, **Loprestietal2008**].

Les moments de Krawchouk ont été utilisés dans [**EIAffaretal2009**] pour décrire les mots de la base IFN/ENIT. D'autres techniques ont été utilisées pour reconnaître l'écriture manuscrite comme : l'analyse en ondelettes [**RajputAnita2013**], les filtres de Gabor [**Elzobietal2014**, **RajputAnita2013**] et la transformée en cosinus discrète (DCT) [**RajputAnita2013**]. La technique des fenêtres glissantes a été utilisée pour extraire des caractéristiques structurelles ou statistiques locales [**AlKhateebetal2009**, **Khorsheed2002**]. Les caractéristiques structurelles décrivent les caractéristiques topologiques et géométriques du mot. Elles comprennent les ascendants, les descendants, les boucles, les diacritiques et leur position par rapport à la ligne de base. Les caractéristiques statistiques donnent quelques "mesures" de l'image du mot telles que la densité de l'écriture dans l'image entière ou dans certaines régions d'intérêt (ROI) dans l'image.

### Classification

Divers classifieurs ont été utilisés pour la reconnaissance de l'écriture manuscrite arabe. Parmi eux, nous citons : les HMM (Modèles Markoviens cachés) [**Khorsheed2007**], les ANN (Réseaux Neuraux Artificiels) [**AbdulKader2008**, **AlHamadAbuZitar2010**, **AlKhateebetal2009**], les réseaux LVQ (Quantification des Vecteurs d'Apprentissage) [**Ali2008**], les réseaux neuronaux récurrents multidimensionnels [**GravesSchmidhuber2008**].

et les PGM (modèles graphiques probabilistes) [Khemirietal2014]. Cependant, au meilleur de notre connaissance, il y a très peu d'articles de recherche sur la reconnaissance des mots écrits en arabe utilisant la technique ELM (Extreme Learning Machines).

Dans le domaine de reconnaissance de formes, l'utilisation de plusieurs classifieurs sur les mêmes ensembles d'apprentissage et ensuite combiner leurs résultats pour une meilleure performance du système final a été l'objet de plusieurs travaux [Tulyakovetal2008]. Les premiers travaux remontent à la fin des années 90 du siècle passé [Hoetal1994, Kunchevaetal2001]. Ce concept a également été appliqué à la reconnaissance des mots manuscrits [KimGhahramani2012, Kunchevaetal2001]. L'application aux mots manuscrits arabes est plus récente [Kessentinietal2015, ParvezMahmoud2013].

Ce que nous avons trouvé de plus récent dans l'étude de l'état de l'art dans le domaine de la reconnaissance du manuscrit arabe est celle de Parvez et Mahmoud dans [ParvezMahmoud2013]. Le tableau ?? résume les résultats obtenus dans la reconnaissance des chiffres arabes, les tableaux ??- ?? sont dédiés aux résultats obtenus sur la reconnaissance du manuscrit arabe concernant les mots et les textes. Ces tableaux sont tirés de cette étude.

Nous citons de plus, dans le tableau ??, certains des travaux les plus récents dans le domaine de la reconnaissance de l'écriture arabe manuscrite.

### Segmentation

Le travail le plus proche du nôtre que ce soit en termes d'approche et dans le temps est celui d'El Elzobi et al. [Elzobietal2014]. Nous distinguons cependant des différences entre les deux travaux que nous résumons par :

- i) Dans le travail suscit , la squelettisation est effectu e sans la s paration des parties principales des parties secondaires respectives. Dans notre cas, la squelettisation ne nous a servi qu'  segmenter les parties principales en graph mes. En effet, nous pensons que notre approche est plus   m me de donner de meilleurs r sultats sachant que les

TABLE 1.1 – Résumé des résultats dans la reconnaissance des chiffres arabes [ParvezMahmoud2013]

Author(s)	Year	Features	Classifier	Data	Accuracy
Shirali-Shahreza et al.	1995	Shadow coding	Probabilistic NN	2,600 digits, 10 writers	97.80%
Said et al.	1999	Pixel values	ANN	2,600 digits, 20 writers	94.00%
Sadri et al.	2003	Derivatives of horizontal and vertical profile of boundary	SVM	10,425 digits (CEN-PARMI Indian digit database)	94.14%
Al-Omari and Al-Jarrah	2004	Vector connecting center of gravity and boundary pixel	Probabilistic NN	1,200 digits, 120 writers	99.75%
Harifi and Aghagolzade	2004	Shadow coding based segmentation pattern	ANN	730 digits, 10 writers	97.60%
Ziaratban et al.	2007	Matching coordinates with templates	ANN	10,000 digits, 200 writers	97.65%
Abdelazeem and El-Sherif	2008	Gradient	SVM with RBF kernel	70,000 digits, 700 writers	99.48%
Alaei et al.	2009	Chain code and transition features	SVM	80,000 digits (Farsi)	99.02%
Alamri et al.	2009	Gradient features	SVM with RBF kernel	30,983 digits (CEN-PARMI numeral database)	98.48%
Liu and Suen	2009	Gradient direction histogram	Discriminative learning quadratic discriminant function (CFPC) Class-specific feature polynomial (DLQDF)	CENPARMI Farsi numerals (1,800 digits) IFHCDB Farsi numerals (17,740 digits)	99.16% 99.73%
Mahmoud and Olatunji	2009, Mahmoud 2008a	Angle, distance, horizontal, and vertical span features	SVM, HMM	21,120 digits, 44 writers	99.39%
Mahmoud and Owaidah	2009, Awaidah and Mahmoud 2009	Gradient, structural, concavity	SVM, HMM	21,120 digits, 44 writers	99.83%
Mahmoud and Abu-Amara	2010	Radon and Fourier Transforms	Nearest mean	21,120 digits, 44 writers	98.66%
Mahmoud and Al-Khatib	2010	Gabor filters	SVM, k-NN, NM	10,425 digits (CEN-PARMI checks database)	98.95%
Parvez and Mahmoud	2010	Fuzzy directions	Fuzzy Turning Function	70,000 digits, 700 writers (ADBase)	97.178%

TABLE 1.2 – Résumé des résultats obtenus par les différentes méthodes de reconnaissance du manuscrit arabe (textes et mots) [ParvezMahmoud2013]

Author(s)	Year	Features	Classifier	Data	Accuracy
Pechwitz and Margner	2006	Skeleton directions, Pixel values	Semi-continuous 1-D HMM	IFN/ENIT Train : a,b,c Test : d	89.1%
Margner et al.	2006	Pixel values	HMM	IFN/ENIT Train : a,b,c,d Test : e	74.69%
Kundu et al.	2007	Geometrical and topological features	Variable Duration HMM	IFN/ENIT Train : a,b,c Test : e	60%
Menasri et al.	2007	Graphemes	Hybrid HMM/NN	IFN/ENIT Train : a,b,c Test : d	87.4%
Touj et al.	2007	Directional values, loops, connection of graphemes, presence of diacritics	Planar HMM	IFN/ENIT Train : a,b,c Test : d	86.1%
Khorsheed	2007	Spectral features	HMM (one for each word)	32,000 words from manuscripts	85%
Dreuw et al.	2008	Image slices and their spatial derivatives	HMM with white-space models	IFN/ENIT Train : a,b,c Test : d	92.86%
Broumandnia et al.	2008	Wavelet transform	Nearest Neighbor based on Mahalanobis distance	800 words	96%
Zavorin et al.	2008	Loops, dots, cross-and turning-points	Discrete HMM	IFN/ENIT Train : a,b,c,e Test : d	52%
Ben Cheikh et al.	2008	Linguistic characteristics	Transparent Neural Networks	2,250 words	91%
Natarajan et al.	2008	Percentile of intensity, energy, correlation, angle	HMM	IFN/ENIT Train : a,b,c Test : d	89.4%
Benouareth et al.	2008	Distribution, concavity and skeleton based features	Semi-continuous HMM	IFN/ENIT Train : a,b,c Test : d	90.20%
Dreuw et al.	2009	Image slices and their spatial derivatives	HMM	IFN/ENIT Train : a,b,c,(d) Test : d,e	d : 94.18% e : 88.78%
Mozaffari et al.	2008	Black-white pixel transitions	Discrete HMM	17,000 words of 200 city names	73.61%
Elbaati et al.	2009	On-line features	HMM	IFN/ENIT Train : a,b,c,(d) Test : d,e	d : 83.71% e : 54.13%
Graves and Schmidhuber	2009	Raw pixel data Multidimensional	Recurrent Neural Network	IFN/ENIT Test : f	91.43%

TABLE 1.3 – Suite du tableau ??

Author(s)	Year	Features	Classifier	Data	Accuracy
Hamdani et al.	2009	Pixel values, density, moment, distribution and concavity	Multiple HMM	IFN/ENIT Train : a,b,c,(d) Test : d,e	d : 96.97% e : 81.93%
Haboubi et al.	2009	pixel description, structural description, Gabor Filter, Fourier Descriptors	ANN	16,107 images from IFN/ENIT	87.1%
Saleem et al.	2009	Percentile of intensity, energy, correlation, angle, GSC	HMM	8,692 pages of text	70%
Mohamad et al.	2009	Distribution and concavity features	Multiple HMM	IFN/ENIT Train : a,b,c Test : d	90.26%
Azizi et al.	2010	Global structural feature, density measures	Multi-classifiers (SVM, k-NN, ANN, HMM)	10,000 words, IFN/ENIT	93.96% 94.89%
Kessentini et al.	2010	Directional, contour and density features	Multi-stream HMM	IFN/ENIT Train : a,b,c,d Test : e	79.6%
Chen et al.	2010	Gabor features	SVM	7,346 PAWs	82.7%

TABLE 1.4 – Quelques-uns des travaux plus récents dans la reconnaissance de l'écriture manuscrite

Features	Classifier	Reference	Performance(%)
Ridgelet transform and pixel density features	Artificial Immune Recognition System	Nemmour and Chibani 2017 [NemmourChibani2017]	85.2
Sliding window and Statistical features	HMM	Kessentini et al., 2015 [Kessentinietal2015]	82
Gabor Wavelet	SVM	Elzobi et al., 2014 [Elzobietal2014]	71
Structural features, baseline	Probabilistic Graphical Models	Khémiri et al., 2014 [Khemirietal2014]	92.19
Fourier descriptors and Normalized chain code	SVM	Rajput and Anita, 2013 [RajputAnita2013]	97
Chebyshev, Hu and Zernike moments	MLP, RBF, Fuzzy ART	Chergui, 2013 [Chergui2013]	90.1

diacritiques peuvent créer des obstacles sur la segmentation.

- ii) Dans l'approche considérée, toute colonne ayant un pixel de largeur est candidate à recevoir un point de segmentation potentiel.
- iii) Les auteurs ont utilisé la présence de diacritiques sur la même co-

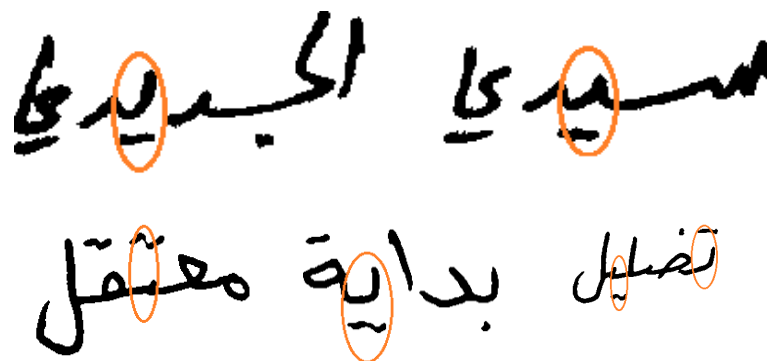


FIGURE 1.14 – Exemples de mots qui ne peuvent pas être correctement segmentés par [Elzobietal2014].

bonne comme une raison pour éliminer le point de la liste des points de segmentation potentiels. Ceci, d'après notre point de vue, n'est pas très réaliste surtout quand il s'agit du manuscrit où les diacritiques peuvent être écrits aléatoirement très loin de leur places appropriées. Ceux-ci peuvent aussi avoir une dimension telle que la prise en compte de cette condition entraîne une mauvaise segmentation du mot. Quelques exemples sont illustrés dans la figure ??.

- iv) Les problèmes de sur segmentation sont résolus par les auteurs lors du processus de segmentation lui-même. Nous pensons que la correction de cette segmentation erronée ne peut pas être effectuée sans une validation préalable.
- v) Dans le processus de reconnaissance, ils procèdent à la normalisation des tailles des caractères ce qui est un non sens pour un script où les différences de tailles entre les caractères est une des principales caractéristiques.

A l'inverse, nous procédons d'abord par séparer les PAWs de leurs diacritiques. Pour minimiser le problème de sur segmentation, nous proposons un seuil pour le nombre de colonnes à un pixel de hauteur pour pouvoir décider d'un point de segmentation potentiel, point que l'on positionne à deux ou trois pixels à partir de la gauche. Ce n'est qu'après avoir fait la segmentation que l'on procède à une deuxième affectation des diacritiques aux parties segmentées. Pour ce faire, nous utilisons des heuristiques basées sur le fait que pour le manuscrit arabe, toutes les

ligatures entre caractères se trouvent approximativement proches de la ligne de base d'écriture. De ce fait, la première heuristique utilisée pour l'élimination des points de segmentation potentiels est la distance de ces derniers par rapport à la ligne de base. Enfin, les caractéristiques que nous avons choisies pour la reconnaissance ne passent pas par la normalisation des tailles des caractères.

## 1.4 Les bases de données concernant le script arabe

### 1.4.1 TBK

La base de données TBK a été réalisée et mise en œuvre par nos soins en 2013. Il s'agit d'une base qui contient tous caractères du script arabe dans leurs différentes formes selon leurs éventuelles positions dans la partie connexe (PAW). Cette base est une base adaptée à la reconnaissance basée segmentation du manuscrit arabe. Pour la construction de cette base, nous avons utilisé un formulaire pour récolter les différentes écritures des caractères manuscrits, les données collectées proviennent de 9 scripteurs de sexes différents et d'âge variant entre 13 et 48 ans avec un niveau intellectuel des scripteurs allant du collègue à l'université. Chaque scripteur reçoit un formulaire contenant 54 cases où il écrit le même caractère dans les 18 cases qui lui sont réservées, ce qui donne la possibilité de collecter le plus de formes pour chaque caractère. Le scan a été effectué avec une résolution de 300 dpi et sauvegardé en binaire sous le format (TIFF : Tagged Image File Format) après avoir bénéficié d'une multitude de pré-traitements rendant cette base prête à toute utilisation ultérieure. Cette base contient :

- L'ensemble début qui contient 2484 formes de caractères écrits au début d'un mot.
- L'ensemble Fin qui contient 3240 formes de caractères en fin de mots.
- L'ensemble milieu qui contient 2486 formes de caractères écrits au milieu d'un mot.
- L'ensemble isolé qui contient 2484 formes de caractères écrits dans leur forme isolée.

Cette base est d'une utilité incontestée pour tout chercheur dans le do-

maine de la reconnaissance du manuscrit arabe. Les méthodes holistiques qui traitent le mot comme étant l'entité à reconnaître, nécessaires et utiles pour les applications à vocabulaire restreint deviennent obsolètes dès que le vocabulaire devient consistant. A ce moment-là, un passage vers la segmentation du mot à reconnaître en ses constituants devient obligatoire pour réduire les entités à reconnaître aux caractères, d'où la venue salutaire de cette base [BoulemkhaliKhouss2013]. De plus, nous avons mis en place un moyen d'enrichissement automatique de cette base avec des caractères segmentés et bien reconnus par notre système tirés de différentes autres bases.

#### 1.4.2 IFN/ENIT

La base de données IFN/ENIT est constituée d'images de mots manuscrits arabes représentant les noms de villes/villages tunisiens (26459 images, pour un vocabulaire de 937 de noms de villes tunisiennes, et 411 scripteurs). Cette base, en plus de sa disponibilité et de sa gratuité pour toute utilisation non commerciale présente le grand avantage d'être variée en termes d'écritures et de styles. En effet, plus de 411 personnes ont participé à son élaboration avec comme consignes une écriture non contrainte pour simuler le comportement d'écriture d'adresses postales.

#### 1.4.3 AHDB

Cette base de données, présentée par Al-Ma'adeed en 2002, contient du texte libre, des mots manuscrits arabes les plus fréquents ainsi que de mots représentant des montants littéraux. Une centaine de scripteurs étaient invités à écrire des mots provenant du vocabulaire des montants numériques, et une vingtaine des mots arabes les plus fréquents et quelques lignes de textes libres. Ces données sont composées d'un lexique de 96 mots, parmi eux 67 mots du vocabulaire de montants littéraux, et 20 mots présentant les mots les plus utilisés dans les textes arabes.

#### 1.4.4 IESK-arDB

La base IESK-arDB est constituée de plus de 4000 images de mots et 6000 images de caractères segmentés. Le vocabulaire de la base couvre la plupart des particules utilisées en Arabe : noms, verbes, noms de villes/pays, termes de sécurité et les mots utilisés pour remplir les montants de chèques. Les 22 scripteurs qui ont contribué à l'élaboration de cette base sont natifs de différents pays où l'arabe est le moyen de communication écrite. Il a été demandé aux scripteurs d'utiliser une écriture le plus possible au style 'Naskh'. Ce style est celui qui est le plus utilisé et il souligne la plupart des particularités structurelles des lettres.

#### 1.4.5 ERIM

La base de données ERIM a été développée dans le début 1995 par l'institut de recherche environnementale du MICHIGAN, cette base de données est composée de 750 pages de textes arabes imprimés [MargnerElAbed2000]

#### 1.4.6 Al-ISRA

La base de données Al-ISRA a été collectée dans l'université de Amman, cette base de données contient une variété de constituants arabes [MargnerElAbed2008] :

- 37000 mots arabes,
- 10000 chiffres arabes et indiens,
- 2500 signatures,
- 500 paragraphes.

#### 1.4.7 Les bases utilisées

Les bases que nous avons utilisées dans ce travail concernent les quatre premières c'est-à-dire TBK, IFN/ENIT, AHDB et IESK-arDB.

### 1.5 Conclusion

Nous avons proposé dans ce chapitre, une entrée dans le domaine de la reconnaissance du manuscrit arabe. Nous avons succinctement pré-

senté les différents problèmes posés par la reconnaissance du manuscrit par rapport à l'imprimé. Nous avons aussi cité quelques particularités du script arabe qui font de ce dernier un script plus difficile à traiter comparé au latin. Nous avons présenté les différents résultats obtenus dans l'état de l'art avec différents types de classifieurs et de caractéristiques. Nous avons enfin cité les quelques bases de données qui servent à valider les prototypes de recherche et nous avons précisé celles que nous avons utilisées dans ce travail.

# Systeme de reconnaissance du manuscrit

## 2.1 Introduction

Après l'entrée introductive au domaine de la reconnaissance du manuscrit arabe, nous allons dans ce chapitre suivre une image manuscrite durant son processus de traitement allant des prétraitements à l'affichage de ce qui a été reconnu de son contenu. Un module de pré-traitements est à la tête de tout système de reconnaissance. Son importance est telle que toute déficience dans ce module peut être la cause d'une déficience de la performance globale du système.

Ceci est encore plus palpable dans le cas de systèmes de reconnaissance des formes. En effet, les images sont souvent bruitées et nécessitent des traitements colossaux pour leur donner un look qui facilite le reste des traitements tels que la segmentation, l'extraction des caractéristiques et finalement la reconnaissance. Le filtrage sert à améliorer l'aspect de l'image par le processus tandis que la binarisation limite les formes de l'image à l'avant-plan (écriture) et l'arrière-plan. La squelettisation résume l'écriture à l'essentiel, c'est-à-dire une écriture à un pixel d'épaisseur. L'image passe ensuite dans le module de segmentation qui a pour but de délimiter les caractères dans le texte. Ces derniers passent dans le module d'extraction de caractéristiques qui transforme chaque image de caractère en un vecteur résumant les caractéristiques essentielles à leur discrimination et reconnaissance. Le module de clas-

sification (reconnaissance) donnera en sortie la suite de mots reconnus moyennant les caractères identifiés dans le texte.

## 2.2 Prétraitements

### 2.2.1 Binarisation

Le but de la binarisation est de faire ressortir l'écriture de l'arrière-plan de l'image. Il s'agit de convertir l'image en niveaux de gris (ou en couleurs) en une image binaire contenant seulement deux couleurs (noir et blanc).

Elle vise à séparer les pixels du texte de ceux de l'arrière plan, et cela en comparant chaque pixel à un seuil pour pouvoir affirmer si oui ou non il fait partie du texte. Le seuil de binarisation adaptatif est le plus approprié pour le traitement automatique des images écrites. Un exemple de seuil est obtenu en calculant [AlAbodiLi2014].

- La moyenne M1 des valeurs des pixels aux quatre coins de l'image ;
- La moyenne M2 des valeurs des pixels au centre de l'image ;
- La moyenne M3 entre M1 et M2 constituera le seuil à prendre pour la binarisation finale.

### 2.2.2 Filtrage

Une multitude de filtres est mise à la disposition du chercheur dans le but de nettoyer au maximum une image en entrée dans le système de reconnaissance du manuscrit arabe. Ces filtres sont disponibles dans des fonctions prédéfinies dans divers langages où le traitement d'images est permis. Parmi les filtres existants, nous citons :

- Les filtres linéaires où la valeur du pixel traité est remplacée par une combinaison linéaire des pixels du voisinage. Un exemple des plus populaires est le filtre moyenneur où le pixel est remplacé par la moyenne des pixels avoisinants.
- Parmi les filtres non linéaires nous citons, le filtre médian et les filtres morphologiques tels que l'érosion, la dilatation, l'ouverture ou la fermeture. Le filtre médian remplace un pixel par la valeur médiane des pixels avoisinants.

Il s'agit alors de choisir la succession de filtres par lesquels on fait passer notre image pour un meilleur rendu et de meilleurs résultats dans les traitements ultérieurs.

### 2.2.3 Squelettisation

La squelettisation de l'écriture arabe manuscrite aspire à donner en sortie une image ne contenant aucune information redondante tout en décrivant complètement la forme originale. Ceci est dans notre cas, une écriture avec une épaisseur d'un pixel avec toute l'information sur la forme, les diacritiques et tous les points essentiels.

### 2.2.4 Ligne de base

La ligne de base ou aussi ligne de référence sont des noms donnés à la ligne fictive (supposée être horizontale) sur laquelle se trouve l'écriture. L'exactitude dans la détection de la ligne de base peut faire une grande différence dans la performance pour un système de reconnaissance de l'écriture. La détection de cette ligne et son redressement si nécessaire sont des étapes très importantes lors de :

1. L'extraction des diacritiques pour la définition de leurs positions par rapport à cette dernière. Notons, ici, que la bonne détection de la ligne de base permet de lever l'ambiguïté sur certains caractères isolés qui peuvent être confondus à des diacritiques comme le 'dal' et le 'alif'.
2. L'extraction de certaines caractéristiques dont il est nécessaire de définir la position par rapport à cette dernière.
3. la segmentation en caractères (ou en graphèmes), les ligatures entre les caractères pour le script arabe se situant souvent le long de cette ligne.

Dans la figure ??, nous présentons trois types de lignes dans un mot manuscrit arabe. Ces lignes sont générées par la méthode de projection. La ligne de base centrale correspond au maximum global de l'histogramme de projection. Les deux lignes supérieure et inférieure correspondent en général aux maxima globaux de part et d'autre du maximum principal de cet histogramme.

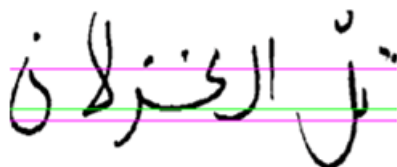


FIGURE 2.1 – Trois lignes de base définies pour un mot de la base IFN/ENIT

## 2.3 Extraction de caractéristiques

### 2.3.1 Caractéristiques globales

#### Les moments orthogonaux (OM)

Les moments sont des entités scalaires utilisées pour représenter une fonction et illustrer ses caractéristiques les plus significatives. Ils ont été largement utilisés pendant des années en statistiques pour décrire la forme des fonctions de densités de probabilité et en mécanique classique pour mesurer la distribution des masses dans un corps. Du point de vue mathématique, les moments représentent les projections d'une fonction sur une base polynomiale. De façon similaire, la transformée de Fourier est une projection d'une fonction sur une base harmonique [Flusseretal2009].

Une image est une fonction réelle à deux variables  $f(x, y)$ , partiellement continue définie sur un domaine  $D \subset R^2$  et ayant une intégrale finie et non nulle.

Un moment général  $M_{pq}(f)$  d'une image  $f(x, y)$ , où  $p, q$  sont des entiers positifs et  $r = p + q$  est appelé l'ordre du moment, est défini comme suit :

$$M_{pq}(f) = \int \int_D P_{pq}(x, y) f(x, y) dx dy \quad (2.1)$$

où  $p_{00}(x, y), p_{10}(x, y), \dots, p_{kj}(x, y), \dots$  sont des fonctions de base polynomiale définies sur  $D$ . Les moments en traitement d'image ont pour origine la notion de moments en physique, ils servent à représenter la distribution des masses dans un corps. Lorsque l'ordre 0 des moments est calculé pour une image binaire, il peut être interprété comme la surface. Le second ordre peut être interprété comme la variance et le troisième comme l'asymétrie. Ceci fait des moments un descripteur mondial et

un puissant outil statistique pour la reconnaissance de formes. Ainsi, grâce aux moments, différents éléments peuvent être utilisés pour décrire l'objet extrait d'une image [2].

Les moments sont dits orthogonaux (OG) si les éléments constituant la base polynomiale satisfont la condition d'orthogonalité :

$$\int \int_{\Omega} P_{pq}(x, y) P_{mn}(x, y) dx dy = 0 \quad (2.2)$$

ou d'orthogonalité pondérée :

$$\int \int_{\Omega} w(x, y) P_{pq}(x, y) P_{mn}(x, y) dx dy = 0 \quad (2.3)$$

pour tous indices  $p \neq m$  or  $q \neq n$ .

Comparés aux moments géométriques standards, les moments orthogonaux (OG) expriment mieux les caractéristiques de l'image et de façon non redondante. Ils présentent aussi l'avantage de pouvoir s'exprimer avec des relations récurrentes ce qui nous permet de les estimer sans utiliser les termes de puissance standards.

Contrairement aux moments géométriques, les moments orthogonaux sont les coordonnées de la fonction dans la base polynomiale au sens commun utilisé en algèbre linéaire. Ceci permet une reconstruction facile de l'image à partir de ces coordonnées avec [Flusseretal2009] :

$$F(x, y) = \sum_{k,j} M_{kj} P_{kj}(x, y) \quad (2.4)$$

Le premier, Hu a introduit ses sept moments invariants par rapport aux transformations affines, à la translation et au changement d'échelle. Seulement, calculer des moments d'ordre supérieur est compliqué et la reconstruction de l'image à partir de ces moments est aussi difficile.

Les moments de Legendre et Zernike sont définis comme des intégrales continues sur un domaine de coordonnées normalisées. Leur implémentation implique des sources d'erreurs suivantes : i) l'approximation discrète des intégrales continues, et ii) la transformation des coordonnées de l'image dans le domaine d'orthogonalité des polynômes [Pawlak2006]. Les moments de Krawchouk et Chebyshev sont définis dans tout  $R^2$  et sont discrets par définition.

**Les moments de Zernike (ZM)** Teague a introduit le concept des moments de Zernike pour recouvrir une image à partir de moments invariants en se basant sur la théorie de polynômes orthogonaux [AbuBakarShamsuddin2009, AlRajabetal2008, Arifetal2009, Chong2003, Marengoetal2008, Norainietal2007, Qaderetal2007]

$$Z_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) R_{nm}(r) \exp(-im\theta) \quad (2.5)$$

L'ensemble des polynômes complexes  $\{V_{nm}(x, y)\}$ , forme un ensemble orthogonal complet à travers le disque unité ( $x^2 + y^2 \leq 1$ ), en coordonnées polaires [Flusseretal2009].

La forme des polynômes est comme suit :

$$V_{n,m}(x, y) = V_{n,m}(r, \theta) = R_{nm}(r) \exp jm\theta$$

où  $n$  est un entier positif ou nul,  $m$  est un entier sujet aux contraintes  $n - |m|$  pair et  $|m| \leq n$ .

$r = \sqrt{x^2 + y^2}$  est la longueur du vecteur de l'origine au pixel  $(x, y)$ ;  $\theta = \arctan \frac{y}{x}$  est l'angle entre le vecteur  $r$  et l'axe des abscisses  $x$  dans le sens contraire à celui des aiguilles d'une montre.  $R_{nm}$  est le polynôme radial défini comme :

$$R_{nm}(r) = \sum_{s=0}^{\frac{n-|m|}{2}} \frac{(-1)^n (n-s)!}{s! [\frac{n+|m|}{2} - s]! [\frac{n-|m|}{2} - s]!} r^{n-2s} \quad (2.6)$$

La formule de récurrence pour l'estimation de la partie radiale (appelée méthode Prata) est :

$$R_{nl}(r) = \frac{2rn}{n+l} R_{n-1,l}(r) - \frac{n-l}{n+l} R_{n-2,l}(r) \quad (2.7)$$

Les valeurs initiales sont :

$$R_{nn}(r) = r^n; R_{n,-n} = r^n; n = 0, 1$$

Les polynômes de Zernike peuvent être calculés pour  $n = 2, 3, \dots; l = -n + 2, -n + 4, \dots, n - 2$ .

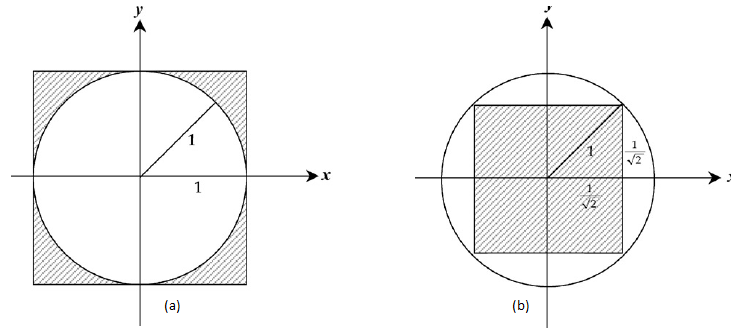


FIGURE 2.2 – Two coordinate normalization schemes : (a) Image coordinate normalization using the mapping  $(0, N - 1) \rightarrow (-1, +1)$ ; (b) Image coordinate normalization using the mapping  $(0, N - 1) \rightarrow (-\frac{1}{\sqrt{2}}, +\frac{1}{\sqrt{2}})$

Les moments bidimensionnels de Zernike à l'ordre  $n$  avec une répétition  $m$  pour une fonction  $f(x,y)$  sont définis comme :

$$Z_{nm} = \frac{n + 1}{\pi} \int \int_{unit\ disk} f(x, y) V_{nm}^*(x, y) dx dy \quad (2.8)$$

où  $V_{n,m}^*(x, y) = V_{n,-m}(x, y)$

Pour évaluer les moments de Zernike pour une image digitale il suffit de remplacer les intégrales par des sommations discrètes :

$$Z_{nm} = \frac{n + 1}{\pi} \sum_x \sum_y f(x, y) V_{nm}^*(x, y) \quad (2.9)$$

où  $x^2 + y^2 \leq 1$ . Comme les moments de Zernike ne sont orthogonaux qu'à l'intérieur du disque unité, lors de l'implémentation, nous devons décider si on doit mettre le disque à l'intérieur de l'image où si c'est l'image qui est mise dans le disque - le premier choix signifie qu'une partie de l'image va être coupé, incluant probablement d'importantes caractéristiques, le second réduit la précision lors de la reconstruction de l'image pour un ordre particulier parce les parties vides du disque à l'extérieur de l'image provoquent une redondance dans l'estimation des moments [Arifetal2009, Chong2003], (figure ??). Comme dans notre travail, nous n'étions pas intéressée par la reconstruction de l'image à partir des moments, nous avons implémenté le second choix (figure ??).b).

Les caractéristiques définies par les moments de Zernike ne sont effectivement invariantes qu'à la rotation. Pour assurer l'invariance des moments de Zernike par rapport à la translation, nous avons déplacé le centre de gravité de l'image vers le centre de l'image avant de les calculer.

**Les moments Pseudo-Zernike (PZM)** Les moments Pseudo-Zernike ont été proposés Bhatia et Wolf [**Hosny2014**] et sont dérivés des moments conventionnels de Zernike et ont la réputation d'être plus robustes et moins sensibles au bruit de l'image que les moments de Zernike. Ils sont bien connus et largement utilisés dans l'analyse de systèmes optiques. Ils sont également utilisés dans l'analyse d'image comme descripteurs de forme. Le polynôme radial Pseudo-Zernike est défini comme :

$$R_{nm}(r) = \sum_{s=0}^{n-|m|} \frac{(-1)^n (2n+1-s)!}{s! [n+|m|+1-s]! [n-|m|-s]!} r^{n-s} \quad (2.10)$$

**Les moments de Legendre (LM)** Les moments avec comme fonction kernel les polynômes de Legendre, appelés les moments de Legendre, ont été introduits pour la première fois par Teague [**Arifetal2009**, **Chongetal2004**, **Flusseretal2009**, **Hosny2010**, **HuangWetal2008**]. Les moments de Legendre font partie des moments orthogonaux et ont été utilisés dans différentes application de reconnaissance de formes [**Flusseretal2009**]. Ils peuvent être utilisés pour atteindre une mesure de redondance proche de zéro parce ces moments correspondent à des caractéristiques indépendantes de l'image [**Chongetal2004**]. Les moments bidimensionnels de Legendre à l'ordre (p + q) pour une fonction intensité de l'image f(x,y) sont définis comme suit :

$$L_{pq} = \frac{(2p+1)(2q+1)}{4} \int_{-1}^1 \int_{-1}^1 P_p(x) P_q(y) f(x,y) dx dy \quad (2.11)$$

Le  $p^{eme}$  ordre du polynôme de Legendre  $P_p(x)$  est défini comme

[Chongetal2004, Flusseretal2009] :

$$P_p(x) = \sum_{k=0, p-k=even}^p B_{kp} x^k \quad (2.12)$$

où,  $x \in [-1, 1]$ , et la matrice des coefficients  $B_{kp}$  est définie par :

$$P_{kb} = (-1)^{\frac{p-k}{2}} \frac{1}{2^p} \frac{(p+q)!}{\left(\frac{p-k}{2}\right)! \left(\frac{p+k}{2}\right)! k!} \quad (2.13)$$

Les polynômes de Legendre obéissent la relation de récurrence suivante :

$$P_{p+1}(x) = \frac{2p+1}{p+1} x P_p(x) - \frac{p}{p+1} P_{p-1}(x) \quad (2.14)$$

avec  $P_0(x) = 1, P_1(x) = x$  et  $p > 1$ . L'ensemble des polynômes de Legendre  $P_p(x)$  forme un ensemble complet de bases orthogonales sur l'intervalle  $[-1, 1]$ . Une image digitale de  $M \times N$  pixels avec une fonction intensité  $f(x_i, y_j)$  est mise en correspondance au domaine  $[-1, 1] \times [-1, 1]$  où,  $1 \leq i \leq M$  and  $1 \leq j \leq N$  par :

$$\begin{cases} x_i = \frac{2 \times i - M - 1}{M - 1} \\ y_j = \frac{2 \times j - N - 1}{N - 1} \end{cases}$$

Avec cette version discrète de l'image, l'équation ?? est généralement approximée en utilisant l'approximation à l'ordre zéro (Zeroth-Order Approximation (ZOA)) comme suit :

$$L_{pq} = \lambda_{pq} \sum_{i=1}^M \sum_{j=1}^N P_p(x_i) P_q(y_j) f(x_i, y_j) \quad (2.15)$$

où,

$$\lambda_{p+q} = \frac{(2p+1)(2q+1)}{MN} \quad (2.16)$$

Les moments de Legendre centrés à l'ordre  $(p+q)$  sont définis par :

$$L_{pq} = \frac{(2p+1)(2q+1)}{4} \int_{-1}^1 \int_{-1}^1 P_p(x) P_q(y) f(x-x_0, y-y_0) dx dy \quad (2.17)$$

où  $(x_0, y_0)$  est le centre de gravité de l'image défini par :

$$x_0 = \frac{\sum_{i=1}^N \sum_{j=1}^N x_i f(x_i, y_j)}{\sum_{i=1}^N \sum_{j=1}^N f(x_i, y_j)}, \quad y_0 = \frac{\sum_{i=1}^N \sum_{j=1}^N y_i f(x_i, y_j)}{\sum_{i=1}^N \sum_{j=1}^N f(x_i, y_j)} \quad (2.18)$$

**Les moments de Chebyshev (CM)** Les moments de Chebychev introduits au domaine du traitement d'images par Mukundan [Mukundanetal2001], définis comme :

$T_n(x) = \cos(n \cdot \arccos(x))$  [Flusseretal2009, MukundanOng2001, Mukundanetal2001, Shuetal2010, Zhuetal2007] sont largement utilisés pour les besoins d'approximations. Pour toute fonction régulière donnée, on peut générer son approximation à partir des moments de Chebyshev par  $f(x) \approx \sum_{n=0}^N \alpha_n T_n(x)$ , où  $T_n$  est un polynôme de  $n$ , et  $\alpha$  est le coefficient de dilatation. Le polynôme de Chebyshev à l'ordre  $n$  est défini comme [Flusseretal2009] :

$$T_n(x) = (1 - N)_n \sum_{k=0}^n \frac{(-n)_k (-x)_k (1 + n)_k}{(k!)^2 (1 - k)_k} \quad (2.19)$$

où  $n, x = 0, 1, \dots, N - 1$  and  $N * N$  est la taille de l'image,  $(a)_k$  est le symbole de Pochhammer donné par :  $(a)_k = a(a + 1)(a + 2) \dots (a + k - 1)$   $k \geq 1$  and  $(a)_0 = 1$ .

Pour une notation plus simple, on introduit :

$$\langle a \rangle_k = (-1)_k (-a)_k = a(a - 1)(a - 2) \dots (a - k + 1)$$

$$k \geq 1 \text{ and } \langle a \rangle_0 = 1$$

L'équation ?? peut être réécrite comme suit :

$$t_n(x) = \sum_{k=0}^n \frac{(n + 1)!}{(n - 1)!(k!)^2} \langle n - N \rangle_{n-k} \langle x \rangle_k = \sum_{k=0}^n B_{nk} \langle x \rangle_k \quad (2.20)$$

avec

$$B_{nk} = \sum_{k=0}^n \frac{(n + 1)!}{(n - 1)!(k!)^2} \langle n - N \rangle_{n-k} \quad (2.21)$$

Les polynômes de Chebyshev satisfont la propriété d'orthogonalité dans le domaine discret suivante :

$$\sum_{x=0}^{N-1} t_n(x)t_m(x) = \rho(n, N)\delta_{nm} \quad (2.22)$$

où  $\delta_{nm}$  représente le symbole Kronecker et la norme au carré de  $\rho(n, N)$  est donnée par :

$$\rho(n, N) = (2n)! \binom{N+n}{2n+1}, n = 0, 1, \dots, N-1$$

Ici  $\binom{p}{q} = \frac{p!}{q!(p-q)!}$  représente le nombre de combinaisons .

La formule de récurrence est la suivante :

$$(n+1)T_{n+1}(x) = (2n+1)(2x-N+1)T_n(x) - n(N^2-n^2)T_{n-1}(x) \quad (2.23)$$

avec les valeurs initiales  $T_0(x) = 1$  and  $T_1(x) = 2x + 1 - N$ . Les moments de Chebyshev pour une image digitale sont donnés par :

$$C_{mn} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} T_m(x)T_n(y)f(x, y) \quad (2.24)$$

**Les moments de Krawchouk (KM)** Contrairement aux moments de Zernike et ceux de Legendre qui sont initialement continus, les moments de Krawchouk [EIAffaretal2009, Flusseretal2009], appartiennent à la classe des moments orthogonaux discrets. Par conséquent, leur implémentation n'implique aucune approximation numérique. De plus, les polynômes de Krawchouk étant orthogonaux sur  $R^2$ , ils ne nécessitent pas de transformation spatiale des coordonnées. La définition des polynômes classiques de Krawchouk à l'ordre n avec paramètre p est donnée par [EIAffaretal2009] :

$$K_n(x; p, n) = \sum_{k=0}^N a_{k,n,p}x^k = {}_2F_1(-n, -x, -N; \frac{1}{p}) \quad (2.25)$$

où  $x, n = 0, 1, 2, \dots, N, N > 0, p \in [0, 1]$ .  ${}_2F_1$  est la fonction hypergéométrique définie comme :

$${}_2F_1(a, b; c; z) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!} \quad (2.26)$$

et  $(a)_k$  est le symbole de Pochhammer (ou rising factorial) donné par

$$(a)_k = a(a+1) \dots (a+k-1) = \frac{\Gamma(a+k)}{\Gamma(a)}$$

$\Gamma$  est la fonction gamma. L'ensemble des  $(N+1)$  polynômes de Krawchouk  $k_n(x; p, n)$  forme un ensemble complet de fonction de bases discrètes avec une fonction poids :

$$w(x; p, N) = \binom{N}{x} p^x (1-p)^{(N-x)}$$

et satisfont la condition d'orthogonalité

$$\sum_{x=0}^N w(x; p, N) k_n(x; p, N) k_m(x; p, N) = \rho(n; p, N) \delta_{nm}$$

où  $n, m = 1, 2, \dots, N$  et

$$\rho(n; p, N) = (-1)^n \left( \frac{1-p}{p} \right)^n \frac{n!}{(-n)_n}$$

La méthode conventionnelle pour éviter les fluctuations numériques lors du calcul des moments est la normalisation par la norme. Les polynômes de Krawchouk normalisés par rapport à la norme sont définis comme :

$$\bar{k}_n(x; p, N) = \frac{k_n(x; p, N)}{\sqrt{\rho(n; p, N)}} \quad (2.27)$$

L'ensemble des polynômes de Krawchouk pondérés est défini par :

$$\bar{k}_n(x; p, N) = k_n(x; p, N) \sqrt{\frac{w(x; p, N)}{\rho(n; p, N)}} \quad (2.28)$$

Les moments de Krawchouk présentent la propriété intéressante d'être capables d'extraire des caractéristiques locales d'une image [Flusseretal2009]. Le paramètre  $p$  détermine la "localisation" du polynôme à l'intérieur de l'intervalle  $[0, N]$ . Si nous avons une connaissance a priori de l'endroit dans l'image où se trouve notre zone d'intérêt, nous pouvons choisir  $p$  de telle sorte que la partie centrale du polynôme est déplacée vers cette zone. Sans connaissance a priori, le choix commun est  $p = 0.5$  [MukundanOng2001]. Les moments de Krawchouk à l'ordre  $(n + m)$  en termes de polynômes pondérés de Krawchouk, pour une image avec une fonction intensité,  $f(x, y)$ , sont définis comme :

$$Q_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \bar{k}_n(x; p1, N - 1) \bar{k}_m(y; p2, M - 1) f(x, y) \quad (2.29)$$

### L'analyse en Ondelettes (DWT)

Les ondelettes sont un outil du traitement du signal récemment développé qui permet l'analyse sur plusieurs échelles de temps des propriétés locales de signaux complexes qui peuvent présenter des zones non-stationnaires [Misitietal2007]. Quand on cherche à analyser un signal, il est très fréquent qu'on établisse, de manière explicite ou implicite, une représentation temps-fréquence de ce signal. La transformée de Fourier n'est pas l'outil approprié pour mener à bien cette analyse puisqu'elle masque l'évolution temporelle du signal. Par contre, la transformée en ondelettes et ses extensions fournissent des solutions intéressantes puisqu'elles peuvent combler une lacune entre deux modes extrêmes : Analyse en fréquence et localisation temporelle. Les ondelettes forment une famille de fonctions de base qui peuvent être employées pour approcher des fonctions générales, ce sont des outils qui ont prouvé leur efficacité en traitement d'image depuis une dizaine d'année. Les techniques basées sur les ondelettes sont appliquées à tous les niveaux du traitement d'images [Walker2007] :

- Compression,
- Dé-bruitage,
- Amélioration de la qualité d'images,

- Reconnaissance,
- Extraction de caractéristiques et enfin,
- Classification de textures.

La principale raison qui fait que l'analyse en ondelettes englobe tant de « visions » du traitement d'images réside dans le fait que l'analyse en ondelettes procure une analyse similaire à celle faite par le système visuel humain. En effet, le système visuel humain réalise une détection hiérarchique des contours à de multiples niveaux de résolution et l'analyse en ondelettes fournit une analyse semblable.

### **L'analyse fractale (FA)**

Il est facile de remarquer que, dans la nature, on ne peut pas construire toutes les formes, simplement à partir de figures géométriques simples, comme des triangles ou des rectangles. Les scientifiques, intrigués, ont longuement tenté d'expliquer ces irrégularités par différents modèles, mais les ont reléguées au titre de " curiosités mathématiques ". Pour faire la lumière sur ces " curiosités ", il a fallu attendre l'arrivée d'un véritable nouveau concept, celui de la géométrie fractale. Quelques idées et modèles relatifs à cette discipline avaient déjà été énoncés au cours du XIXe siècle (poussière de Cantor, courbe de Von Koch), mais ce n'est que vers les années 70, grâce aux travaux de Benoît Mandelbrot, que les fractales seront considérées comme un véritable domaine mathématique. Benoît Mandelbrot a désigné la géométrie fractale pour caractériser ces objets aux propriétés inhabituelles en géométrie classique. La géométrie fractale peut être comprise comme une étendue de la géométrie Euclidienne. Les fractales étant des objets présentant des irrégularités géométriques, leur description et leur modélisation s'effectuent à l'aide d'un indice puissant de mesure de ces irrégularités appelé « dimension fractale ».

**Définition d'une fractale** Le mot fractal est un terme proposé par Mandelbrot (1975) à partir de la racine latine « fractus » et qui signifie irrégulier ou brisé. Il caractérise toutes les formes géométriques qu'on peut imaginer comme faites de parties dont chacune est un modèle ré-

duit du tout [Mozaffarietal2007]. Un objet fractal est ainsi un objet naturel qui combine les caractéristiques suivantes [Pesquet2002] :

- Ses parties ont la même forme ou structure que le tout, à ceci près qu'elles sont à une échelle différente et peuvent être légèrement déformées (propriété d'auto-similarité).
- Sa forme est, soit extrêmement irrégulière, soit extrêmement interrompue ou fragmentée, quelle que soit l'échelle d'examen.
- Il contient des 'éléments distinctifs' dont les échelles sont très variées et couvrent une très large gamme.

Ainsi, une fractale serait un objet mathématique se caractérisant par son irrégularité et son auto-similarité.

#### La transformée en cosinus discrète (DCT)

La DCT est la transformée la plus populaire utilisée en traitement du signal. Elle transforme un signal du domaine spatial vers le domaine fréquentiel. Grâce à sa bonne performance, elle a été utilisée dans la norme de compression d'image JPEG.

La transformée de cosinus discrète (DCT) a été également largement utilisée dans des problèmes de reconnaissance de motifs telle que :

- La reconnaissance de visage
- La reconnaissance de l'iris
- La reconnaissance du manuscrit.

La DCT unidimensionnelle est utile dans le traitement unidimensionnel tel que les signaux de parole, formes d'onde. Pour l'analyse des signaux à deux dimensions (2D) tels que des images, nous avons besoin d'une version 2D de la DCT, et elle est calculée comme suit :

$$D(i, j) = C(i)C(j)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}p(x, y)\cos\left[\frac{2x+1}{2M}i\pi\right]\cos\left[\frac{2y+1}{2N}j\pi\right] \quad (2.30)$$

avec :

$$C(i) = \begin{cases} \sqrt{\frac{1}{M}} & \text{si } i = 0, \\ \sqrt{\frac{2}{M}} & \text{si } 1 \leq i \leq M - 1. \end{cases}$$

$$C(j) = \begin{cases} \sqrt{\frac{1}{N}} & \text{si } j = 0, \\ \sqrt{\frac{2}{N}} & \text{si } 1 \leq j \leq N - 1. \end{cases}$$

### Propriétés et utilité de la DCT [RajputAnita2013]

- La transformée en cosinus discrète DCT est une technique pour convertir les valeurs des pixels d'une image en ses composantes de fréquences élémentaires. En appliquant la DCT sur chaque image, on obtient des coefficients DCT de cette image. Le nombre de coefficients est égal au nombre de pixels dans l'image.
- Une grande partie de l'énergie du signal se trouve dans les coefficients de fréquences basses. Ceux-ci apparaissent dans le coin supérieur gauche de la DCT et les valeurs du coin inférieur droit représentent les plus hautes fréquences. Cette technique polarise les coefficients de grande valeur dans le coin supérieur gauche et les coefficients de faible valeur en bas à droite de la matrice.
- L'une des caractéristiques de la DCT est sa capacité à concentrer l'énergie de l'image en un petit nombre de coefficients significatifs. Les coefficients de la DCT sont extraits en utilisant différentes approches qui sont ensuite mémorisées dans un vecteur.

### Les différentes approches d'extraction des coefficients de la DCT

**DCT upper left corner (ULC) coefficients** Cette approche applique la DCT sur chaque image de l'ensemble de données et conserve les N coefficients significatifs dans la partie supérieure gauche de l'image. La matrice de coefficients retenue est ensuite transformée en un vecteur pour former le vecteur de caractéristique qui sera introduit dans le classifieur (voir figure ??). Le nombre N de coefficients est choisi expérimentalement.

**DCT zigzag coefficients** La même idée que l'approche précédente, d'abord la DCT est appliquée à chaque image, puis les plus hautes valeurs de coefficient de la DCT qui correspondent aux basses fréquences sont extraites d'une manière en zigzag et stockées dans un vecteur comme illustré sur la figure ??.

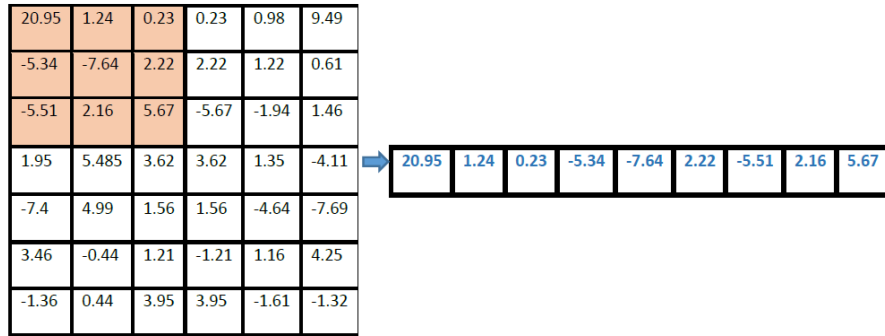


FIGURE 2.3 – Exemple d’une DCT pour laquelle on retient les coefficients ULC  $3 \times 3$

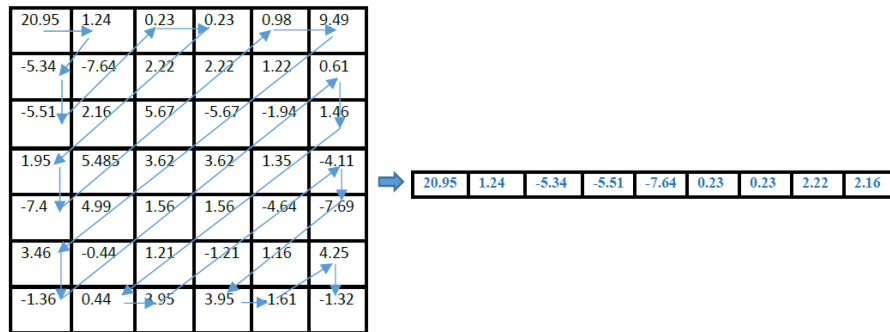


FIGURE 2.4 – Prise de  $3 \times 3$  coefficients DCT en zigzag

**Block based DCT ULC coefficients** L'image entrée est partitionnée en blocs, puis la DCT est appliquée à chaque bloc. Les coefficients sont extraits avec la méthode ULC. Les coefficients retenus sont concaténés pour former le vecteur caractéristiques.

**Block based DCT zigzag coefficients** La méthode zigzag est appliquée pour l'extraction des coefficients DCT sur différents blocs de l'image.

### La décomposition en valeurs singulières (SVD)

La décomposition de matrices est une transformation d'une matrice donnée en un produit de matrices comme LU, QR, Cholesky, et la décomposition en valeurs singulières (SVD) [Baker2013]. La SVD, parmi les factorisations matricielles éminentes, est un des outils de base et sans aucun doute le plus important en algèbre linéaire. Cette décomposition possède une large gamme d'applications, y compris :

- Le traitement du signal
- Le contrôle automatique
- L'exploration de données
- L'approximation de la matrice.
- La reconnaissance des formes
- La détermination du rang et inversion des matrices
- La compression de l'information.

**Propriété de la SVD** La particularité de la SVD est qu'elle peut être effectuée sur toute matrice complexe ou réelle  $A$  ( $m, n$ ) [Baker2013]. Supposons que nous ayons une matrice à  $m$  lignes et  $n$  colonnes, la décomposition en valeurs singulières de cette matrice est la factorisation de  $A$  en un produit de trois matrices (voir figure ??).

- $U$  est une matrice  $m \times m$  orthogonale contenant les vecteurs singuliers gauches :

$$U = (u_1, u_2, \dots, u_r, u_{r+1}, \dots, u_m)$$

- $V$  est une matrice  $n \times n$  orthogonale contenant les vecteurs singuliers droits :



décomposer une image numérique en trois matrices où les données prennent une forme dans laquelle les premières valeurs singulières ont une grande quantité d'information sur l'image. Avec cela, nous pouvons utiliser seulement quelques valeurs singulières pour représenter l'image.

- Les valeurs singulières d'une telle factorisation nous permettent de représenter l'image avec un plus petit ensemble de valeurs, qui peut préserver des caractéristiques utiles de l'image d'origine en gardant l'essentiel de l'information. La SVD a la propriété de concentrer l'énergie d'une image dans le moins de coefficients.

### Les filtres de Gabor (GF)

Les filtres de Gabor sont des filtres passe-bandes sensibles à la direction, ils ont l'avantage d'être paramétrables en fréquence et en orientation, par opposition à d'autres techniques globales comme la Transformée de Fourier. Pour cela, les filtres de Gabor sont très souvent utilisés pour extraire les caractéristiques dans une direction souhaitée de l'image du document [**Chenetal2010**, **Elzobietal2014**, **RajputAnita2013**].

Ce sont des filtres simples à créer et ont une résolution conjointe spatiale/fréquentielle c'est-à-dire qu'ils offrent la meilleure localisation simultanée en temps et en fréquence d'après le principe d'incertitude d'Heisenberg.

Cette famille de fonctions élémentaires 2-D constitue une généralisation des fonctions élémentaires 1-D proposées en 1946 par Gabor dans son célèbre monographie, "Théorie de la communication." Le filtrage par Gabor conserve les aspects temporels et fréquentiels du signal.

**Fonction de Gabor** [**Chenetal2010**] La fonction de Gabor est une fonction bidimensionnelle constituée d'une onde plane sinusoïdale modulée par une gaussienne bidimensionnelle.

La fonction sinusoïdale est caractérisée par sa fréquence et par son orientation, où le terme fréquence se réfère à la variation de luminosité et ou à la couleur sur l'image. Dans le domaine spatial, l'application des

filtres de Gabor est effectuée en calculant la convolution de l'image avec une fonction réglée à une des textures.

Nous avons trouvé dans la littérature, plusieurs variantes de la formulation de la fonction de Gabor, appliquée au traitement d'images. Celle que nous avons adoptée est celle citée dans [Haghighatetal2015] :

$$g(x, y) = \frac{f^2}{\pi\gamma\eta} e^{(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2})} e^{(j2\pi f x' + \psi)} \quad (2.31)$$

avec :  $x' = x \cos \theta + y \sin \theta$  et  $y' = y \cos \theta - x \sin \theta$

Où :

- $\sigma$  : l'écart type de la gaussienne
- $\theta (\theta \in [0, \pi])$  : l'orientation des filtres de Gabor
- $f$  : la fréquence radiale de la sinusoïde
- $\psi$  : décalage de phase (phase shift)
- $\gamma$  : rapport d'aspect qui spécifie l'ellipticité du filtre

#### Propriétés des filtres de Gabor [Chenetal2010, Elzobietal2014, RajputAnita2013]

- Les filtres de Gabor minimisent l'incertitude de Heisenberg qui exprime le fait que plus on est précis dans l'analyse de l'information dans l'espace du pixel et moins on le sera dans l'espace fréquentiel et vice versa. Il a été démontré que la fonction complexe de Gabor atteint la limite inférieure du principe d'incertitude. Ils sont :
- Réglables pour détecter la présence de composantes fréquentielles orientées dans une direction particulière.
- Capables de détecter des périodicités de fréquence et d'orientation choisies.
- Le filtrage par Gabor conserve les aspects temporels et fréquentiels du signal.
- La production de filtrage est robuste aux bruits divers [RajputAnita2013].

### 2.3.2 Caractéristiques analytiques

#### Caractéristiques topologiques ou métriques

Le terme métrique désigne la mesure d'une distance. La topologie est "l'étude des propriétés de l'espace (et des ensembles) du seul point de

vue quantitatif". Concrètement, la topologie consiste, à l'aide de sondes appliquées directement sur l'image "brute", à effectuer par exemple sur l'échantillon les mesures et les tests suivants :

- compter dans une forme le nombre de trous,
- évaluer les concavités,
- mesurer des pentes et autres paramètres de courbures et évaluer des orientations principales,
- mesurer la longueur et l'épaisseur des traits,
- détecter les croisements et les jonctions des traits,
- mesurer les surfaces, les périmètres,
- déterminer le rectangle délimitant l'échantillon, ou la boîte englobante,
- évaluer le rapport d'élongation (ou allongement) longueur/largeur, ...
- rendre compte de la disposition relative de ces primitives.

#### Caractéristiques structurelles

A la différence des primitives topologiques, les primitives structurelles sont généralement extraites non pas de l'image brute, mais à partir d'une représentation de la forme par le squelette ou par le contour. Ainsi, on ne parle plus de trous, mais de boucles ou de cycles dans une représentation filiforme du caractère. Cependant, pour le reste, les primitives structurelles correspondent à peu près aux primitives topologiques, il s'agit principalement :

- des segments de droite,
- des arcs, boucles et concavités, des pentes,
- des angularités, points extrêmes et points terminaux, jonctions et croisements.

#### Caractéristiques statistiques

Elles véhiculent une information qui est distribuée sur toute l'image. L'histogramme, qui représente le nombre de pixels sur chaque ligne ou colonne de l'image, en est un exemple classique et simple à calculer. L'histogramme directionnel est plus long à calculer car il nécessite par exemple l'utilisation d'un algorithme de Bresenham [**Bresenham1965**]

permettant de compter le nombre de pixels contenus sur une ligne de direction quelconque de l'image. L'histogramme des transitions, comme l'indique son nom, ne retient que le nombre des transitions 0-1 et 1-0. Un exemple d'application des primitives statistiques est réalisée en cherchant l'intersection de la forme avec un réseau de droites pour la reconnaissance des chiffres manuscrits [Dargenton1994]. Dans ce cas, des histogrammes de transition sont construits seulement avec un échantillon de quelques droites au lieu de la totalité de celles-ci, ce qui permet une réduction a priori des données caractéristiques. Pour s'affranchir du choix arbitraire de ces droites (espacement et orientation des droites du réseau) qui doit nécessairement être fixe entre l'apprentissage et la reconnaissance, on peut faire intervenir les probabilités en calculant la fréquence des intersections de la forme avec une série de droites aléatoires [Dargenton1994]. On utilise aussi une autre primitive statistique basée sur le calcul de la moyenne des pixels situés à l'intérieur d'un masque rectangulaire. La construction d'une matrice de masque recouvrant la totalité de la forme permet une représentation statistique à partir d'un nombre réduit de valeurs correspondant à chaque masque.

### 2.3.3 Technique de zonage ou fenêtrage [ImpedovoPirlo2014]

La technique de zonage ou fenêtrage est une technique destinée à récolter des informations locales sur les images à traiter sans pour autant être obligé de passer par le lourd processus de segmentation. Le zonage implique l'utilisation de fenêtres par lesquelles on peut regarder de 'plus près' dans des zones d'intérêt de l'image. Ces fenêtres peuvent être de n'importe quelle forme. Elles peuvent être de simples fenêtres glissantes ayant la hauteur de l'image et coulissant de la droite vers la gauche ou vice versa. Celles-ci peuvent être aussi des zones de formes plus générales allant des blocs de différentes tailles circulant librement dans l'image aux formes circulaires centrées sur les centres de gravité de zones d'intérêt autour de la ligne de base d'écriture par exemple.

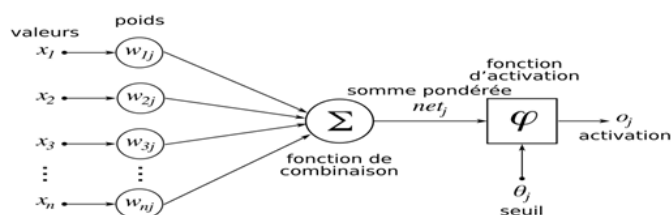


FIGURE 2.7 – Le neurone formel

## 2.4 Méthodes de classification

Dans le domaine de reconnaissance des formes, une panoplie de classifieurs ont été mis en application. Dans ce qui suit, nous allons succinctement décrire les seuls classifieurs qui ont été utilisés dans ce travail.

### 2.4.1 Les K plus proches voisins (K Nearest Neighbors KNN)

Le classifieur basé sur les K plus proches voisins est un simple classifieur. Il a pourtant été prouvé qu'il est utilisé avec succès même quand la séparation entre les classes n'est pas du tout évidente [**Hastieetal2008**].

Dans le présent travail, nous avons utilisé le classifieur K-Nearest Neighbors avec la distance «Corrélation» après avoir testé trois autres distances, à savoir la distance «Euclidienne», «City Block» ou «Manhattan» et la distance «Cosinus». Nous avons testé K-NN avec différentes valeurs du paramètre  $K$ . Empiriquement, la valeur  $K$  qui a donné les meilleurs résultats a été  $K = 10$ .

### 2.4.2 Les réseaux de neurones (RN)

Un réseau de neurones artificiels formels tel que, McCulloch et Pitts le définissent, est un ensemble de composants calculatoires appelés "neurones" faisant la somme pondérée des signaux reçus en entrée, à laquelle on applique une fonction de transfert (voir Figure ??).

#### Les réseaux de neurones multi-couches feedforward

Un perceptron multicouches est un modèle feedforward de réseaux de neurones artificiels. Le réseau de neurones feedforward a été le premier

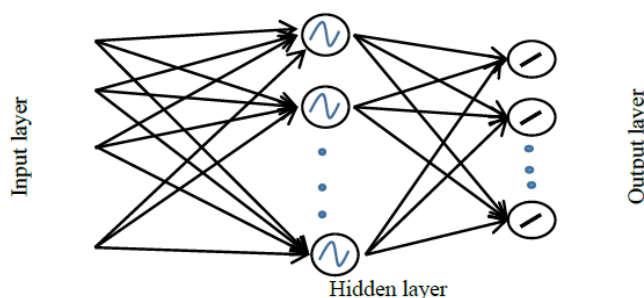


FIGURE 2.8 – Feedforward Neural Network

et le type le plus simple de réseaux de neurones artificiels conçus. Il n'y a pas de boucles dans ce réseau et l'information ne se déplace que vers l'avant, démarrant des nœuds d'entrée, à travers les nœuds cachés (s'il y en a) et vers les nœuds de sortie (voir Figure ??).

Les réseaux neuronaux feedforward peuvent prédire tout comportement ou prendre toute décision de matching ou de classification uniquement en les entraînant avec un nombre suffisant de données significatives. Ils ont été utilisés avec succès dans de nombreux domaines en raison de leur pouvoir : (1) d'approcher des fonctions complexes non linéaires directement à partir des échantillons d'entrée ; et (2) pour fournir des modèles pour une grande classe de phénomènes naturels et artificiels qui sont difficiles à manipuler en utilisant des techniques classiques [Huangetal2006b, HuangChen2007].

### 2.4.3 Les Machines à vecteurs de support (Support Vector Machines (SVM))

Durant les dernières décades, les machines à vecteurs de support (Support Vector Machines SVM) ont joué un rôle important dans des applications dédiées à la recherche et la classification [Elarianetal2015, Elzobietal2014, Huangetal2004]. Si l'on considère un ensemble de données d'apprentissage  $(x_i, c_i)_{i=1}^N$ , où  $x_i \in R^d$  et  $c_i \in \{-1, 1\}$ , les SVM permettent de trouver l'hyperplan optimal maximisant la marge de séparation entre les deux classes [Burges1998, Fletcher2009, Huang2014]

par :

$$\text{Minimization : } D = \frac{1}{2}w.w + C \sum_{i=1}^N \varepsilon_i \quad (2.32)$$

Sujet aux contraintes :  $c_i(w.x_i + b) \geq 1 - \varepsilon_i, \forall_i \varepsilon_i \geq 0, \forall_i$  où :  $C$  est un paramètre qui permet de réaliser le compromis entre la minimisation de l'erreur d'apprentissage et la maximisation de la distance  $\frac{2}{\|w\|}$  de la marge de séparation entre les deux différentes classes. Les données utilisées pour trouver l'hyperplan optimal sont appelées "les vecteurs de support".

Dans le cas où les données sont non linéairement séparables, une mise en correspondance appelée "noyau" (kernel) entre les données et l'espace de caractéristiques permet de rendre le problème linéairement séparable dans le nouvel espace. Les machines à vecteurs de support avec différents noyaux peuvent transformer la majorité des problèmes non linéairement séparables en des problèmes linéairement séparables [Burges1998, Huangetal2004]. Parmi les noyaux les plus fréquemment utilisés, nous avons :

- Le noyau linéaire :  $K(x_i, x_j) = x_i^T x_j$
  - Le noyau polynomial :  $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
  - Le noyau radial basis function (RBF) :  $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$
  - Le noyau sigmoïde :  $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$ .
- $\gamma, r$  et  $d$  sont des paramètres des différents noyaux.

Initialement, les machines à vecteurs de support ont été conçues pour trouver des frontières entre deux classes de données. Quelques stratégies ont été ensuite proposées pour étendre cette technique aux problèmes multi-classes [Burges1998, NiuSuen2012]. Parmi ces techniques, citons :

- **La stratégie "Un contre tous" (One Against All (OAA)) :**  
 Dans cette stratégie, étant donné un problème de classification à  $n$  classes,  $n$  SVM $_i, i = 1, \dots, n$  sont entraînés pour séparer, chacun les données appartenant à une classe  $i$ , des données appartenant à toutes les autres classes. Des scores qui correspondent aux taux d'apprentissage à leurs classes respectives, sont affectés à chaque SVM. Durant

l'étape de test, l'entrée inconnue est présentée à chaque SVM, et est affectée à la classe correspondant au meilleur SVM (en termes des scores affectés lors de l'apprentissage) qui reconnaît la classe comme lui appartenant.

**La stratégie "Un conte un" (One Against One (OAO)) :** Dans ce cas, on construit un SVM pour chaque paire de classes. Ainsi, on sera amené à entraîner  $\frac{n(n+1)}{2}$  SVM binaires. Cette approche est associée en général à une opération de vote majoritaire pour la diffusion des décisions des  $\frac{n(n+1)}{2}$  classifieurs SVM. Cette approche présente l'avantage de dessiner moins de zones d'ambiguïté mais elle a l'inconvénient de demander le calcul de  $\frac{n(n+1)}{2}$  jeux de paramètre au lieu de  $n$ . Le Problème de l'ambiguïté peut être résolu par un système ultérieur de décision sur l'ensemble des classes, prenant en compte des paramètres supplémentaires, par exemple la distance aux hyperplans appris et/ou les probabilités etc.

#### 2.4.4 Les Extreme Learning Machines (ELM)

Huang & al [**Huangetal2006b, HuangChen2008, Huangetal2012**], ont été les premiers à remarquer le fait que quand il apprend avec des exemples, l'être humain et même un enfant est capable de déduire la classe d'un objet avec un temps d'apprentissage pratiquement nul en se basant uniquement sur les exemples fournis. Il n'a besoin d'aucune préparation d'aucune sorte pour ses déductions. Inspirés par ces caractéristiques biologiques d'apprentissage, Huang & al [**Huangetal2006a, Huangetal2006b, HuangChen2007, HuangChen2008, Huangetal2012, Huang2014, Huangetal2004**] ont supposé qu'une partie du cerveau humain a des neurones aléatoires dont le fonctionnement est complètement indépendant de l'environnement. Ils ont nommé cette technique "machine à apprentissage extrême" (Extreme Learning Machine ELM).

Cette technique est une alternative pour remédier aux problèmes liés à l'algorithme de rétro-propagation du gradient utilisé dans les réseaux de neurones multi-couches et au rythme assez lent d'apprentissage de ces derniers. c'est une alternative qui rencontre de plus en plus d'enthousiasme.

La technique ELM a d'abord été proposée pour entraîner un réseau de neurone feedforward à une seule couche cachée de neurones (Single Layer feedforward neural network (SLFN)). Elle a ensuite été généralisée aux réseaux où la couche cachée n'est pas obligée d'être à structure neuronale.

La fonction de sortie d'un ELM pour un réseau de neurones monocouche est donnée par :

$$f_L(x) = \sum_{i=1}^L \beta_i h_i(x) = h(x)\beta \quad (2.33)$$

où  $\beta = [\beta_1, \dots, \beta_L]^T$  est le vecteur de poids de sortie entre la couche cachée de  $L$  noeuds aux  $m \geq 1$  noeuds de sortie, et  $h(x) = [h_1(x), \dots, h_L(x)]$  est le vecteur (ligne) de sortie de la couche cachée par rapport à l'entrée  $x$ .  $h_i(x)$  est la sortie du  $i^{eme}$  noeud caché de sortie, et les fonctions de sortie des noeuds cachés ne sont pas obligées d'être uniques [Huangetal2004].

#### 2.4.5 Les Modèles de Markov Cachés (HMM)

Un modèle de Markov caché (MMC) en anglais Hidden Markov Model (HMM) est un modèle statistique dans lequel le système modélisé est supposé être un processus markovien de paramètres inconnus. Les HMM ont été décrits pour la première fois dans une série de publication de statistiques par « Leonard E. Baum » et d'autres auteurs après 1965. Ils ont été appliqués dès la fin des années 1970 à la reconnaissance de la parole. Et comme ils ont donné de bons résultats dans le domaine du traitement de signal qui partage des points avec la reconnaissance de l'écriture manuscrite (signal bidimensionnel), cela a encouragé les chercheurs de cette dernière à les utiliser. L'application des HMMs à la reconnaissance de l'écriture se ramène généralement à la transformation de la forme en primitives, traduisant les formes initiales en un signal unidimensionnel composé d'une suite de séquences appelées observations destinée à être traitée par un modèle de Markov. Des applications relativement limitées basées sur les HMMs, ont été développées pour la reconnaissance

de l'arabe, aussi bien imprimé que manuscrit, en ligne et hors-ligne.

### Modèles de Markov cachés unidimensionnels (HMM 1D)

Un modèle de Markov caché unidimensionnel, est une chaîne de Markov stationnaire d'ordre 1. Le modèle résultant est un processus doublement stochastique et il est désigné par :

$$(O_t, Q_t) \text{ tel que : } 1 \leq t \leq T.$$

- $Q = q_1, q_2, \dots, q_T$  une suite (inconnue) des états où  $Q_t$  une chaîne interne qui change d'états selon une loi de transition (non observable).
- $O = o_1, o_2, \dots, o_T$  la suite des symboles d'observations dans chaque état. L'observateur ne peut voir que les sorties des fonctions aléatoires associées aux états et ne peut pas voir les états internes d'où le terme de Modèles de Markov Cachés.

Le processus  $(Q_t)$  où  $0 \leq t \leq T$  est une chaîne de Markov d'ordre 1, qui doit vérifier :

$$A = \{a_{ij}\}$$

$$a_{ij} = P(Q_{t+1}=q_i | Q_t=q_j, \dots, Q_0=q_0) = p(Q_{t+1}=q_i | Q_t=q_j), 1 \leq i, j \leq N.$$

$$\sum_{j=1}^N a_{ij} = 1$$

Où :  $A$  : est la matrice des probabilités de transition entre les états.  $N$  : le nombre d'états dans le modèle.

#### 2.4.6 Le classifieur forêts aléatoires (Random forests)

Les arbres de décision sont utilisés pour diverses tâches d'apprentissage automatique. Les arbres qui sont cultivés de façon très dense ont tendance à apprendre des modèles très irréguliers : ils sur-apprennent leurs ensembles de données d'apprentissage. Ils ont de ce fait un pouvoir de généralisation aux données non apprises très réduit. Les forêts aléatoires sont un moyen de faire la moyenne de plusieurs arbres de décision

denses, entraînés sur différentes parties d'un même ensemble de données d'apprentissage, dans le but de réduire la variance. Ceci améliore grandement la performance du modèle final [**Breiman2001**].

## 2.5 Conclusion

Dans ce chapitre, il a été question de présenter les différents modules constituant tout système de reconnaissance du manuscrit arabe. Nous avons succinctement décrits les modules essentiels dans tout système de reconnaissance du manuscrit. Il s'est agi des modules de prétraitements, d'extraction de caractéristiques et enfin de classification. Nous détaillerons notre contribution dans les différents aspects suscités à partir du chapitre suivant.

## Partie II : Contributions

Il est question dans cette partie, de résumer quelques unes de nos contributions dans le domaine de reconnaissance du manuscrit arabe. Ces contributions sont subdivisées selon les grands modules constituant tout système de reconnaissance du manuscrit à savoir :

- Le module de prétraitements qui constitue un préalable à tout traitement ultérieur. Nous expliquons succinctement les différents prétraitements implémentés et nous limitons ici ces derniers à deux grands axes qui sont :
  - La squelettisation ;
  - La détection et redressement de la ligne de base.
- La deuxième partie est consacrée au module d'extraction de caractéristiques qui lui aussi est décisif quant aux taux de reconnaissance obtenus par le système. L'extraction de caractéristiques ainsi que la classification étant fortement subordonnées l'une de l'autre, notre contribution dans ces modules est décrite de façon conjointe. Le module de classification résumera la majorité des classifieurs utilisés dans ce travail. Il inclura aussi une proposition de combinaison de classifieurs qui a donné naissance à des résultats compétitifs par rapport à ceux obtenus dans l'état de l'art actuel.
- La dernière partie concernera le module de segmentation qui est toujours d'actualité et présente encore de très grands challenges aux cher-

cheurs. Ce module permet de réduire le volume du lexique utilisé dans l'apprentissage et la validation qui va à ce moment-là se limiter aux caractères (ou graphèmes) obtenus lors de la segmentation du manuscrit.

## Contribution I : Prétraitements

### 3.1 Introduction

Dans ce chapitre, nous allons passer en revue les différents traitements que nous avons effectués et nous allons nous concentrer sur les deux composantes de base de tout prétraitement à savoir :

- La squelettisation ;
- La détection et le redressement de la ligne de base d'écriture.

Nous allons aussi décrire dans ce chapitre, la base de données TBK que nous avons eu à réaliser pour valider notre système basé sur la segmentation en caractères du manuscrit arabe.

### 3.2 Squelettisation

Quand on a affaire à un problème de reconnaissance des formes, nous avons souvent recours à des techniques de réduction de la quantité de l'information traitée. Lorsque nous traitons l'information contenue dans un texte écrit que ce soit de l'imprimé ou du manuscrit, on se rend compte que l'information utile se trouve uniquement dans la forme de l'écriture sans considération pour l'épaisseur de cette dernière. C'est pourquoi, il est souvent d'usage, dans ce cas de procéder à la squelettisation de ce dernier avant tout traitement ultérieur. Un algorithme de squelettisation doit vérifier les propriétés suivantes :

- Il doit préserver toute l'information pertinente contenue dans la forme.

- Il doit assurer une épaisseur de 1 pixel pour le squelette résultat.
- Il doit préserver toutes les connectivités des formes.
- Il doit être insensible au bruit.

Cette étape est un prétraitement d'une grande importance surtout pour les systèmes voués à passer par la segmentation avant la reconnaissance.

Avant de pouvoir expliquer le principe des algorithmes de squelettisation, donnons quelques définitions.

- Le masque de lecture qui nous permet de définir le voisinage d'un pixel P dans les algorithmes qui suivent est celui de la figure ??.

P8	P1	P2
P7	<b>P</b>	P3
P6	P5	P4

FIGURE 3.1 – Le voisinage  $3 \times 3$  d'un point

- $A(P)$  représente le nombre de transitions 0 vers 1 dans la séquence P1, P2, P3, P4, P5, P6, P7, P8, P1.
- $B(P)$  est le nombre de pixels noirs dans le voisinage de P. La figure ?? montre un exemple de calcul de ces deux paramètres.

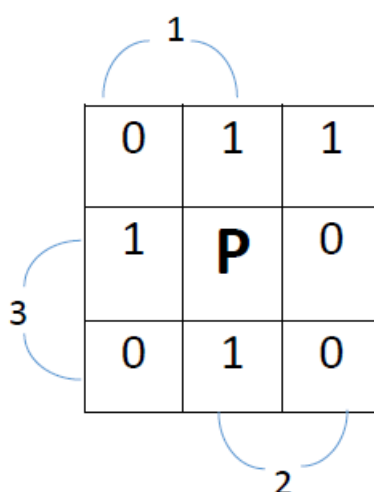


FIGURE 3.2 – Calcul des paramètres  $A(P)$  et  $B(P)$  sur un exemple :  $A(P)=3$ ,  $B(P)=4$

Nous avons implémenté sept algorithmes de squelettisation que nous allons décrire dans ce qui suit.

### 3.2.1 Thomé [Lametal1992]

C'est un algorithme parallèle décomposant la suppression en quatre sous-cycles. Chaque sous-cycle définit ses propres points à effacer comme suit :

- Sous-cycle 1 : P est effaçable si :  $P1 = 0$ .
- Sous-cycle 2 : P est effaçable si :  $P1 = 1$  et  $P5 = 0$ .
- Sous-cycle 3 : P est effaçable si :  $P1 = P5 = 1$  et  $P3 = 0$ .
- Sous-cycle 4 : P est effaçable si :  $P1 = P5 = P3 = 1$  et  $P7 = 0$ .

### 3.2.2 Marthon [Olleretal2003]

C'est un algorithme séquentiel qui sélectionne les points qui sont des maxima par leur distance à la frontière du caractère. Pour chaque point, il calcule la somme des vecteurs joignant ce point à ses voisins, les coordonnées des points voisins prennent leurs valeurs dans  $\{-1, 0, 1\}$  suivant leur position par rapport au point central, considéré comme l'origine. La longueur  $D(P)$  est calculée en sommant les valeurs absolues des composantes X et Y du vecteur somme. Elle est comprise entre 0 et 4. Considérons comme exemple, un point P ayant cinq voisins allumés P1, P2, P5, P6 et P7 comme suit (figure ??) :

0	1	1
1	<b>P</b>	0
1	1	0

FIGURE 3.3 – Exemple du calcul de la distance d'un point par l'algorithme de Marthon, dans ce cas :  $D(P) = 1$

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 \\ -1 \end{pmatrix} + \begin{pmatrix} 0 \\ -1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \implies |X| + |Y| = 1$$

- L'algorithme de squelettisation élimine les points dont la valeur est égale à 4 et retient ceux dont la valeur est inférieure ou égale à 2.
- Les points dont la valeur est égale à 3 nécessitent d'abord une étude du voisinage pour éviter de causer une disconnexion de la forme, donc on ajoute la condition de  $A(P) = 1$  pour supprimer le pixel.

### 3.2.3 Hilditch [Hilditch1983, Karthikeyan, YuLi2009, Stefanelli1986, Verwer1988]

C'est un algorithme d'amincissement séquentiel. L'image est balayée de gauche à droite et de haut en bas puis les pixels qui répondent simultanément aux quatre critères suivants sont marqués comme effaçables :

1.  $2 \leq B(P) \leq 6$
  2.  $A(P) = 1$
  3.  $P1 \times P3 \times P7 = 0$  ou  $A(P1) \neq 1$
  4.  $P1 \times P3 \times P5 = 0$  ou  $A(P3) \neq 1$
- La condition  $B(P) \leq 6$ , assure que P est un pixel du contour.
  - La condition  $2 \leq B(P1)$ , assure la préservation des pixels isolés ( $B(P1) = 0$ ) et d'extrémité (end-point  $B(P1) = 1$ ).
  - La condition  $A(P1) = 1$ , maintient le critère de la connectivité.
  - La condition (3) (respectivement (4)) assure que les lignes verticales (respectivement horizontales) de deux pixels de largeur ne soient pas totalement érodées.

Tous les pixels marqués comme effaçables sont ensuite effacés, et l'algorithme est ré exécuté sur la nouvelle image, jusqu'à ce que plus aucun pixel ne soit effaçable.

### 3.2.4 Zhang & Suen [ZhangSuen1984, LijWang1986]

C'est un algorithme de squelettisation parallèle. Il est divisé en deux étapes, la première supprime les pixels du contour situés au sud-est de la forme et les pixels du coin au nord-ouest, tandis que la deuxième supprime les pixels du contour situés au sud-ouest de la forme et les

pixels du coin au nord-est. Lors de la première étape, un pixel noir  $P$  est marqué comme effaçable s'il respecte les quatre conditions suivantes :

1.  $2 \leq B(P) \leq 6$
2.  $A(P) = 1$
3.  $P1 \times P3 \times P5 = 0$
4.  $P3 \times P5 \times P7 = 0$

Les deux dernières conditions assurent que le pixel  $P$  est un pixel du contour situé, soit au sud-est de la forme, soit au coin nord-ouest. Lors de la deuxième étape, un pixel noir  $P$  est marqué comme effaçable s'il respecte les quatre conditions suivantes :

1.  $2 \leq B(P) \leq 6$
2.  $A(P) = 1$
3.  $P1 \times P3 \times P7 = 0$
4.  $P1 \times P5 \times P7 = 0$

Les deux dernières conditions assurent que le pixel  $P$  est un pixel du contour situé, soit au nord-est de la forme, soit au coin sud-ouest.

### 3.2.5 Zhou et al. [Zhouetal1995]

Le masque utilisé dans cet algorithme est montré dans la figure ???. Une table, appelée table des drapeaux (flag map) sert à mettre des drapeaux sur les pixels qui vont éventuellement être éliminés du squelette. La taille de cette table est la même que celle de l'image entière. Dans la figure ??? les points  $Qi$  correspondent aux  $Pi$  dans le premier tableau.

P1	P2	P3
P8	<b>P</b>	P4
P7	P6	P5

Q1	Q2	Q3
Q8	<b>Q</b>	Q4
Q7	Q6	Q5

FIGURE 3.4 – Le masque et la table des drapeaux utilisés dans l'algorithme de squelettisation de Zhou et al. pour la détermination des points susceptibles d'être effacés de l'image.

Les cases de la table des drapeaux sont initialisés à 1. Dès qu'un point est "flagué" pour être éventuellement effacé du squelette, son drapeau

correspondant est mis à 0. Les critères utilisés dans cet algorithme pour décider qu'un point est à effacer ou pas sont les suivants :

1. Le nombre de voisins allumés PN (Previous neighborhood) de P issus de la dernière itération défini par :

$$PN(P) = \sum_{i=1}^8 Pi$$

2. La fonction de voisinage actuel CN (Current Neighborhood) qui donne la valeur actuelle des points est définie par :

$$CN(P) = \sum_{i=1}^8 Pi \times Qi$$

3. La fonction qui calcule le nombre de transitions  $0 \rightarrow 1$  est définie en utilisant l'image actuelle et sa table des drapeaux comme suit :

$$Trans(P) = \sum_{i=1}^8 count(P)$$

où

$$count(P) = \begin{cases} 1 & \text{si } (Pi \times Qi) = 0 \text{ et } (P_{i+1} \times Q_{i+1} = 1) \\ 0 & \text{sinon} \end{cases}$$

sachant que  $P_9 = P_1$  et  $Q_9 = Q_1$ .

L'algorithme est décrit comme suit :

Step 1 : Pour chaque P de l'image, faire

- i. calculer PN(P), CN(P) et Trans(P) ;
- ii. si P est un pixel allumé qui satisfait la condition 1 et (la condition 2 ou la condition 3) alors mettre un drapeau dans la case correspondante dans la table des drapeaux.

fait

Step 2 : Supprimer tous les points avec drapeaux

Step 3 : Répéter les étapes 1 et 2 jusqu'à ce qu'il n'y ait plus de points à supprimer.

Les conditions suscitées sont :

condition 1 :  $(CN(P) > 1)$  et  $(CN(P) < 6)$

condition 2 :  $(Trans(P) == 1)$

condition 3 : P et ses voisins correspondent aux cas décrits dans la figure ??

0	0	0
1	<b>1</b>	0
0	1	X

X	0	0
1	<b>1</b>	0
0	1	0

0	1	X
1	<b>1</b>	0
0	0	0

0	1	0
1	<b>1</b>	0
X	0	0

FIGURE 3.5 – Les cas correspondant à la condition 3 de l’algorithme de Zhou et al. X est quelconque.

### 3.2.6 Datta & Parui [DattaParui1994]

L’algorithme de squelettisation de Datta & Parui se base sur les théorèmes suivants :

**Théorème 1** : Un point P est dit effaçable s’il n’est ni un point de fin ni un point critique.

**Théorème 2** : Un point P qui répond à la configuration (a) est un point de fin si, en considérant le voisinage de P comme dans la figure ?? :  $P1 + P2 + P4 + P5 + P6 + P8 = 0$ . On déduit de façon similaire les conditions si le point répond aux configuration (b)-(d).

**Théorème 3** : Un point P est dit point critique si son effacement provoque une discontinuité ou la formation d’un trou dans la forme. Un point P qui répond à la configuration (a) est un point critique si et seulement si ( $P1 = 0$  et  $P2 = 1$ ) ou bien ( $P4 = 0$  et  $P5 = 1$ ). On déduit de façon similaire les conditions si le point répond aux configuration (b)-(d).

L’algorithme peut être décrit comme suit :

**Step 1** : Effacer tous les pixels de la forme d’entrée dans les quatre passes (sous-étapes) suivantes :

- a) Effacer tous les pixels qui répondent à la configuration (a) et qui sont effaçables ;
- b) Effacer tous les pixels qui répondent à la configuration (b) et qui sont effaçables ;
- c) Effacer tous les pixels qui répondent à la configuration (c) et qui sont effaçables ;
- d) Effacer tous les pixels qui répondent à la configuration (d) et qui sont effaçables.

**Step 2 :** Répéter l'étape : Step 1 jusqu'à ce qu'il n'y ait plus de changement dans l'image de la forme, c'est-à-dire plus aucun pixel ne peut être effacé.

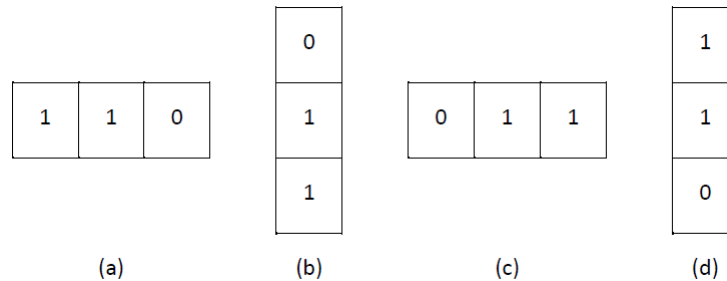


FIGURE 3.6 – Les configurations à considérer et le voisinage tel que envisagé par Datta & Parui

### 3.2.7 L'algorithme OPTA (One-pass Parallel Thinning Algorithm) [ChinWan1987]

La figure ?? montre les configurations utilisées dans cet algorithme. Nous avons 12 configurations  $3 \times 3$  (les configurations (a), (b) et (e) jusqu'à (n)), une configuration  $3 \times 4$  (la configuration (c)) et une  $4 \times 3$  (la configuration (d)).

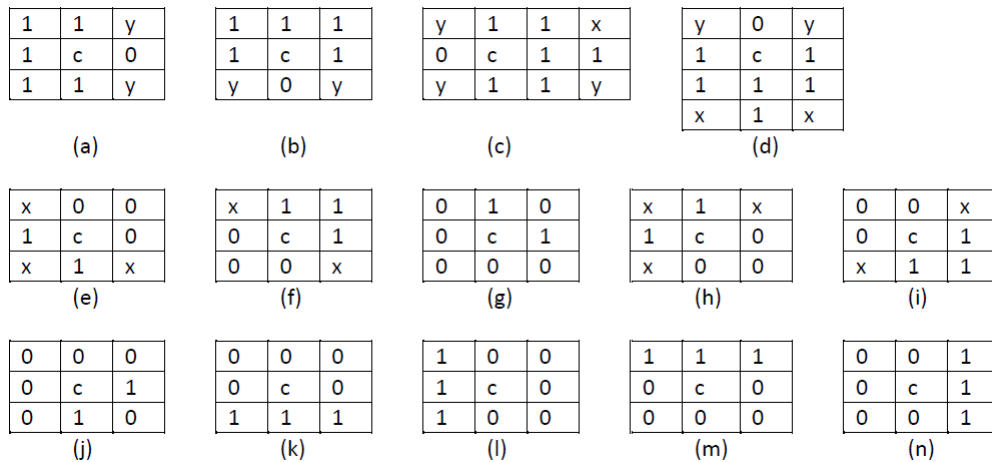


FIGURE 3.7 – Les configurations utilisées dans l'algorithme OPTA

Les symboles 'c', '0', '1' et 'x' dénotent le pixel en cours de test, un pixel blanc, un pixel noir et un pixel dont la valeur n'est pas importante respectivement. Ces symboles suivent les conventions communes.

Par contre le symbole 'y' apparaissant dans les configurations est un symbole spécial. Il n'apparaît jamais seul dans les configurations et au moins un pixel noté 'y' doit être un pixel blanc. L'algorithme OPTA utilise ces configurations pour supprimer les pixels de bord d'un objet épais couche par couche en parallèle. Plus précisément, l'algorithme applique un test d'appariement à chaque pixel de l'image de façon itérative pour déterminer si le pixel devrait changer de la valeur pixel de la forme à un pixel de fond (de 1 vers zéro). Ce pixel est effacé s'il matche avec l'une quelconque des configurations (a)-(n). Dans chaque itération, l'information sur le pixel courant et ses voisins sont ceux obtenus par l'itération précédente.

L'algorithme OPTA se décrit comme suit :

Input : une image binaire  $f^0$

Output : l'image du résultat de squelettisation

Étape 1  $i := 0$

Étape 2  $i := i + 1$ ;  $flag := false$

Étape 3 Tester chaque pixel de  $f^i$ . Si c'est un pixel noir et ses voisins matchent l'une des configurations (a) - (n), alors le changer en blanc et mettre le flag à true.

Étape 4 Si le flag est à faux, ce qui signifie que l'image est entièrement squelettisée, alors aller à l'étape 5 avec  $f^i$  comme résultat de la squelettisation. Sinon, aller à l'étape 2 pour exécuter la prochaine itération.

Étape 5 Retourner le résultat de la squelettisation.

Le but d'implémenter ces algorithmes a été de faire une comparaison entre eux pour décider lequel est le mieux adapté à la squelettisation du manuscrit arabe. Pour ce faire, nous citons ci-dessous les critères qui nous ont permis d'atteindre ce but.

### 3.2.8 Critères d'évaluation et de comparaison

Une étude comparative en bonne et due forme doit être basée sur des critères clairs et convaincants. Nous en avons utilisé de deux sortes : visuels et calculables.

### Critères visuels

- Le squelette doit avoir le squelette le plus fin possible (idéalement une épaisseur d'un pixel)
- Il doit respecter les connectivités de la forme originale.
- Il doit être centré dans la forme qu'il décrit.
- Il doit garder la forme initiale de l'objet qu'il décrit.
- Il doit garder tous les diacritiques et autres formes secondaires.

### Critères calculables

**Ratio de la longueur** Quand une image d'écriture manuscrite est bien squelettisée, sa longueur est inférieure ou égale à la moitié de la longueur de son contour. Le calcul du ratio longueur devrait donner une bonne information sur la précision de l'algorithme de squelettisation.

$$LR = (length(skeleton))/(length(contour))$$

Ce critère assure la présence minimale de barbules. D'après Zhou et al. [Zhouetal1995], un bon ration est autour de 0.4.

Les trois critères qui suivent sont aussi tirés de [Zhouetal1995].

**Mesure de l'épaisseur TM (Thinness measurement)** Un forme squelettisée devrait être amincie comme son nom l'indique. Mais dans quelle mesure peut-on dire qu'un objet dans une image est complètement aminci ou aminci à un certain degré? La mesure d'amincissement TM (The thinness measurement) donne une information sur le degré d'amincissement d'un objet dans une image.

Ce paramètre se calcule comme suit :

$$TM = 1 - TM1 + TM2 \tag{3.1}$$

$$TM1 = \sum_{i=1}^n \sum_{j=1}^m triangle\_count(p(i, j))$$

$$triangle\_count(p(i, j)) = P \times P(8) \times P(1) + P \times P(1) \times (2) + P \times P(2) \times P(3) + P \times P(3) \times P(4)$$

La fonction calcule l'épaisseur du squelette (voir ??).

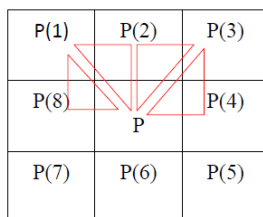


FIGURE 3.8 – Calcul de l'épaisseur du squelette TM.

$$TM2 = 4 \times (\max(m, n) - 1)^2$$

TM prend la valeur 0 lorsque l'image est complètement squelettisée. Il prend la valeur 1 si elle est complètement noire. Pour notre implémentation, nous l'avons modifié de telle sorte à donner la valeur 1 quand l'image est complètement squelettisée et 0 sinon. De la sorte, plus cette valeur est proche de 1 et plus l'image est fine. La nouvelle valeur de TM est donnée par :

$$TM = 1 - \frac{TM1}{TM2}$$

**Mesure de la connectivité CM (Connectivity measurement)** Ce paramètre est utilisé pour mesurer la connectivité du squelette résultant. Quand une image présente des discontinuités, elle présente des parties brisées là où il ne faut pas. De plus, quand on est en présence de lignes déconnectées, on a plus de points de terminaison et de points discrets. Ce nombre peut être utilisé comme un critère effectif de comparaison entre les algorithmes de squelettisation.

$$CM = \sum_{i=1}^n \sum_{j=1}^m S(p(i, j))$$

où

$$S(p(i, j)) = \begin{cases} 1 & \text{si } CN(p(i, j)) < 2 \\ 0 & \text{sinon} \end{cases}$$

$CN(p(i, j))$  est le nombre de voisins allumés du pixel  $p(i, j)$  dans l'image. Ce nombre quand il est élevé, dénote un squelette plus discontinu et donc un algorithme moins efficace.

**Mesure de sensibilité SM (Sensitivity measurement)** La mesure de sensibilité de l'algorithme se calcule par :

$$SM = \sum_{i=1}^n \sum_{j=1}^m S(p(i, j))$$

où

$$S(p(i, j)) = \begin{cases} 1 & \text{si Trans}(p(i, j)) > 2 \\ 0 & \text{sinon} \end{cases}$$

Cette mesure donne une information sur les petites discontinuités situées le long du squelette. Là, on n'est nullement concernés par les barbules ou les points extra qui peuvent s'éparpiller sur l'image. Ceux-ci peuvent être éliminés par des traitements préalables (prétraitements) pour les points extra ou post-traitements pour les barbules. Ce paramètre calcule le nombre de points de croisement dans le squelette de sortie. Ces points de croisement excédentaires sont causés par la vulnérabilité au bruit et l'inefficacité de l'algorithme d'amincissement. Moins on a de points de croisement, plus l'immunité de l'algorithme au bruit est élevée.

Trois autres paramètres ont été utilisés pour comparer entre les différents algorithmes implémentés et qui sont :

- Le temps d'exécution.
- Le nombre d'itérations qui ont été nécessaires pour l'obtention du squelette final.
- Le pourcentage des pixels supprimés de l'image.

Après implémentation des algorithmes de squelettisation suscités, comparaison visuelle et calculée (voir les résultats dans le chapitre ??), nous avons décidé d'apporter à l'algorithme de Zhang & Suen une amélioration que nous décrivons ci-dessous.

### 3.2.9 Post-traitement pour l'amélioration de l'algorithme de Zhang & Suen

L'algorithme de Zhang & Suen a présenté les meilleures performances visuelles. C'est pourquoi c'est celui que nous avons choisi pour proposer une amélioration dans les critères calculables. Comme le but final de notre besoin du squelette fourni par cet algorithme est son utilisation

future dans les étapes ultérieures d'un système de reconnaissance en l'occurrence la segmentation, nous trouvons que le critère le plus primordial est celui de pouvoir assurer :

- Un squelette d'épaisseur un pixel.
- Respect de la forme d'entrée et surtout,
- Nous avons noté que souvent, nous avons eu à corriger les directions des segments issus de la squelettisation de la forme d'entrée pour assurer un squelette qui permet de faciliter la segmentation.

Avant de détailler les solutions proposées, voyons de plus près les problèmes rencontrés dans les squelettes de Zhang & Suen.

#### Problèmes rencontrés dans les squelettes de Zhang & Suen

Quelques uns des problèmes rencontrés avec les squelettes issus de l'algorithme de Zhang & Suen sont illustrés dans la figure ??.

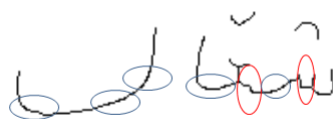


FIGURE 3.9 – Quelques uns des problèmes rencontrés avec les squelettes de Zhang & Suen. En bleu, des zones avec l'épaisseur supérieure à un pixel. En rouge, les segments présentent des portions verticales qui peuvent fausser la détection des zones de segmentation.

**Remarque** Ces problèmes ne sont pas seulement d'une importance visuelle : ils engendrent des problèmes lors de l'utilisation future d'un algorithme de segmentation. En effet, la plupart des algorithmes de segmentation, segmentent les squelettes dans des sections qui ont une largeur de 1 pixel. Les deux problèmes représentés sur la figure ?? créent des sections qui n'ont pas une largeur de 1 pixel dans des emplacements où éventuellement, la segmentation devrait être effectuée. C'est pourquoi il nous a paru utile de résoudre ces problèmes.

**Solution au problème 1 : Largeur du squelette**

Afin de résoudre ce problème qui revêt une grande importance lors de l'utilisation des squelettes dans l'étape de segmentation, nous avons appliqué les masques de la figure ?? pour que la largeur du squelette soit partout égale à un pixel.

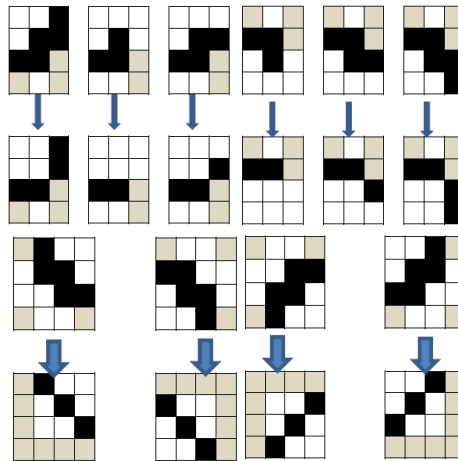


FIGURE 3.10 – Les masques utilisés pour résoudre les problèmes liés à l'épaisseur du squelette. Les cellules grises représentent les pixels à valeurs indifférentes.

Les résultats de ce traitement sont montrés dans la figure ??.

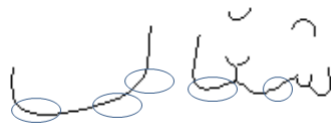


FIGURE 3.11 – Les résultats obtenus par le premier traitement concernant l'épaisseur du squelette.

**Solution au problème 2 : Correction des chemins du squelette**

Afin de corriger les chemins des squelettes, un ensemble d'autres masques (illustrés à la figure ??) a été appliqué et les résultats sont présentés dans la figure ??.

Et maintenant, allons vers le deuxième prétraitement auquel nous avons apporté une petite contribution. Il s'agit de la détection et redressement de la ligne de base d'écriture.

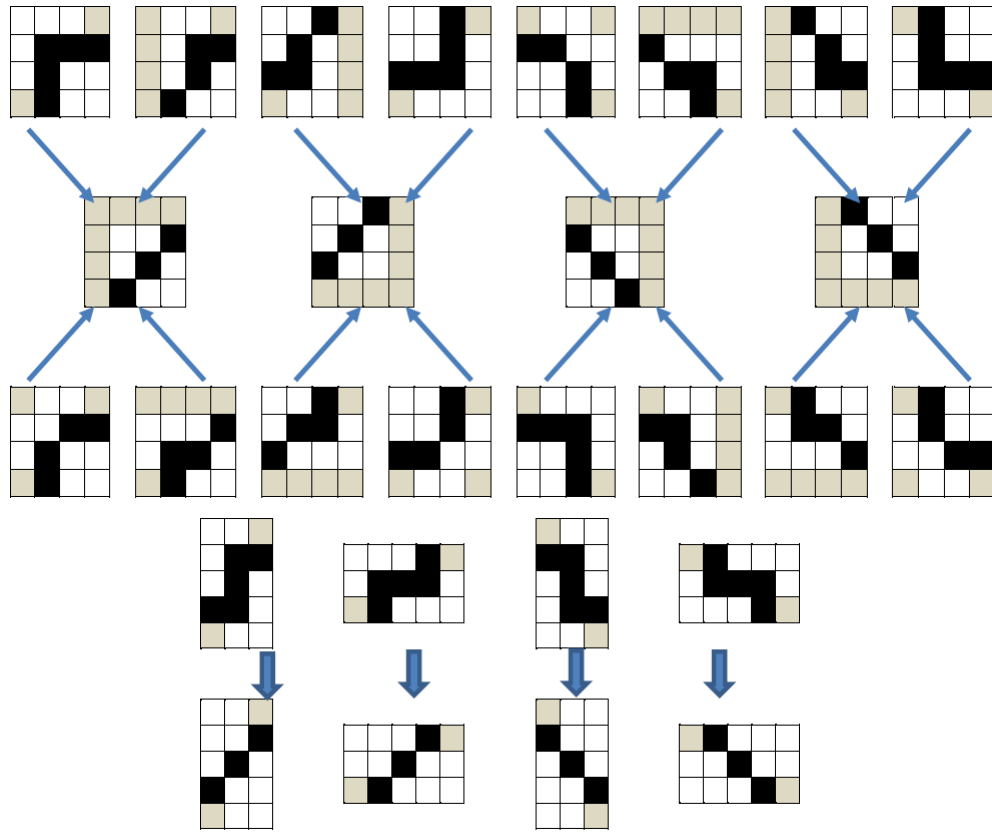


FIGURE 3.12 – Les masques utilisés pour la résolution du deuxième problème concernant la correction des chemins du squelette.



FIGURE 3.13 – Les résultats obtenus par le deuxième traitement concernant la correction des chemins du squelette.

### 3.3 Détection et redressement de la ligne de base

#### 3.3.1 Détection

Les techniques de détection de la ligne de base reposent en général sur le fait que cette dernière est composée d'une suite de pixels alignés joignant un nombre maximal de pixels noirs d'un mot, d'une partie de

mot ou d'une ligne de texte [Snoussi et al 2008]. Pour la détection de cette ligne fictive d'écriture, plusieurs méthodes sont offertes dans la littérature.

### Méthode de projection d'histogramme

**Principe de la méthode** Le principe de la méthode consiste à :

- Faire des rotations de l'image d'un angle qui varie en général entre -30 et +30 degrés avec un pas de 1 degré.
- On calcule pour chaque angle la position de la valeur maximale de l'histogramme horizontal de l'image.
- L'angle de rotation qui nous donne le pic maximum correspond à l'angle d'inclinaison de l'écriture.

Cette méthode présente deux variantes selon le fait qu'elle s'applique sur l'image brute ou bien l'image squelettisée.

**Image non squelettisée** La projection de l'histogramme horizontal de l'image d'un texte est souvent caractérisée par un pic correspondant à la partie principale du texte. La position de ce pic donne l'information sur la position de la ligne de base d'écriture. Cette méthode est facile à implémenter et très utilisée dans le domaine.

La figure ?? montre l'image d'un mot de la base IFN/ENIT avec la projection de l'histogramme horizontal avec un pic situé sur la ligne virtuelle d'écriture.

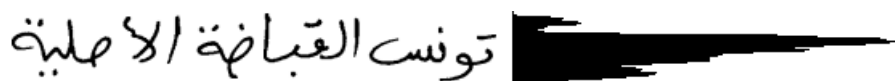


FIGURE 3.14 – Un mot et la projection de son histogramme horizontal

**Image squelettisée** La méthode de projection d'histogramme de l'image d'écriture est appliquée telle qu'elle après squelettisation de cette dernière. La figure ?? en montre un exemple.



FIGURE 3.15 – L'image squelettisée d'un mot et la projection de son histogramme horizontal

### Méthode de la transformée de Hough

La transformée de Hough s'applique sur l'image du contour de l'écriture. La figure ?? montre l'image d'un mot et celle de son contour.

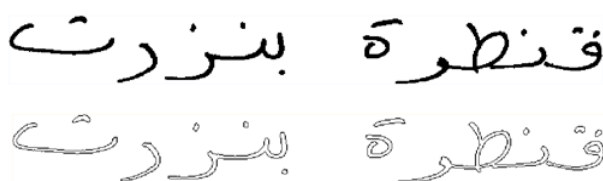


FIGURE 3.16 – Un mot et l'image de son contour

Nous avons utilisé sur les images d'écriture cette transformée dans le but de détecter la ligne de base. Elle nous permet de transformer les points caractéristiques de l'image vers l'espace de Hough. Ce dernier est représenté par un tableau 2D appelé accumulateur qui contient les cellules d'accumulation. Ces cellules sont remplies par les votes de pixels appartenant à une droite donnée de paramètres  $(\rho, \theta)$ . Chaque maximum obtenu renseigne directement sur les paramètres spécifiques de la droite détectée. La ligne de base est repérée par la présence de maximum dans l'accumulateur.

**Principe de la méthode** Pour chaque point des contours, on considère toutes les droites qui passent par ce point. Les droites les plus citées sont celles présentes sur l'image.

- **Dans l'espace cartésien**

Dans l'espace cartésien  $(x,y)$ , une droite a une équation :  $y = ax + b$ . Si l'on fixe un point particulier  $P(x_p, y_p)$ , toutes les droites passant par ce point, auront la même équation avec  $a$  et  $b$  variables et  $x_p$  et  $y_p$  constants. Cette équation s'écrira :  $b = -x_p a + y_p$ . C'est aussi l'équation d'une droite dans l'espace  $(a,b)$ . La figure ?? montre l'équivalence entre les deux espaces.

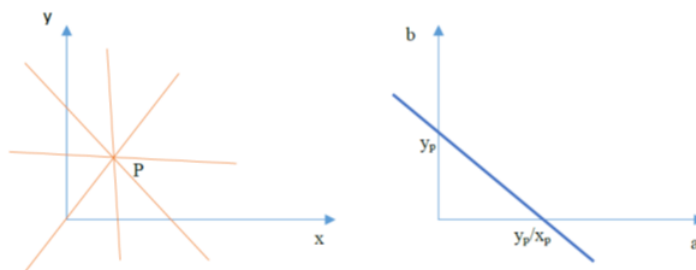


FIGURE 3.17 – Équivalence entre les espaces  $(x,y)$  et  $(a,b)$

De façon similaire, tous les points situés sur la même droite D dans l'espace  $(x,y)$  sont représentés par des droites qui passent toutes par le même point dans l'espace  $(a,b)$ . Ce point  $(a_p, b_p)$  donne les paramètres recherchés de l'équation de la droite D :  $y = ax + b$  (voir figure ??).

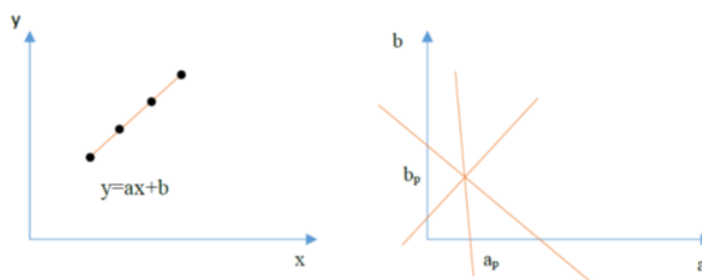


FIGURE 3.18 – Une droite dans l'espace  $(x,y)$  et son équivalent dans l'espace  $(a,b)$

### • Dans l'espace polaire

A un point  $P(x,y)$  des contours, on associe la sinusoïde  $\rho = x \cos \theta + y \sin \theta$  dans l'espace  $(\rho, \theta)$ . Les points appartenant à une même droite dans l'image, seront représentés dans l'espace  $(\rho, \theta)$  par un ensemble de sinusoïdes passant par un même point dans cet espace (voir figure ??).

Et donc, à chaque point de cet espace correspond une droite de l'image. Le point sur lequel passe le maximum de sinusoïdes donne les informations  $(\rho$  et  $\theta)$  concernant la ligne de base d'écriture.

### Méthode de la minimisation d'entropie

Cette méthode est basée sur la projection horizontale de l'image du contour sur l'axe des ordonnées selon plusieurs axes inclinés. Les angles

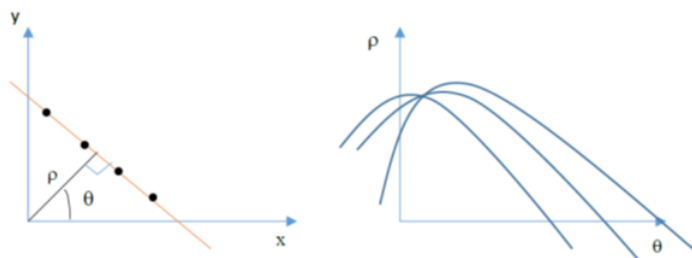


FIGURE 3.19 – Une droite dans l'image et son équivalent dans l'espace  $(\theta, \rho)$

inclinés changent de  $-\theta$  à  $+\theta$  avec un pas de 1. Plusieurs projections sont effectuées selon la valeur de  $\theta$ . Pour chacune de ces projections, la densité de chaque histogramme est calculée. L'entropie est une mesure d'information fournie par la formule suivante :

$$E = -\sum_{i=1}^{nbl} P_i \log(P_i) \quad (3.2)$$

avec  $P_i = \frac{N_i}{N}$

nbl est le nombre de lignes dans l'image du contour,  $N_i$  représente le nombre de pixels dans la ligne numéro  $i$ ,  $N$  le nombre total de pixels dans le contour du mot et  $P_i$  la probabilité de l'occurrence des pixels dans la ligne numéro  $i$ . L'entropie  $E$  est maximale si toutes les probabilités  $P_i$  sont égales. Elle est minimale si toutes les probabilités sont nulles exceptée une. La direction choisie est celle qui représente un histogramme de projection du mot le plus compact ayant une entropie minimale.

#### Méthode du contour inférieur

La méthode du Cont-Inf est basée sur la détection de la ligne de base par la méthode de projection horizontale (sans rotation), avec un entraînement de cette ligne d'une marge vers le bas et d'une autre marge vers le haut, puis sur l'extraction des points essentiels du contour inférieur du mot. A partir d'un premier point, nous parcourons le contour inférieur en vue d'extraire les minima. Nous définissons comme minima tous les points  $(i, j)$  du contour inférieur ayant comme voisin immédiat un pixel noir avec les coordonnées  $(i - 1, j - 1)$ ,  $(i - 1, j)$  ou  $(i - 1, j + 1)$ , voir la figure ???. La ligne de base est déduite à partir de la moyenne des

ordonnées des minima qui sont supérieurs à la ligne de base inférieure, et inférieurs à la ligne de base supérieure provenant de la méthode de projection horizontale. D'où la méthode hybride Cont-Inf. La figure ?? illustre la ligne de base de quelques images par la méthode du Cont-Inf.

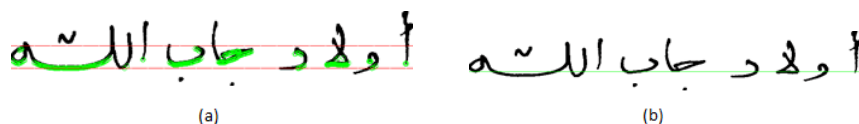


FIGURE 3.20 – (a) Représentation des points minima du contour. Les lignes en rouge sont la ligne supérieure et la ligne inférieure provenant de la méthode de projection horizontale. (b) La ligne en vert est la ligne de base extraite par la méthode de Cont-Inf

L'algorithme de cette méthode est :

1. Faire la projection horizontale sans rotation et extraire la ligne de base.
2. Entrainer la ligne de base horizontale d'une marge vers le haut et d'une marge vers le bas pour construire une bande de base qui délimite les points minima.
3. Extraire les positions des points minima.
4. Calculer la position moyenne de ces points et tracer la ligne de base à partir de cette position.

#### Méthode Det-Boucles

Dans cette méthode on extrait les points de détournement du contour inférieur où le trait de l'écriture va changer de sens vers le haut et les points de boucles dont la position est minimale. En premier lieu, on extrait la ligne de base par la méthode de projection, puis entraîne cette ligne d'un pourcentage vers le haut et d'un pourcentage vers le bas pour construire une bande de base pour éviter les descendants et les signes diacritiques. Nous définissons un point de détournement vers le haut dans le sens droite-gauche, comme tout point  $(i, j)$  du contour inférieur ayant comme voisin un pixel noir avec les coordonnées  $(i, j+1)$ ,  $(i-1, j-1)$ , et un pixel blanc avec les coordonnées  $(i, j-1)$ ,  $(i+1, j)$ ,  $(i+1, j+1)$ , ou un pixel noir avec les coordonnées  $(i, j+1)$ ,  $(i-1, j-2)$ ,  $(i-1, j-3)$ , ou  $(i, j+2)$ ,  $(i, j+3)$ ,  $(i, j+4)$ , et un pixel blanc avec les coordonnées

$(i+1, j+2)$ ,  $(i+1, j+3)$ ,  $(i+1, j+4)$ , (Figure ??). Quant-aux boucles, on parcourt tous les points des boucles (s'il y en a) et on garde le point dont la première coordonnée de la position  $(i, j)$  est minimale (Figure ?? (a)). Pour déduire la ligne de base, on partitionne le mot en deux parties (droite et gauche), on extrait les points de détournement et on calcule la position moyenne de ces points dans chaque partie, de même on garde pour chaque partie un point de boucle, on calcule la moyenne entre les positions des points de boucles et la position moyenne des points de détournement pour chaque partie, on obtient deux positions qui définissent la ligne de base (Figure ?? (b)).

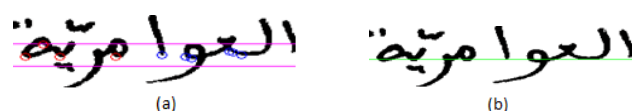


FIGURE 3.21 – (a) Les lignes en rose représentent la bande de base extraite par la méthode de projection, les marques en bleu et en rouge représentent les points de détournement des parties droite et gauche du mot respectivement. (b) La ligne de base extraite par la méthode Det-Boucle.

Cette méthode est implémentée selon l'algorithme suivant :

1. Faire la projection et extraire la ligne de base.
2. Entrainer la ligne de base d'un pourcentage vers le haut et vers le bas pour construire une bande de base qui délimite les points de détournement.
3. Partitionner le mot verticalement au milieu en deux parties.
4. Extraire les positions des points de détournement et calculer leur moyenne dans chaque partie.
5. Extraire la position des points de boucles dont la première coordonnée est minimale dans chaque partie (s'il en existe).
6. Calculer la moyenne entre la position des points de détournement et la position du point de boucle dans chaque partie pour tracer la ligne de base.

### 3.3.2 Critères d'évaluation

#### Critères visuels

Contrairement à ce qui est observé pour l'évaluation des algorithmes de squelettisation, les critères visuels pour l'évaluation de la ligne de base s'appuient globalement sur les connaissances a priori de l'écriture en question. En effet, sans ces connaissances et à titre d'exemple, une personne ne pourra pas dire laquelle des deux lignes de base de la figure ?? est la plus correcte.

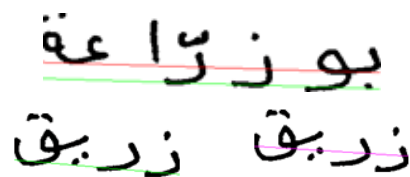


FIGURE 3.22 – Deux exemples où des lignes sont difficiles à évaluer sans connaissances a priori de l'écriture arabe.

#### Critères calculables

Nous considérons que le résultat de détection de ligne de base par l'une quelconque des méthodes est acceptable si :

$$\begin{cases} |y_1 - y_{b1}| \leq 7 \\ et \\ |y_2 - y_{b2}| \leq 7 \end{cases}$$

Où :

- $y_1, y_2$  sont les positions de la ligne de base extraite par une des six méthodes suscitées ;
- $y_{b1}, y_{b2}$  sont les positions de la ligne de base de la base de données IFN/ENIT extraites des fichiers ground truth procurés pour chaque image de cette base.

### 3.3.3 Redressement

Les trois méthodes suscitées donnent toutes comme informations sur la ligne de base, l'angle d'inclinaison de cette dernière par rapport à l'axe

des abscisses. Pour voir redresser l'image s'il y a lieu, il suffit de faire une rotation de cette dernière d'un angle inverse. La figure ?? montre un redressement effectué sur un mot incliné de la base IFN/ENIT à l'aide de la projection d'histogramme avec ses deux variantes. Ceci peut-être réalisé par :

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

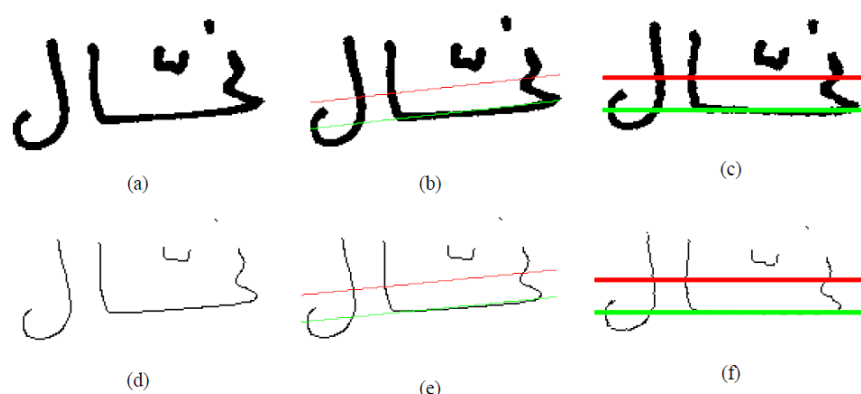


FIGURE 3.23 – Détection et redressement de la ligne de base d'un mot a-c) sur l'image originale, d-f) sur le squelette de l'image

La validation de tout système de reconnaissance du manuscrit basé-segmentation nécessite un apprentissage sur des lettres pré-segmentées.

### 3.4 Mise en place d'une base de données pour les caractères arabes manuscrits pré-segmentés

Pour la création de la base de données nous avons préparé deux types de formulaires le premier servira à recueillir des informations sur le scripteur (âge, sexe, profession, établissement d'étude), le second sera utilisé pour récolter les caractères manuscrits (voir figures ?? et ??).

Les données collectées proviennent de 9 scripteurs de sexe différents et d'âge variant entre 13 et 48 ans avec un niveau intellectuel allant du collège à l'université au niveau de la wilaya d'Alger. Chaque scripteur reçoit un formulaire contenant 54 cases où il écrit des caractères dans les 18 cases qui lui sont réservées (voir figure ??).

Nom	
Lieu études avant université	
Lieu études universitaires	
Etablissement universitaire	
Age	

FIGURE 3.24 – Formulaire d'information

Ghin isolé					
ġ	ġ	ġ	ġ	ġ	ġ
ġ	ġ	ġ	ġ	ġ	ġ
ġ	ġ	ġ	ġ	ġ	ġ
ġ	ġ	ġ	ġ	ġ	ġ
ġ	ġ	ġ	ġ	ġ	ġ
ġ	ġ	ġ	ġ	ġ	ġ
ġ	ġ	ġ	ġ	ġ	ġ
ġ	ġ	ġ	ġ	ġ	ġ

FIGURE 3.25 – Formulaire d'acquisition des données



FIGURE 3.26 – Un spécimen de caractères manuscrits arabes dans la base TBK. (a) Milieu de PAWs, (b) à la fin, (c) isolés, (d) au début des PAWs.

### 3.5 Conclusion

Dans ce chapitre, il a été question de voir de plus près ce qui a été fait dans le module de prétraitements dans un système de reconnaissance du manuscrit arabe. Nos contributions se résument comme suit :

- Concernant la squelettisation :
  - Nous avons implémenté sept des algorithmes les plus connus dans la littérature concernant la squelettisation du manuscrit en vue de les tester sur le manuscrit arabe.
  - Nous avons procédé à une comparaison entre les algorithmes moyennant des critères visuels et d'autres du type calculable.
  - Nous avons proposé une phase de post-traitement à l'algorithme qui a présenté les meilleures caractéristiques à savoir l'algorithme de Zhang & Suen.
- Concernant la détection-redressement de la ligne de base, nous avons

implémenté quatre des méthodes les plus usuelles de détection de la ligne de base et nous avons adapté deux méthodes trouvées dans [Snoussi**etal**2008].

Les résultats de ces études sont explicités dans le chapitre ??.

## Contribution II : Extraction de caractéristiques-Classification

### 4.1 Introduction

Dans ce chapitre, il s'agira de résumer quelques une des méthodes d'extraction de caractéristiques que nous avons implémentées ainsi que les classifieurs utilisés. Une méthode de combinaisons de classifieurs et des caractéristiques est décrite. Celle-ci a donné des résultats qui ont concurrencé ceux de l'état de l'art [Tamenetal2017]. Les résultats sont légués au chapitre ??.

### 4.2 Extraction de caractéristiques

#### 4.2.1 Analyse en ondelettes [Kechad2016]

Lors de notre travail avec l'analyse en ondelettes, nous avons été confrontée à quelques grands problèmes, à savoir :

1. Avec quelles familles travailler ?
2. Quels sont les coefficients à prendre en considération lors de cette analyse ?
3. L'analyse en ondelettes nécessitant des images de tailles normalisées, quelle est la meilleure taille à prendre ?

La littérature n'offre pas de réponses concluantes aux questions suscitées. Nous avons donc mené nos propres expériences dans le but de

trouver la meilleure configuration (famille, taille, coefficients) qui est à même de mieux discriminer notre manuscrit arabe.

Comme le but final est un système capable de travailler avec ou sans segmentation, les deux bases de données que nous avons choisies pour ces expérimentations sont :

- La base TBK qui contient les images de caractères pré segmentés avec les différentes formes possibles selon les positions de ces derniers dans la PAW. Le travail avec cette base, permettra dans le cas où l'on est amenée à travailler avec l'analyse en ondelettes avec un système avec segmentation explicite ou implicite, un apprentissage sur cette base s'avèrera nécessaire.
- La base IFN/ENIT constituée de noms de villes tunisiennes contenant entre une et une dizaine de PAWs. Cette base représente le maximum de difficultés dans l'écriture. En effet, elle constitue, par excellence, la base la plus difficile dans le domaine.

Pour répondre aux trois questions suscitées, nous avons procédé comme suit :

1. Les familles d'ondelettes avec lesquelles nous avons décidé de travailler sont :

- La famille Daubechies,
- La famille Coiflet,
- La famille Symlet.

Le choix de ces familles a été motivé surtout par la popularité de ces familles dans le domaine de reconnaissance des formes en général.

2. Les tailles que nous avons considérées pour les deux bases sont les suivantes :

- Pour la base TBK :
  - $64 \times 64$
  - $96 \times 96$
  - $128 \times 128$
  - $160 \times 160$
- Pour la base IFN/ENIT : Le fait d'avoir déjà largement travaillé avec cette base nous a permis de limiter les tailles prises pour cette analyse, à savoir :

- $32 \times 128$
- $64 \times 256$ .

En utilisant cette base et pour avoir un maximum d'information sur les images et pour simuler un effet de segmentation, nous avons appliqué un système de fenêtrage à base de fenêtres glissantes à dimensions et chevauchements variables. Le fenêtrage a consisté à découper le mot en fenêtres avec une largeur variable avec ou sans chevauchement. Le chevauchement permet d'éviter toute perte d'information due au découpage. Le fenêtrage est effectué de la droite vers la gauche pour suivre le sens de lecture/écriture du script arabe. La figure suivante ?? illustre le fenêtrage appliqué à un mot de la base IFN/ENIT.



FIGURE 4.1 – Fenêtrage sur un mot de la base IFN/ENIT avec un chevauchement de 3 pixels.

3. Les paramètres que nous avons considérés dans cette étude sont :
- la moyenne arithmétique,
  - l'écart type,
  - l'énergie,
  - les valeurs propres des images de décomposition.

#### Paramètres pris en compte

Pour chaque niveau de décomposition nous avons calculé un des paramètres statistiques suscités.

- La taille du vecteur de caractéristiques varie selon le paramètre statistique pris, comme suit :
  - Pour les paramètres à valeur réelle unique, comme la moyenne, l'énergie, l'écart-type, cette taille va être :

$$T = 4 * n_{dec}$$

où 4 représente le nombre d'images de décomposition par niveau,  $n_{dec}$  est le niveau de décomposition.

- Pour les paramètres à plusieurs valeurs comme la décomposition en composantes principales, la taille du vecteur caractéristique se calcule par :

$$T = 4 * k * n_{dec}$$

où  $k$  représente le nombre de valeurs propres prises pour chaque image de décomposition.

### Moyenne Arithmétique

#### 1. Définition

La moyenne arithmétique (ou moyenne empirique) d'une série statistique est la somme des valeurs divisée par le nombre de valeurs.

#### 2. Application de la moyenne

Après transformation par ondelette de l'image, nous avons calculé la moyenne de chaque composante de la transformation comme suit :

$$m_d^k = \frac{1}{n * m} \sum_{i=1}^n \sum_{j=1}^m x_{ij}^k \quad (4.1)$$

où  $n \times m$  représente le nombre de coefficients contenus dans chaque image de décomposition au niveau de décomposition  $d$ ,  $k = 1, \dots, 4$  représente l'image issue de la décomposition ( $k=1$ , approximation,  $k=2$ , détails horizontaux,  $k=3$ , détails verticaux et  $k=4$  détails diagonaux).

### Écart-Type

#### 1. Définition

Il consiste en une mesure de dispersion des données. Il est défini comme la racine carrée de la variance, ou de manière équivalente comme la moyenne quadratique des écarts par rapport à la moyenne. L'écart-type sert donc à mesurer la dispersion d'une série statistique autour de sa moyenne. La variance permet de combiner toutes les

valeurs à l'intérieur d'un ensemble de données afin d'obtenir la mesure de dispersion. La variance (symbolisée par  $\sigma^2$ ) et l'écart-type (la racine carrée de la variance, symbolisée par  $\sigma$ ) sont les mesures de dispersion les plus couramment utilisées.

## 2. Calcul

Il se calcule comme suit :

$$\sigma_{X^k} = \sqrt{E[(X^k - E[X^k])^2]} \quad (4.2)$$

où  $X^k$  représente l'image de décomposition.

**Énergie :** Pour une décomposition bidimensionnelle en ondelettes, ce paramètre représente le pourcentage d'énergie correspondant aux coefficients d'approximation et de détails horizontaux, verticaux et diagonaux.

Ce paramètre se calcule comme suit :

$$E = \sum_{i=1}^N \sum_{j=1}^M Im(i, j)^2$$

Où :  $N \times M$  représente la taille de l'image et :  $Im(i, j)$  la valeur de l'image à la position  $(i, j)$ .

**Vecteur Propres, Valeurs propres :** Les valeurs propres d'une matrice A sont les valeurs  $\lambda$  pour lesquelles nous avons :

$$\det(A - \lambda * I) = 0$$

où I est la matrice identité de même rang que la matrice. Initialement utilisés pour étudier les principaux axes de mouvement des corps rigides en rotation, les valeurs et vecteurs propres ont aujourd'hui un large éventail d'applications, par exemple dans l'analyse de la stabilité des systèmes, l'étude et analyse des vibrations, étude des orbites atomiques, la reconnaissance faciale, etc.

**Combinaison des paramètres statistiques :** On ne s'est pas contenté de prendre chaque paramètre statistique individuellement, on a concaténé

TABLE 4.1 – Les ensembles considérés

Total	Train	Test	Nb classes	Représentativité.	Variabilité
3597	2409	1188	10	345-381	0.09
7560	5065	2495	22	306-381	0.20

les paramètres statistiques entre eux en prenant en compte toutes les combinaisons possibles.

La classification lors de l'utilisation de l'analyse en ondelettes a été effectuée moyennant les réseaux de neurones multi-couches avec l'algorithme d'apprentissage de rétro-propagation du gradient.

#### 4.2.2 Moments orthogonaux

Les moments orthogonaux pour lesquels nous montrons les résultats, sont les moments de Zernike, de Legendre, de Chebyshev et ceux de Krawchouk. Le pouvoir discriminatoire de ces moments est étudié en utilisant la méthodologie suivante :

- L'utilisation des moments nécessitant la normalisation de la taille des images, nous avons expérimenté différentes tailles parmi elles nous avons :  $32 \times 128$ ,  $64 \times 256$ ,  $80 \times 80$ .
- Nous avons aussi étudié l'effet du nombre de classes et la taille des ensembles. L'efficacité des quatre moments orthogonaux (OM pour Orthogonal Moments) a été testée d'abord sur plusieurs ensembles d'images prises de tous les sets  $a$ ,  $b$ ,  $c$ ,  $d$  et  $e$  de la base IFN/ENIT.

Nous avons d'abord considéré deux ensembles de données de la base de données IFN/ENIT [Pechwitzetal2002]. Le premier test a été réalisé sur un ensemble de 3597 images, constitué de 10 classes de représentativité minimum de 345 images et maximum de 381 images. La table ?? résume les deux premiers ensembles utilisés dans l'expérimentation. Pendant le processus de reconnaissance, environ 67 % (2/3 du total des images) des images dans l'ensemble considéré sont utilisés pour l'apprentissage et le reste pour le test.

La variation entre la représentativité des différents ensembles est calculée comme  $(max - min)/max$  où  $max$  est le nombre maximum

d'images par classe dans l'ensemble et *min* est le minimum. On peut remarquer que le premier ensemble présente les meilleures caractéristiques pour ce genre de tests : le nombre minimum de classes, la représentativité maximale (nombre d'images par classe) et la variabilité minimale dans cette dernière. Ainsi, les résultats sur cet ensemble peuvent être considérés comme significatifs. Nous avons utilisé cet ensemble dans le but de comparaison effective entre les quatre moments utilisés dans ce travail. Nous avons varié l'ordre des moments de 7 à 32 de la manière suivante : 7 - 10 - 15 - 25 - 32.

Nous avons utilisé le classifieur K-NN dans le but de comparer les différents moments dans leur pouvoir de discrimination des images de mots manuscrits arabes parce que dans ce classifieur, les résultats ne dépendent pas d'une initialisation aléatoire contrairement aux autres classifieurs utilisés dans ce travail. Néanmoins, nous avons entraîné le classifieur des forêts aléatoires (RF) dans ce même contexte afin de valider les résultats. Pour ce dernier classifieur, nous avons varié le nombre d'arbres de 100 à 10 000 avec un incrément de 100. Le nombre de variables échantillonnées au hasard comme candidats pour chaque fraction varie de 2 à la racine carrée du nombre de variables par incrément de 2. Pour réduire l'effet de l'aléa dans les résultats nous avons répété chaque configuration 10 fois et estimé la moyenne des résultats obtenus au cours des différentes exécutions.

#### 4.2.3 Caractéristiques statistiques basées sur le contour (SCF)

Les moments étant des caractéristiques basées sur tous les pixels constituant l'image, nous avons pensé que l'ajout de caractéristiques basées essentiellement sur le contour des images pourraient améliorer le pouvoir discriminatoire de ces derniers. Le but de ces caractéristiques est d'extraire des informations locales basées sur le contour des images de mots.

Ceci est réalisé en suivant différentes étapes comme suit :

1. D'abord nous estimons les contours des images de mots. Comme les images traitées sont déjà binarisées, il suffit de colorer les pixels écrits qui sont en contact avec le background. Le reste des pixels écrits sont

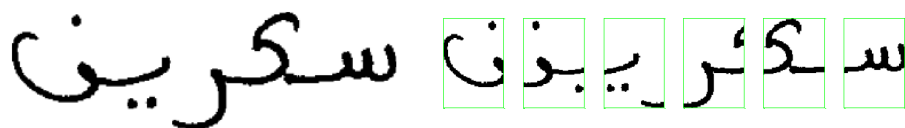


FIGURE 4.2 – Sliding window 60 pixels width and 12 pixels overlap

considérés comme background et uniquement les pixels colorés sont considérés dans les futurs traitements.

2. On calcule la densité de l'image écrite comme le rapport entre la somme des pixels écrits et la surface de l'image.
3. Des fenêtres glissantes de tailles et chevauchements différents sont utilisées pour capturer l'information locale de l'écriture. Cette technique consiste à déplacer une fenêtre rectangulaire de la droite vers la gauche sur l'image. Ce déplacement peut avoir un chevauchement entre la nouvelle position et la précédente.

La figure ?? montre un exemple de fenêtres glissantes de 60 pixels de largeur et 12 pixels de chevauchement appliquées à une image de notre base de données. Pour chaque fenêtre, nous sauvegardons la position relative du centre de masse, normalisée par la hauteur et largeur de la fenêtre.

4. Pour chaque fenêtre, nous sauvegardons le nombre de points d'intersection entre le contour de l'écriture et les lignes dessinées à partir du centre de masse selon les huit directions de Freeman.
5. La distance du point le plus éloigné dans chaque direction normalisé par la largeur de la fenêtre est aussi sauvegardé.
6. Pour plus de détails, chaque fenêtre est subdivisée en  $m \times n$  blocs (voir figure ??, où nous montrons la dernière fenêtre divisée en  $2 \times 4$  blocs). Dans chaque bloc, nous estimons la densité de l'écriture et le nombre de transitions 0 – 1 dans les deux directions horizontale et verticale.

Toutes les valeurs sauvegardées durant ces étapes, constituent le vecteur caractéristique qui décrit une image à classifier. La figure ?? montre la façon dont les caractéristiques SCF sont extraites d'une image de

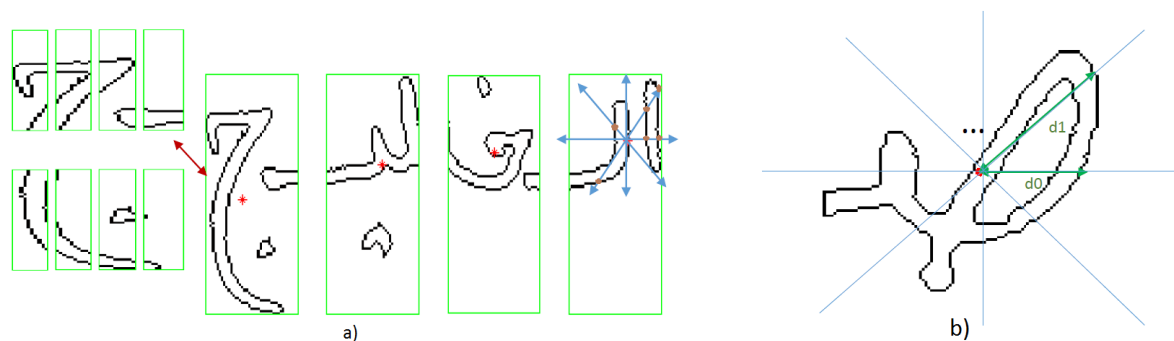


FIGURE 4.3 – Extraction of SCF features : a) four windows and  $2 \times 4$  blocks per window  
 b) distances  $d_i, i = 1, \dots, 8$  are saved after normalization

la base IFN/ENIT. Ces caractéristiques nous permettent de distinguer entre des mots qui sont similaires dans leurs formes globales mais avec un contenu en caractères différent. Un exemple est montré en figure ?? où la distance euclidienne est calculée entre deux images de mots en utilisant leurs vecteurs caractéristiques : Le vecteur avec les CM et le vecteur avec les caractéristiques SCF. Les valeurs obtenues sont respectivement 0.8742 et 1.9704 montrant que les caractéristiques SCF sont plus discriminantes dans ce cas. Ceci peut donc améliorer l'information procurée par les moments.

### 4.3 Classification

Notre contribution concernant la classification peut être résumée en ces points :

- Comparaisons : Les comparaisons permettent de décider du :
  - meilleur classifieur en utilisant un même type de caractéristiques avec des classifieurs différents.
  - meilleur type de caractéristiques en utilisant différents types de caractéristiques dans les mêmes conditions avec un même classifieur.
- Combinaison : Comme son nom l'indique, la combinaison permet de créer des systèmes hybrides en utilisant :
  - une combinaison de caractéristiques de différents types.

- une combinaison de classifieurs.
- une combinaison caractéristiques/classifieurs.

Ces différents points sont succinctement décrits dans ce chapitre et les résultats sont explicités dans le chapitre ??.

### 4.3.1 Comparaisons

Les différentes comparaisons que nous avons effectuées se situent dans les deux niveaux et peuvent se résumer comme suit :

- Sur le plan caractéristiques :
  - DCT VS. SVD avec les réseaux de neurones.
  - Les moments de Krawchouk, Legendre, Zernike et Chebyshev avec le classifieur KNN et le classifieur random forest.
- Sur le plan classifieur : Nous avons utilisé les moments de Chebyshev ainsi que les caractéristiques SCF avec :
  1. les réseaux de neurones ;
  2. les SVM ;
  3. les ELM.

### 4.3.2 Combinaison

#### Combinaison de caractéristiques

Nous avons combiné plusieurs types de caractéristiques, à savoir :

- Les moments de Legendre et les moments de Chebyshev.
- La transformée en cosinus discrète DCT et la décomposition en valeurs singulières SVD.

#### Combinaison de caractéristiques/classifieurs

Dans le but d'améliorer le taux de reconnaissance à l'aide de la combinaison de classifieurs, nous avons utilisé les trois classifieurs MLP, SVM et ELM. Chaque classifieur est entraîné avec deux sets de caractéristiques CM et SCF séparément résultant en six classifieurs à savoir :  $MLP_{CM}$ ,  $MLP_{SCF}$ ,  $SVM_{CM}$ ,  $SVM_{SCF}$ ,  $ELM_{CM}$  and  $ELM_{SCF}$ .

Pour accomplir de bons résultats, nous avons procédé par l'utilisation de

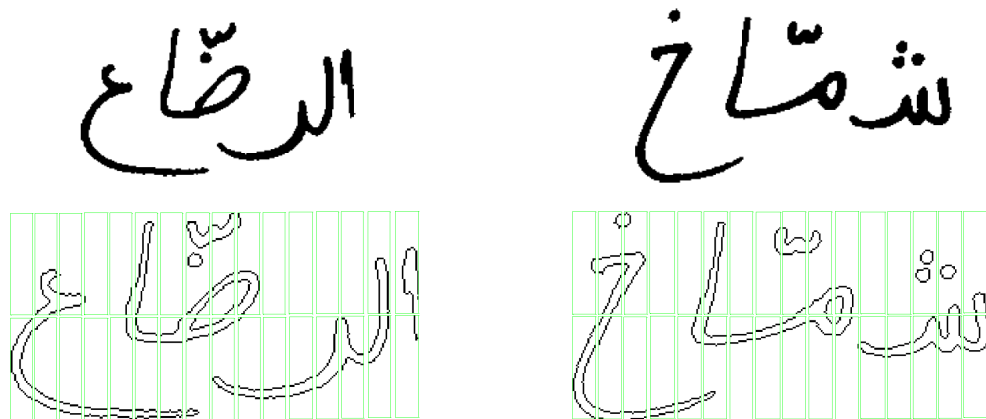


FIGURE 4.4 – SCF to differentiate words that are globally similar. The Euclidean distances are respectively 0.8742 (CM) and 1.9704 (SCF)

plusieurs méthodes de combinaison de ces six classifieurs. La première technique utilisée dans ce sens a consisté à entraîner deux classifieurs : un MLP et un ELM. Ensuite, nous proposons plusieurs règles de combinaison des six classifieurs, une adaptation de la Borda Count et plus loin, une combinaison basée sur la *théorie des fonctions de croyance* de Dempster-Shafer. La manière dont nous avons accompli ces combinaisons est expliquée ci-dessous. Un second niveau de combinaison est ensuite discuté, voir la figure ??.

La sortie d'un classificateur  $s$  pour une image d'entrée  $x$  est un vecteur de  $n$  éléments, avec  $n$  le nombre de classes dans le dataset :  $y_s = (y_{1s}, \dots, y_{ns})$  où  $y_{is}$  correspond à une probabilité d'appartenance de l'entrée par rapport à la classe  $i$ . L'indice de la valeur maximale dans  $y_s$  donne le rang de la classe d'appartenance estimée par le classificateur  $s$  pour cette entrée.

1. MLP and ELM : La combinaison avec MLP et ELM des six classifieurs est réalisée comme suit :
  - La sortie de chaque classificateur est transformée en un vecteur de  $1 \times (n \times N)$  où  $N$  est le nombre d'images dans l'ensemble d'apprentissage. Les sorties des six classifieurs sont concaténées pour former un tableau  $6 \times (n \times N)$ . Le même processus est fait aux

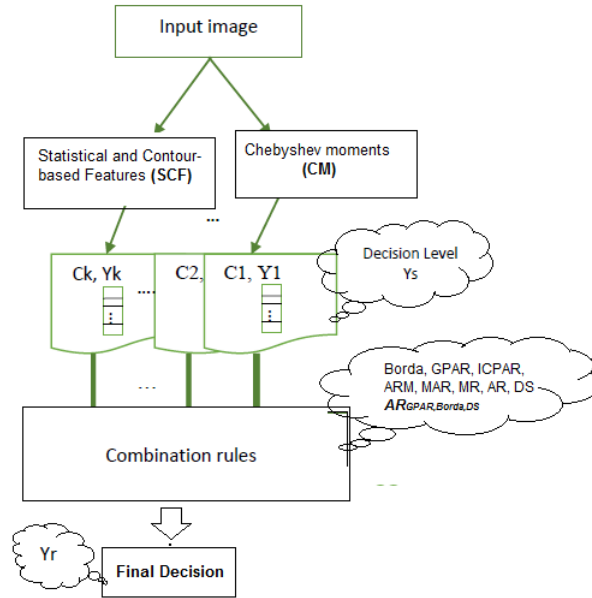


FIGURE 4.5 – Combination of the  $C_k$  classifiers in the Decision Level

cibles pour avoir un tableau de  $1 \times (n \times N)$ . Un MLP et un ELM d'architecture  $(6 \times nc_p (nc_m) \times 1)$  sont ensuite entraînés avec ces données où 6 est le nombre de classificateurs (neurones d'entrée),  $nc_p (nc_m)$  sont les nombres de Neurones cachés dans les réseaux MLP (ELM) respectivement. Les réseaux ont 1 neurone dans la couche de sortie.

- La validation se fait en comparant les sorties  $n$  éléments par  $n$ , correspondant aux sorties effectives des classificateurs à combiner.
- Durant l'étape de test, à l'image d'entrée, chaque classificateur a sa propre sortie  $Y_s$  qui est un vecteur de  $n$  éléments. La concaténation donne un vecteur de  $6 \times n$  éléments. Chaque colonne de six éléments est mise en entrée au classifieur-combinateur et est convertie par ce dernier en une sortie. Nous avons donc en sortie un vecteur de  $n$  éléments qui est le  $Y_r$  estimé.

Dans ce qui suit, nous détaillons les autres règles utilisées dans le processus de combinaison.

2. La règle du Maximum (MR) : Dans la combinaison des classificateurs à l'aide de la règle max, la sortie du classificateur résultant est un

vecteur de  $n$  éléments tels que :  $y_r = (y_{1r}, \dots, y_{Nr})$  où :

$$y_{ir} = \max(y_{is}), \quad s = 1, \dots, k ; \quad k = 6 \quad (4.3)$$

3. La règle de la moyenne (AR) : Dans la combinaison avec la règle moyenne, nous avons :

$$y_{ir} = \frac{1}{k} \sum_{s=1}^k y_{is}, \quad s = 1, \dots, k ; \quad k = 6 \quad (4.4)$$

Le vecteur résultant est la moyenne des vecteurs résultats de chaque classifieur.

4. Règle moyenne pondérée générale (GWAR) : Dans cette règle de combinaison, nous calculons d'abord le taux de reconnaissance global de chaque classifieur pendant l'étape d'apprentissage. Ces valeurs sont utilisées pour pondérer la moyenne des sorties individuelles :

$$y_{ir} = \frac{1}{k} \sum_{s=1}^k \mu_s \times y_{is}, \quad s = 1, \dots, k ; \quad k = 6 \quad (4.5)$$

où  $\mu_s$  est le taux de reconnaissance global atteint par le classifieur  $s$  au cours de l'étape d'apprentissage.

5. La combinaison avec la règle moyenne pondérée intra-classe (ICWAR) : Pendant l'étape d'apprentissage, nous calculons le taux de reconnaissance de chaque classifieur pour chaque classe et ensuite nous utilisons ces valeurs comme paramètres pour pondérer le calcul de la moyenne :

$$y_{ir} = \frac{1}{k} \sum_{s=1}^k \mu_{is} \times y_{is}, \quad s = 1, \dots, k ; \quad k = 6 \quad (4.6)$$

où  $\mu_{is}$  est le taux de reconnaissance atteint par le classifieur  $s$  pour la classe  $i$  pendant l'étape d'apprentissage.

6. La règle de combinaison max-moyenne (MAR) : Dans cette combinaison, nous séparons les résultats obtenus par les différents classifieurs comme suit :

- Dans un premier temps, nous calculons la moyenne des résultats obtenus par chaque classifieur (MLP, SVM ou ELM) en utilisant les caractéristiques de CM ou SCF :

$$y_{is} = \frac{1}{2}(y_{iCM} + y_{iSCF})_s \quad (4.7)$$

La moyenne est estimée au niveau 'caractéristiques'.

- Nous considérons alors la sortie maximale des trois classifieurs, à savoir ELM, SVM et MLP indépendamment des caractéristiques :

$$y_{ir} = \max(y_{is}), \quad s = 1, \dots, k ; k = 3 \quad (4.8)$$

Le maximum est calculé au niveau 'classifieurs'.

7. La règle de combinaison moyenne-max (ARM) : Dans la combinaison utilisant la règle ARM, nous séparons les résultats obtenus par les différents classifieurs comme suit :

- Tout d'abord, nous calculons le maximum des résultats obtenus par chaque classifieur (MLP, SVM et ELM) lors de l'utilisation des caractéristiques CM et SCF :

$$y_{is} = \max(y_{iCM}, y_{iSCF})_s \quad (4.9)$$

Le maximum est estimé au niveau 'caractéristiques'.

- Nous calculons alors la moyenne des trois classifieurs, à savoir ELM, SVM et MLP indépendamment des caractéristiques :

$$y_{ir} = \frac{1}{3} \sum_{s=1}^k (y_{is}), \quad s = 1, \dots, k ; k = 3 \quad (4.10)$$

La moyenne est calculée au niveau 'classifieurs'.

8. La règle Borda Count : Pour appliquer la Borda Count, la sortie de chaque classifieur est transformée en donnant des scores allant de 1 à  $n$  ( $n$  est le nombre de classes) à chaque classe sur la base de la valeur donnée par le classifieur. Maximum  $n$  est accordé à la classe la plus probable ayant le maximum  $y_{is}$ . Si plusieurs classes ont la même sortie par un classifieur, ces classes reçoivent le même score et à la classe qui leur succède est donné un score de  $n_i - c$  ( $n_i$  est le score

de la classe précédente et  $c$  est le nombre de classes avec la même sortie). Un exemple est donné comme : Si nous avons une *sortie* = [.3; .2; .9; .3; .1; .4; .8; .4; .3; .9], les scores correspondants seront les suivants : [5; 2; 10; 5; 1; 7; 8; 7; 5; 10]. (Dans l'exemple  $n = 10$ ). Les scores attribués à chaque classe sont additionnés pour tous les classifieurs. Ensuite, la classe ayant le score le plus élevé est élue par la règle de combinaison Borda Count adaptée.

9. La règle de combinaison Dempster-Shafer est une stratégie de combinaison évidentielle. Elle a été utilisée par [Kessentinietal2015] dans le domaine de la reconnaissance des mots manuscrits arabes. Nous avons utilisé dans ce travail, la règle de combinaison Dempster-Shafer basée sur les modèles décisionnels tels que formulés par Kuncheva et al. dans [Kunchevaetal2001]. Pour estimer les degrés de croyance d'une classe  $i$ , Kuncheva et al [Kunchevaetal2001] ont introduit la notion de modèles de décision ( $DT_i$ ) qui sont estimés dans l'étape d'apprentissage. Soit  $Z = z_1, \dots, z_N$ ,  $z_j \in R^N$ ,  $N$  le nombre d'éléments dans l'ensemble d'apprentissage. Un modèle de décision  $DT_i$  d'une classe  $i$  est une matrice  $k \times n$  définie par :

$$dt_i(p, q) = \frac{\sum_{j=1}^N ind(z_j, i) S_p^q(z_j)}{\sum_{j=1}^N ind(z_j, i)} \quad (4.11)$$

$p = 1, \dots, n; q = 1, \dots, k$ ; où  $ind(z_k, i)$  vaut 1 si  $z_k$  appartient à la classe  $i$  et 0 sinon.  $Dt_i(p, q)$  est la moyenne des résultats obtenus par le classifieur  $q$  à tous les éléments de la classe  $p$  pendant l'étape d'apprentissage. Le modèle de décision  $DT_i$  représente la référence à laquelle les sorties des classifieurs doivent se rapprocher pour élire la classe  $i$  pour l'entrée  $x$ . Le profil de décision d'une entrée  $x$ ,  $DP(x)$  est défini comme le résultat des  $k$  classifieurs à cette entrée. Cette entrée est attribuée par la règle de combinaison Dempster-Shafer à la classe qui maximise la similarité donnée par l'équation ?? entre le DP(x) et les  $DT_s$  :

$$\Phi_{i,j}(x) = \frac{(1 + ||DT_i^j - D_s(x)||^2)^{-1}}{\sum_{p=1}^n (1 + ||DT_p^j - D_j(x)||^2)^{-1}} \quad (4.12)$$

où  $||.||$  est n'importe quelle norme matricielle. En utilisant l'équation ??, nous calculons les degrés de croyance pour chaque classe  $i$  ( $i = 1,$

$\dots, n)$  et chaque classifieur  $s$  ( $s = 1, \dots, K$ ) comme :

$$b_i(D_s(x)) = \frac{\Phi_{i,s}(x) \prod_{k \neq s} (1 - \Phi_{k,s}(x))}{1 - \Phi_{i,s}(x) [1 - \prod_{k \neq s} (1 - \Phi_{k,s}(x))]} \quad (4.13)$$

Le vecteur résultant des degrés d'appartenance a les composantes :

$$\mu_{\hat{D}}^i(x) = C \prod_{s=1}^k b_i(D_s(x)), i = 1, \dots, n. \quad (4.14)$$

Où  $C$  est une constante de normalisation. Dans ce travail, nous avons utilisé comme norme matricielle la distance euclidienne et la valeur de  $C$  a été choisie empiriquement de 0,5.

10. La règle  $AR_{GWAR,Borda,DS}$  : Un deuxième niveau de combinaison est testé avec trois des règles discutées ci-dessus, à savoir le GWAR, la Borda Count et la règle Dempster-Shafer. Il consiste à estimer la règle AR sur ces règles. La moyenne est calculée en utilisant les résultats de ces règles de combinaison.

#### 4.4 Conclusion

Dans ce chapitre, nous avons résumé quelques unes de nos contributions concernant les domaines d'extraction de caractéristiques et la classification. Nous avons catégorisé les contributions en termes de comparaisons et de combinaisons. Les deux catégories appliquées aux deux domaines ouvre quatre voies d'exploration dont les résultats seront détaillés dans le chapitre ??.

Le prochain chapitre est consacré à la dernière voie de contribution que nous avons tentée. Celle-ci concerne le domaine de segmentation du manuscrit arabe.

## Contribution III : Segmentation

### 5.1 Introduction

Segmenter un manuscrit arabe est une tâche difficile parce qu'un caractère n'est pas seulement une entité qui a une forme qui est différente par rapport aux autres mais il peut être constitué d'un ensemble de formes hétérogènes. De plus, une lettre peut être similaire à une autre ne différant que par quelques constituants qui peuvent se résumer en un simple point ! Ces sous-parties peuvent être écrites loin du caractère lui-même ce qui donne naissance à une autre tâche à challenges qui consiste à affecter les parties secondaires aux parties principales correspondantes.

Le but de ce chapitre est de mettre en évidence notre contribution pour résoudre le problème de segmentation du manuscrit arabe en caractères. La méthode que nous proposons peut être subdivisée en étapes variant en difficultés mais dont les résultats de l'une sont subordonnés de ceux des autres.

- D'abord, nous segmentons le texte en lignes. On procède d'abord par une estimation de la rectitude de la page de texte pour éventuellement redresser cette dernière.
- En deuxième étape, nous délimitons les mots.
- Le mot est ensuite subdivisé en PAWs.
- Chaque PAW est segmentée en graphèmes. Ces graphèmes peuvent être constitués dans le meilleur des cas de caractères entiers. Très souvent par contre, ces graphèmes constituent des parties de caractères

et on parle alors de sur-segmentation. La sous-segmentation concerne le cas où une combinaison de caractères est le résultat de la segmentation.

- Pour résoudre le problème de sous-segmentation, on procède souvent par l'inclusion dans le lexique, de certaines des combinaisons usuelles du manuscrit arabe.
- La sur-segmentation nécessite quant à elle un traitement postérieur à la segmentation qui consiste à concaténer quelques parties segmentées dans le but de reconstituer le caractère entier. Ce traitement pose les grandes questions suivantes :
  - Quand est-ce qu'il faut procéder à ce traitement ou plus simplement comment savoir de manière automatique que l'on est dans le cas de sous-segmentation ?
  - Si on est et on sait que l'on est dans ce cas, comment savoir de manière automatique combien de parties faut-il concaténer ?

## 5.2 Description globale du travail effectué

Dans le script arabe, l'objet à segmenter c'est-à-dire le caractère est constitué d'un ensemble de formes hétérogènes qui constituent la partie principale représentant le corps de la lettre et quelques parties secondaires appelées diacritiques. Sous la nomination de 'diacritiques', nous incluons les points, les voyelles et autres formes telles que "chadda", "madda", et "hamza". Certaines de ces parties secondaires peuvent survenir, au-dessus ou en dessous de la partie principale du caractère. Il arrive même parfois qu'elles surviennent à l'intérieur du corps de lettre comme cela arrive pour les lettres "djim" ou "noun". La figure ?? résume les cas suscités.

Ces formes peuvent être très différentes les unes des autres et malgré cela, elles doivent être segmentées comme une seule entité ce qui rend la tâche très difficile.

Dans ce cadre, nous avons structuré le travail effectué comme suit :

1. Nous commençons par estimer la ligne de base de l'écriture par l'une des méthodes trouvées dans la littérature (se référer au chapitre ??

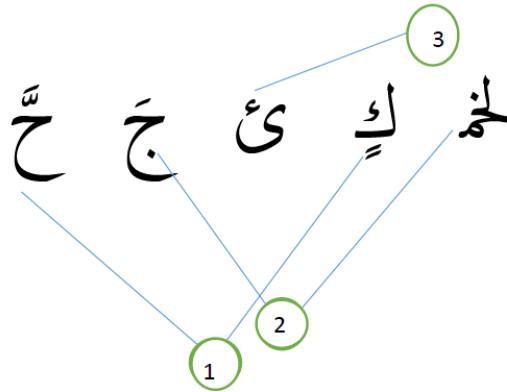


FIGURE 5.1 – Quelques uns des problèmes que l'on peut rencontrer avec la présence des diacritiques (1) Diacritiques au-dessus et en dessous avec différentes formes (2) Diacritiques (points) écrits à l'intérieur des corps de lettres (3) Une lettre avec son diacritique ayant globalement la même forme.

pour plus de détails).

2. Une segmentation en parties connexes est alors effectuée. Une partie connexe concerne toute partie ayant des pixels connectés selon un 4-voisinage ou 8-voisinage. Ceci peut concerner un corps de lettre ou un diacritique (point, voyelle, chadda, etc.).
3. Des critères basés sur les surfaces occupées par les parties connexes, permettent une première séparation entre parties principales et parties secondaires.
4. Une affectation des parties secondaires aux parties principales correspondantes est réalisée moyennant quelques heuristiques.
5. Chaque partie principale avec les parties secondaires respectives est candidate à la segmentation en graphèmes. Quelques heuristiques permettent de ne pas segmenter les caractères isolés. Toutes les autres parties sont mises en entrée du module de segmentation après les avoir séparé de leurs diacritiques pour simplifier la tâche de segmentation.
6. Comme certains caractères, comme **س**, **ص** ou même **ش**, peuvent être sur segmentés, certaines règles basées sur des notions de proximité, de chevauchement horizontal et des règles de l'écriture du script arabe,

sont utilisées pour assurer une concaténation des parties segmentées en vue d'obtenir des caractères entiers.

7. Une validation automatique de ces traitements a été un peu difficile pour des raisons parmi lesquelles, nous citons :
  - (a) Au début, nous n'avions pas à notre disposition une base de données contenant des images de caractères pré segmentés avec les différentes formes de ces derniers suivant leur éventuelle position dans la PAW.
  - (b) Ceci nous a obligée à réaliser cette base. Une description de cette base est donnée très succinctement dans la section ?? du chapitre ??.
  - (c) Même une fois créée, la base a posé quelques problèmes que nous détaillerons comme suit :
    - Les formulaires utilisés pour récolter les données ont forcé sans aucune intervention de notre part les scripteurs à écrire de façon contrainte.
    - Utiliser des caractères pré segmentés écrits de façon con-trainte pour valider une segmentation sur une écriture très libre comme celle de la base IFN/ENIT a posé des problèmes d'une autre source, qui s'est vue superposée aux problèmes inhérents à ceux de la segmentation elle-même.
    - Ceci a été déduit après analyse des parties segmentées qui même lorsqu'elles représentaient des caractères entiers et bien segmen-tés ont été mal reconnus par des classifieurs qui ont donné de très bons taux de reconnaissance sur la base TBK.
    - Nous nous sommes rendue compte récemment que les chercheurs ont toujours utilisé pour la validation de leurs systèmes des ca-ractères segmentés manuellement sur la même base utilisée pour la segmentation. Nous n'avons pas encore testé cette façon de faire.
8. Les validations à critères visuels concernant les segmentations en PAWs et en caractères sont effectuées et nous ont servi à compa-rer notre approche avec celle décrite dans le travail le plus proche du

nôtre dans le temps et que nous avons décrit dans la section ?? du chapitre ??.

Nous avons donc proposé deux algorithmes de segmentation, l'un validé sur une écriture contrainte sensée respecter le maximum de règles d'écriture. L'autre est validé sur la base IFN/ENIT, base connue pour être constituée d'une écriture très libre assurée par plus de 411 scribes.

### 5.3 Segmentation de l'écriture contrainte

Dans ce qui suit, nous allons détailler notre premier système basé segmentation. Nous commençons par décrire son architecture générale et puis nous allons détailler les deux sous-systèmes à savoir :

- Segmentation sans squelettisation ;
- Segmentation après squelettisation.

Comme ces deux systèmes adressent une écriture contrainte, c'est-à-dire respectant au maximum les règles d'écriture soignée, ces deux types de segmentation sont possibles, contrairement aux systèmes non-contraints qui eux sont obligés de compter sur une 'bonne' squelettisation pour assurer une segmentation correcte.

#### 5.3.1 Architecture du système

Comme le montre la figure ??, dans ce système, deux étapes majeures sont prises en charge :

- L'apprentissage des caractères par les classifieurs de Kohonen et MLP.
- L'algorithme SFVA (Special Features Vectors Addition) que nous proposons et sa validation en utilisant des images de textes manuscrits arabes.

#### 5.3.2 Apprentissage

L'étape d'apprentissage est assurée par :

- La délimitation des caractères dans les images de caractères pré-segmentés et leur segmentation.

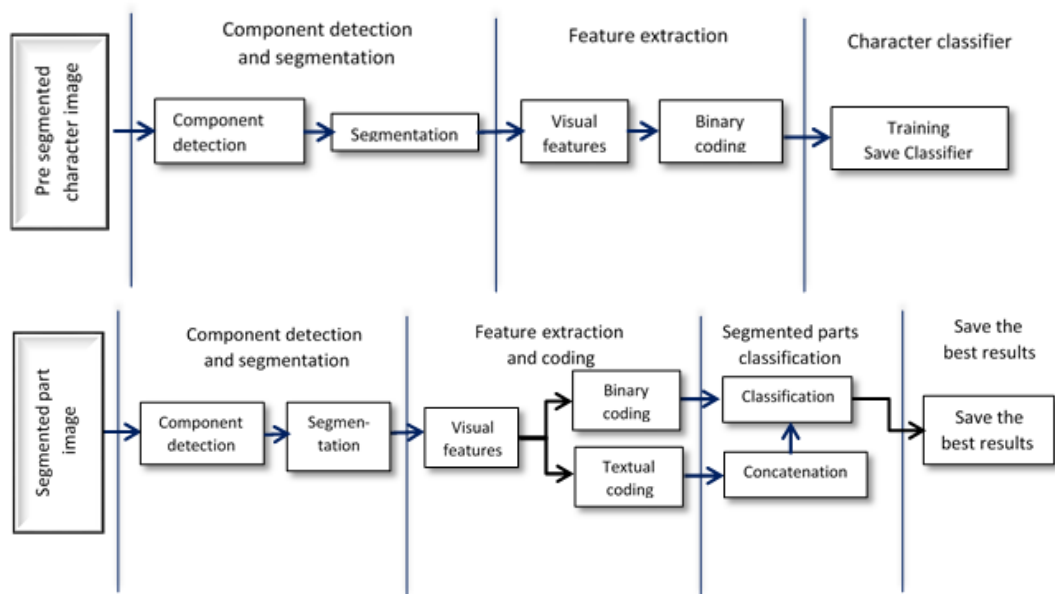


FIGURE 5.2 – Diagramme bloc du système de reconnaissance (a) Étape d'apprentissage (b) Étape de Concaténation/Validation

- Extraction des caractéristiques.
- Apprentissage et Classification des caractères obtenus.

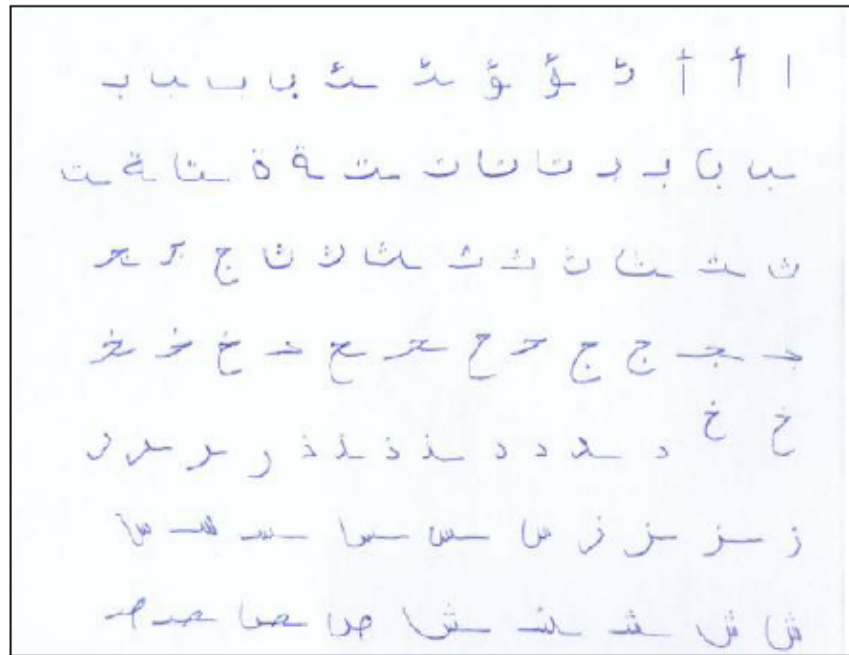
### Délimitation des caractères et leur segmentation

Les entrées de ce module sont des images de caractères pré-segmentés. Un exemple est montré dans la figure ?? (a). Une série de pré-traitements est appliquée sur les différentes images pour éliminer le bruit, et binariser les images obtenues (figure ?? (b)).

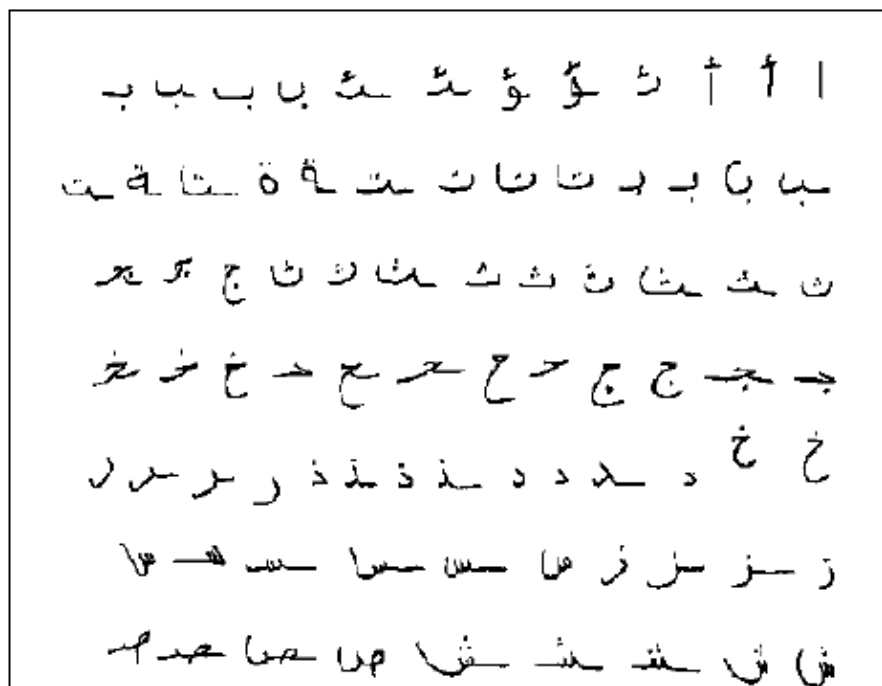
La figure ?? montre les images de quelques caractères extraits de l'image de la figure ?? (b).

### Extraction des caractéristiques : Caractéristiques du type visuel

Pour procéder à l'extraction des caractéristiques, les images des parties segmentées sont d'abord squelettisées par l'algorithme de Zhang & Suen amélioré (voir paragraphe ??). Remarquer que pour cette étape, la squelettisation s'effectue après la segmentation indépendamment du fait du fait que cette dernière soit faite avec ou sans squelettisation. Les



(a)



(b)

FIGURE 5.3 – Image de caractères pré-segmentés (a) Avant filtrage (b) Après filtrage et binarisation

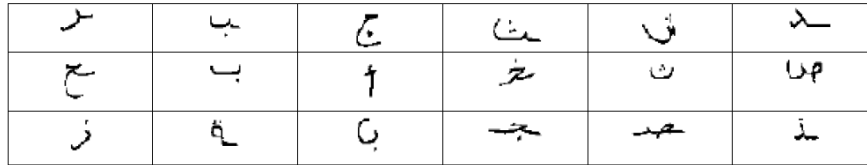


FIGURE 5.4 – Images de quelques caractères segmentés de l’image de la figure ?? (b)

caractéristiques que nous avons choisies pour décrire nos images de caractères sont celles qui, à notre sens ressemblent le plus à celles que les êtres humains utilisent dans les étapes d’apprentissage et de reconnaissance de l’écriture. Elles sont du type visuel et englobent :

- La forme du caractère (carrée, allongée, etc.)
- La présence de boucles et leur localisation.
- La présence ou l’absence de diacritiques et leurs éventuelles positions.

Aussi, pour simuler le comportement humain lors du processus de concaténation, nous avons codé les primitives suscitées de deux manières différentes :

- Un codage binaire nécessaire pour les classifieurs utilisés.
- Un codage textuel qui nous a permis de faciliter l’utilisation de règles linguistiques de la langue arabe dans le processus de concaténation.

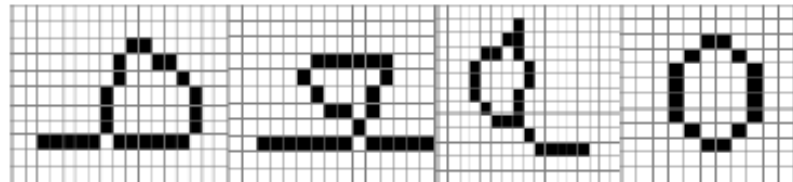
La structure utilisée pour coder le codage pseudo-textuel des primitives utilisées a la forme suivante :

```

struct Caract_enreg
{
    int nbp, HMZ, Nb_pex, Nb_pbr, Nb_pcr, Bcl, Rpl_hg, Rpl_bg,
    Rpl_hd, Rpl_bd;
    Pos_p char[20];
    Pos_HMZ char[20];
    Pos_bcl char[20];
    Frm_car char[20].
}

```

Nb\_pex représente le nombre de points extrêmes et pos\_bcl représente la position de la boucle. La figure ?? représente les différentes caractéristiques prises en compte dans ce système.



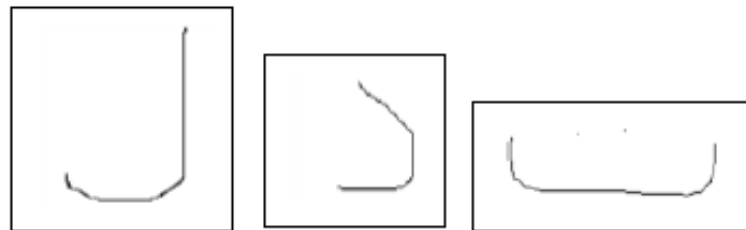
Loops and their positions

Extreme Points      Connection Point



Crossing Point

Characteristic points



upright

square

long

Characters forms

FIGURE 5.5 – Images de quelques caractères segmentés de l'image de la figure ?? (b)

### 5.3.3 Description de la méthode de segmentation

Elle est subdivisée en trois étapes :

- La segmentation horizontale.
- La segmentation verticale.
- La segmentation en composantes connexes.
- La segmentation en caractère sans squelettisation ou avec squelettisation.

**La segmentation horizontale** Elle a pour but de localiser les lignes de texte en projetant horizontalement la page de texte sur un axe vertical. L'histogramme ainsi obtenu présente des zones non nulles qui représentent les lignes de texte et des zones nulles qui représentent les interlignes. Le début de la ligne correspond à la première ligne de la matrice binaire qui possède au minimum un pt noir. La fin de la ligne de texte correspond à la ligne de la matrice binaire qui ne possède aucun point noir (voir figure ??).

#### Algorithme

##### Début

Tant que non fin ligne de l'image

    Faire

        Tant que la ligne de l'image ne contient aucun point noir

            Faire

                Saut vers la ligne suivante ;

            Fait ;

        Sauvegarder le début de la ligne de l'image ;

        Tant que la ligne de l'image contient un point noir au minimum

            Faire

                Saut vers la ligne suivante ;

            Fait ;

        Sauvegarder la fin de la ligne de texte ;

    Fait ;

**Fin.**

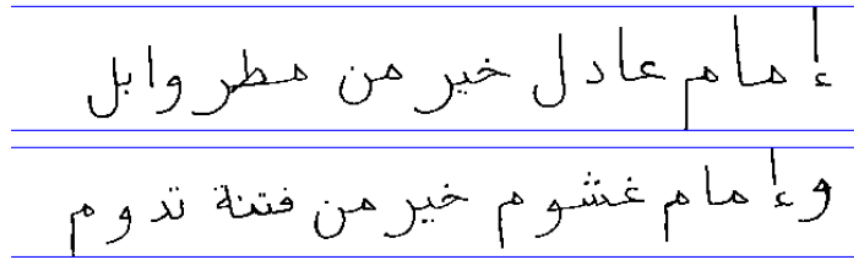


FIGURE 5.6 – Segmentation en lignes

**La segmentation verticale** Cette méthode a pour but la séparation d'une ligne de texte en parties connexes en parcourant l'histogramme vertical de la ligne colonne par colonne. Le début de la partie connexe correspond à la première colonne de la matrice binaire qui contient au minimum un point noir, la fin de la partie connexe correspond à la première colonne de la matrice binaire qui ne contient aucun point noir (voir figure ??).

### Algorithme

#### Début

Pour chaque ligne de l'image

    Tant que la colonne de l'image ne contient aucun point noir

    Faire

        Saut vers la colonne suivante;

    Fait;

    Sauvegarder la colonne début de la partie connexe;

    Tant que la colonne de l'image contient un point noir au minimum

    Faire

        Saut vers la colonne suivante;

    Fait;

    Sauvegarder la colonne fin de la partie connexe;

Fait;

#### Fin



FIGURE 5.7 – Segmentation verticale après une segmentation horizontale

**Étiquetage à l'aide d'une table d'équivalence « l'algorithme de Rosenfeld » : Segmentation en parties connexes** Cet algorithme ne nécessite que deux balayages de l'image, mais requiert l'utilisation d'une table d'équivalence. C'est la solution la plus couramment employée de par sa simplicité de mise en œuvre.

Lors d'un premier balayage de l'image, une nouvelle étiquette est assignée à chaque point rencontré non encore étiqueté et sans voisin étiqueté. Si le voisinage du point courant comporte un ou plusieurs points déjà étiquetés, la valeur minimum de ces étiquettes est choisie et les conflits sont stockés dans la table d'équivalence. Les équivalences sont résolues à la fin du premier balayage et un second balayage attribue à chaque pixel son étiquette définitive.

En bref, cet algorithme est exécuté en deux phases :

**La première phase :** Elle consiste en un balayage de l'image ligne par ligne pendant lequel on affecte une étiquette temporaire à chaque pixel.

**La deuxième phase :** Elle comporte la résolution des équivalences et un deuxième balayage de l'image pour l'attribution des étiquettes définitives aux pixels.

Avant la description le processus d'étiquetage, citons quelques définitions :

**Équivalence :** Une étiquette 'a' est dite équivalente à l'étiquette 'b'

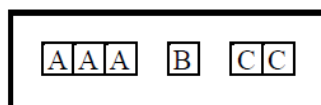


FIGURE 5.8 – Propagation de l'étiquetage des pixels de gauche à droite sur une ligne horizontale

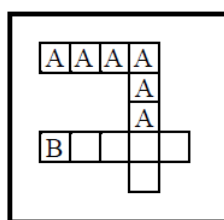


FIGURE 5.9 – Propagation de l'étiquetage des pixels de haut en bas sur une colonne verticale

s'il existe un chemin reliant le point correspondant à 'a' au point correspondant à 'b', tel que ce point ne passe pas par 0.

**Table d'équivalence :** C'est un tableau dynamique dont la taille augmente pendant le parcours séquentiel de l'image selon le besoin (lors de l'apparition de nouvelles étiquettes et lors de l'insertion des étiquettes équivalentes à d'autres étiquettes).

**Le processus d'étiquetage :** En parcourant une ligne horizontalement de la gauche vers la droite, on associe un numéro (une étiquette) à chaque pixel de telle sorte que tous les pixels voisins portent le même numéro (le numéro zéro est réservé au pixel "vide"). Lorsque sur cette ligne, le voisinage est interrompu, puis reprend plus loin, le numéro est incrémenté de 1. Les étiquettes sont représentées par les lettres A, B et C sur la figure ??.

Lorsqu'une nouvelle ligne est commencée, on propage naturellement l'étiquetage de haut en bas en recopiant le numéro du pixel qui se trouve au-dessus du premier pixel de la nouvelle ligne. S'il n'y a pas de pixel au-dessus, un nouveau numéro est utilisé (figure ??).

Lorsqu'un conflit se présente entre la propagation horizontale et la

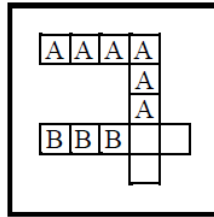


FIGURE 5.10 – Conflit entre la propagation horizontale et verticale

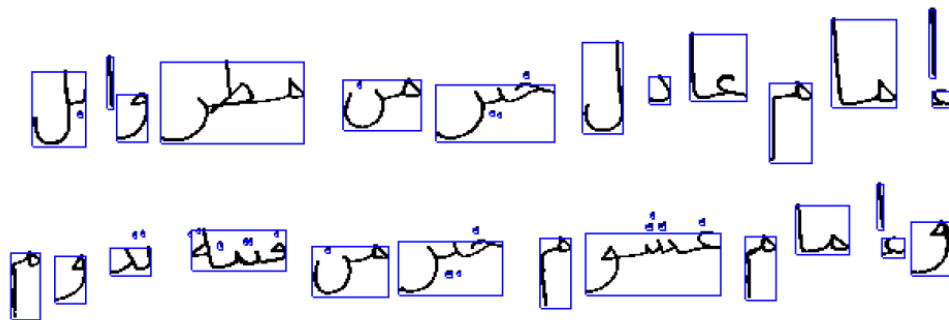


FIGURE 5.11 – Étiquetage (segmentation) en composantes connexes

propagation verticale des étiquettes, deux cas se présentent alors (figure ??).

La figure ?? montre un exemple du résultat de l'étiquetage d'un texte en composantes connexes.

**La segmentation en caractère sans squelettisation :** Le but de cette étape est l'extraction des caractères de chaque partie connexe obtenue dans la segmentation verticale.

Nous définissons le seuil de segmentation comme la valeur la plus répétitive parmi les valeurs de l'histogramme vertical, il varie d'une partie connexe à une autre.

Pour extraire un caractère, il faut déterminer son début et sa fin. Quelques conditions sur la colonne qui correspond au début d'un caractère sont :

لا تحير فيا ود امرئ متلوتا اذا الريح مالت  
مالا حيث تميل وما أكثر الخوات حير  
تعدهم ولكنهم في الثابتات قليل  
الحفظ لسانك أيها إلى لسان لا يلد عندك انه  
نعسان كم في المقابر من قليل لسانه كانت  
تهاب لقاءه الأقران  
ولرب نازلة يفتق لها الفتى ذرعا وعند الله  
منها المخرج ضاقت فلما استحكمت حلقاتها  
فرجت وكانت أظنما لا تفرج

FIGURE 5.12 – Segmentation sans squelettisation

1. L'histogramme vertical de la colonne testée  $i$  doit être inférieur ou égal au seuil.
2. Le nombre de transitions de la colonne testée  $i$  est inférieur ou égale à 2.
3. L'histogramme vertical de la colonne  $i-1$  et  $i-2$  doit être supérieur au seuil.
4. Le nombre de transitions de la colonne  $i+1$  et  $i+2$  est inférieur ou égal à 2.

Un exemple est montré en figure ??.

**Segmentation avec squelettisation :** Un exemple est donné dans la figure ?? . Cet exemple montre quelques points de segmentation pro-

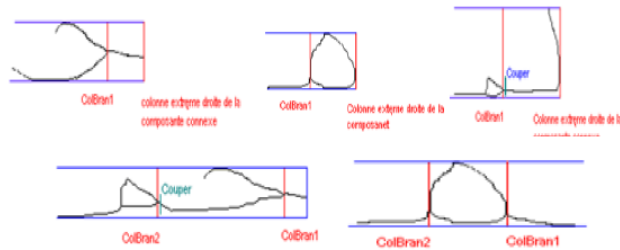


FIGURE 5.13 – Quelques points de segmentation proposés par notre deuxième algorithm

posés par notre algorithm. La figure ?? montre quelques résultats de squelettisation et segmentation.

### 5.3.4 La concaténation

Nous pouvons remarquer que, souvent, les parties segmentées n'englobent pas la totalité des formes des caractères. En effet, les parties encerclées de bleu dans la figure ?? montrent quelques cas de sur-segmentation rencontrés par notre algorithm.

Quand on est dans un état de sur-segmentation, la solution évidente serait de "coller" les parties correspondant au même caractère. Mais :

- Comment savoir qu'on est réellement dans cet état ?
- Comment coller les parties sur segmentées ?
- Combien de parties coller pour être sûr d'avoir reconstitué la totalité du caractère ?

L'algorithm que nous proposons pour la concaténation des parties résultant de la segmentation de l'écriture contrainte est le suivant :

#### Début

Pour tous les vecteurs des parties segmentées de l'image du texte

Faire

cpt=1

Introduire le vecteur en entrée du réseau

Enregistrer la sortie

Tant que (cond-past &&  $cpt \leq nb_{max}$ )

Faire

من وضعه ذنبه رفعه صلبه

من وضعه ذنبه رفعه صلبه

(a)

احفظ لسانك أيتها الإنسان يا بلاد عنك انه  
ثعبان كم في التابير من قبيل السانه كانت  
تصاب لقاءه الا قران

ولرب نازلة يفتيق لها القنى ذرعا وعند الله  
منها المخرج ضائفت فلما التحكمت حلقاتها  
فارجعت وكننت اطلما لا يفرج

(b)

FIGURE 5.14 – Squelettisation (a) et segmentation (b). Les parties encerclées avec du bleu montrent les cas de sur segmentation.

Concaténer

Introduire le vecteur résultat en entrée du réseau

Enregistrer la sortie

$cpt=cpt+1$

Fait

Sauvegarder la concaténation qui a donné les meilleurs résultats

Sauvegarder la sortie correspondante

Localiser le prochain vecteur

Fait

**Fin**

$nb_{max}$  représente le nombre maximal de concaténations admises par le système. Il a été estimé de manière empirique à 5.

Les conditions considérées dans "cond-past" sont relatives aux caractéristiques de l'écriture arabe et permettent de formaliser les conditions de concaténation. A titre d'exemple, nous pouvons citer :

1. La partie segmentée ne doit pas être considérée isolée. Une partie est considérée comme isolée si elle respecte l'une ou les conditions suivantes :
  - Le ratio largeur sur hauteur de la boîte englobant cette partie est inférieur à un seuil. Cette condition permet de ne pas essayer de segmenter des caractères comme "alif", "dal" ou "ra" quand ils ont isolés.
  - Si ce ratio n'est pas inférieur au seuil, l'histogramme vertical doit vérifier la condition de pas avoir plus de deux pics supérieurs à un seuil. C'est pour détecter les caractères "bba" ou "ta" quand ils sont isolés.
2. La partie segmentée ne doit pas être localisée à la fin de la PAW mère. Les informations sur la position de la partie segmentée sont automatiquement sauvegardées lors du processus de segmentation.
3. Si la partie courante contient une boucle, la partie suivante ne doit pas contenir de boucle aussi.
4. Une partie avec des diacritiques au-dessus ne doit pas être concaténée à une partie ayant des diacritiques en-dessous etc.

### 5.3.5 Addition Spéciale des vecteurs caractéristiques des parties à concaténer (SFVA)

Quand toutes les conditions sont satisfaites, nous réalisons la concaténation en effectuant "l'addition spéciale" des deux vecteurs caractéristiques des parties à concaténer. Quelques exemples des règles utilisées pour calculer le vecteur résultat de la SFVA sont :

1. Le nombre de points caractéristiques dans le vecteur résultant est égal à  $n1 + n2 - 2$  où  $n1$  et  $n2$  sont les nombres des points caractéristiques dans les deux vecteurs à additionner.
2. Si le premier vecteur appartient à une partie au début de la PAW et le deuxième à une partie en fin de PAW, le vecteur résultant sera celui d'un caractère isolé.
3. Si la première partie est au milieu et la seconde à la fin, le résultat est une partie de fin de PAW.
4. Si la première partie est au début avec une deuxième partie au milieu, le résultat est au début de la PAW.
5. Si les deux parties sont au milieu de la PAW, le résultat est aussi au milieu de la PAW, etc.

Le vecteur résultat est codé pour exprimer toutes les règles de concaténation.

### 5.3.6 Validation

La procédure que nous avons suivie pour valider notre système est la suivante :

1. Nous avons d'abord entraîné deux types de réseaux de neurones pour apprendre les caractères pré segmentés. Ces deux réseaux sont le réseau de Kohonen et le réseau de neurones multicouche.
2. Nous avons entré les vecteurs des différentes parties segmentées dans les réseaux et nous avons récupéré la classe prédite avec la valeur réelle de la probabilité d'appartenance.

3. Nous avons appliqué la SFVA à chaque fois que les règles d'applicabilité sont vérifiées et que le nombre maximum de concaténations n'était pas dépassé.
4. Nous avons entré les vecteurs résultats des concaténations dans les réseaux et nous avons dans tous les cas sauvegardé la configuration qui a donné la meilleure probabilité.
5. Nous avons donc reproduit les caractères constituant les PAWs en termes de vecteurs caractéristiques représentés par les concaténations sélectionnées.
6. Le calcul du taux de bonne reconnaissance a été à ce moment-là calculé de façon visuelle.

#### 5.4 Segmentation sur une écriture non contrainte

Pour traiter l'écriture non contrainte, la procédure suivie comprend les grandes étapes suivantes :

1. La segmentation en PAWs : Dans cette étape, il s'agit de délimiter les parties principales des mots ou sous-mots et leur affectation des diacritiques correspondants.
2. La segmentation en graphèmes : Là, les différentes PAW sont segmentées en graphèmes avec affectation aux graphèmes des diacritiques correspondants.
3. L'étape de concaténation : La concaténation est faite au niveau image. Le résultat est une image où des parties segmentées sont collées.
4. Les vecteurs caractéristiques sont à chaque fois extraits des parties segmentées ou des parties obtenues après concaténation de parties segmentées.

Tout ce ci est expliqué dans ce qui suit.

##### 5.4.1 Segmentation en PAWs

Un exemple de résultat escompté de ce traitement est montré dans la figure ???. Pour arriver à ce résultat, plusieurs étapes sont nécessaires.



FIGURE 5.15 – Exemples de résultats escomptés de l'étape de segmentation en PAW.



FIGURE 5.16 – Quelques étapes de la segmentation en PAW. À gauche : estimation et À droite : redressement de la ligne de base.

La première étape consiste à estimer la ligne de base d'écriture. Une rotation de l'image est appliquée sur l'image si nécessaire pour avoir une ligne d'écriture horizontale. La méthode utilisée est la méthode de projection d'histogramme. Nous allons suivre une image qui subit toutes les étapes jusqu'à la finalisation de la segmentation en PAW. La figure ?? montre le résultat de l'étape d'estimation de la ligne de base, et de son redressement.

L'algorithme proposé pour décider laquelle des différentes composantes connexes est une partie principale et lesquelles forment les diacritiques est le suivant :

**Algorithme : Séparation des diacritiques et des parties principales**

For each CC do

If  $(CC.area > min_{area1}) \&\& (CC.width < min_{width})$   
 $image.nb_{mb} ++$ ; //  $nb_{mb}$ =number of main bodies  
 $image.main_{body}(nb_{mb}) = CC$ ;

elseif

$(CC.area > min_{area2}) \&\& (dist(CC, lb) < dist_{min})$  // lb=baseline  
 $image.nb_{mb} ++$ ;  
 $image.main_{body}(nb_{mb}) = CC$ ;

else

$image.nb_{dc} ++$ ; //  $nb_{dc}$  =number of diacritics  
 $image(nb_{dc}) = CC$ ;



FIGURE 5.17 – Segmentation en parties connexes, séparations parties principales et diacritiques.

End.

La figure ?? montre les étapes de segmentation de l’image en parties connexes suivie de l’estimation des parties principales et des diacritiques.

- La première condition permet de sélectionner la caractère ا (alif) comme partie principale.
- La deuxième condition donne la main pour déterminer les parties principales selon des conditions sur les surfaces et la distance de la CC par rapport à la ligne de base.
- La distance est estimée entre le centre de gravité de la CC et la ligne de base.
- La largeur de la CC est la largeur de la boîte englobante correspondante.
- $min_{area1}$ ,  $min_{area2}$  et  $min_{width}$  sont estimées empiriquement.

Après avoir exécuté cet algorithme sur une image, nous pouvons séparer les parties principales et les diacritiques. Il est maintenant le moment de décider quels diacritiques doivent être affectés à quelles parties principales. C’est l’affaire de l’algorithme suivant :

**Algorithme : Affectation des diacritiques aux MB**

For each DC do

    Compute the horizontal overlap (HO) of the bounding box of the DC and all the MB ones ;

    Compute the distances (DB) between the bounding box of the DC and the MB ones ;

    If HO has values not equal to zero

        The DC is assigned to the MB that maximizes HO

    Else

        Assign the DC to the MB that minimizes DB

End

Le résultat de cet algorithme quand il est appliqué à l'image suscitée est montré dans la figure ??.

The image shows the handwritten Arabic phrase 'جبل الزمان' (Jabal al-Zaman). The characters are colored: 'ج' is blue, 'ب' is green, 'ل' is orange, 'ا' is red, 'ز' is blue, 'م' is green, and 'ان' is orange. This represents the final step of diacritic assignment in the PAW segmentation process.

FIGURE 5.18 – Dernière étape de la segmentation en PAW. Affectation des diacritiques aux parties principales correspondantes.

La figure ?? montre comment cet algorithme résout quelques problèmes d'affectation de quelques diacritiques aux parties principales correspondantes tandis que la figure ?? montre quelques cas non résolus par l'algorithme. Dans ce cas, d'autres connaissances linguistiques sont nécessaires pour résoudre les problèmes cités.

La figure ?? montre les résultats de cet algorithme quand il est appliqué à un texte de la base AHDB. Nous remarquons que dès que notre algorithme est appliqué est à une écriture contrainte, il offre de bons résultats.

#### 5.4.2 Segmentation en graphèmes

Dans le but de segmenter des mots en graphèmes qui sont des entités qui peuvent représenter un caractère ou plus ou moins qu'un caractère, nous procédons comme suit :

1. D'abord, nous soustrayons temporairement les diacritiques de chaque partie principale. Ces derniers sont appelés à être réaffectés aux graphèmes correspondants après segmentation.
2. Pour éviter de segmenter des entités qui représentent des caractères

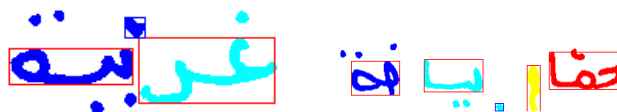
The image shows two examples of handwritten Arabic text. The first is 'غربة' (Gharba) with colored diacritics and a red box around the 'ر' and 'ب' characters. The second is 'صفا' (Safa) with colored diacritics and a red box around the 'ف' and 'ا' characters. This illustrates the resolution of diacritic assignment problems.

FIGURE 5.19 – Résolution de quelques problèmes d'affectation des diacritiques aux parties principales correspondantes.

الخليج السد الحوفي بلا ريجا

FIGURE 5.20 – Quelques problèmes non résolus par l’algorithme. De gauche à droite, un diacritique assimilé à une partie principale parce qu’il a été écrit trop grand ; deux PAW ont été assimilées à une seule parce qu’elles ont été écrites collées et enfin, deux PAW collées ont été considérées comme une seule et deux parties d’une PAW ont été prises pour des PAW différentes parce que écrites très séparées les unes des autres.

يهدف البحث إلى دراسة الخواص الحرارية  
والضوئية والميكانيكية لمادة البوليمر والبحث  
عن تغير خواصها بفعل العوامل المؤثرة لكي  
نعرف مدى استجابتها للمؤثرات الخارجية  
بالتشجيع والتعرض للجو الخارجي وتأثير ماء  
البحر. وذلك من أجل تحسين الاداء العملي  
تستخدم هذه المادة في الصناعات كالعوازل  
الكهربية ومجالات التغليف والطب وغيره  
حيث تعود هذه الأهمية إلى سهولة التصنيع  
وانخفاض التكلفة وسهولة التشكيل

FIGURE 5.21 – Un texte de la base AHDB segmenté en PAW par notre algorithme.



FIGURE 5.22 – Un mot squelettisé avec l’algorithme de Zhang & Suen suivi du post-traitement.

isolés, un ratio entre la hauteur et la largeur est calculé. Si ce ratio est inférieur à un seuil défini empiriquement :

- Si dans cette partie, aucun point caractéristique (croisement ou branchement) n’existe alors la partie n’est pas à segmenter.
- S’il existe au moins un point caractéristique, la partie principale (MB) correspondante est considérée comme candidate à la segmentation en graphèmes.

3. Parce que les ligatures entre les caractères dans le script arabe sont généralement toutes réalisées le long de la ligne de base d’écriture nous excluons tout point de segmentation loin de cette ligne vers le bas ou vers le haut. Pour ce faire, chaque partie sujette à être segmentée se voit estimer sa ligne de base. Pour accepter un point candidat à la segmentation, il doit avoir une distance inférieure à un seuil empirique par rapport à la ligne de base.
4. La segmentation en graphèmes de la partie MB est effectuée après squelettisation de cette dernière à l’aide de l’algorithme de Zhang & Suen suivie du post-traitement proposé dans la section ?? du chapitre ?. La figure ?? montre un exemple de mot après squelettisation et post-traitement.
5. Les points de segmentation sont choisis à quelques pixels de la fin de sections avec un pixel d’épaisseur (voir figure ??).

La condition ?? est là pour éviter de segmenter des caractères isolés avec un petit ratio comme les lettres ب, ت ou ث. Dans la condition ??, nous n’oublions pas que souvent, les scribes ont tendance à effectuer des ligatures verticales. Ce problème doit être résolu autrement que par la segmentation mais par ajout dans la base des caractères à apprendre certaines ligatures impossibles à segmenter.



FIGURE 5.23 – Segmentation d’un mot en graphèmes avec filtrage des points potentiels de segmentation. La suppression du deuxième point de segmentation de la première partie principale (MB) à partir de la droite est assurée par la condition ?? suscitée.

### Affectation des diacritiques aux graphèmes correspondants

Comme expliqué plus haut, les points de segmentation sont choisis proches de la ligne de base d’écriture vers la fin de sections avec un pixel d’épaisseur dans le squelette amélioré de Zhang & Suen de l’image du mot. Après segmentation de la MB, nous procédons comme suit :

1. Nous retournons à l’image avant squelettisation.
2. Nous segmentons au niveau des points de segmentation finaux.
3. Nous affectons chaque diacritique au plus proche graphème. Remarquez qu’ici, le même algorithme que celui décrit en ?? est utilisé.
4. Si le dernier point de segmentation d’une MB présente la particularité de ne pas être suivi par une forme particulière, ou qu’il est à une distance inférieure à un seuil du point de fin de PAW, il est supprimé des points de segmentation finaux.

La figure ?? montre tout le processus quand il est appliqué à une image de mot. La figure ?? montre le résultat sur l’image d’une ligne de texte.

## 5.5 Conclusion

Dans ce chapitre, il s’est agi de faire un tour sur notre contribution concernant le module de segmentation dans un système de reconnaissance du manuscrit arabe. Ce module, dû à son importance quant à tout autre traitement postérieur nous a pris et nous prend toujours beaucoup de temps et d’effort. En effet, après un long travail sur le manuscrit contraint où nous avons mis au point un algorithme de segmentation en conséquence, un deuxième a concerné le manuscrit non contraint pour

تونست العباضة الاصلية  
تونست العباضة الاصلية  
تونست العباضة الاصلية  
تونس عباضة الاصلية  
تونس عباضة الاصلية  
تونس عباضة الاصلية  
تونس عباضة الاصلية

FIGURE 5.24 – Un exemple de nom de ville segmenté en graphèmes. Les parties non présentes dans la figure sont celles qui ne sont pas passées par le module de segmentation.

يهدف البحث إلى دراسة الخواص الحرارية  
يهدف لبحث سة لحوص لحرارية

FIGURE 5.25 – La segmentation d'une ligne de texte en graphèmes. Les parties non présentes dans la figure sont celles qui ne sont pas passées par le module de segmentation.

répondre au besoin de travailler sur le manuscrit dans toutes ses formes. Actuellement, nous travaillons sur la possibilité de normaliser l'écriture manuscrite arabe afin de pallier encore plus à toutes les difficultés relatives aux traitements du manuscrit quelle que soit sa source et sa forme. Le chapitre qui suit sera consacré aux principaux résultats obtenus dans tout ce qui a concerné notre travail et contribution dans le domaine.

## Partie III : Expérimentations et Résultats

Cette partie est consacrée à présenter les résultats obtenus dans les trois modules :

- Prétraitements qui inclue :
  - La squelettisation.
  - La détection et redressement de la ligne de base.
- Extraction de caractéristiques.
- Classification.
- Segmentation.

# Chapitre 6

## Résultats expérimentaux

### 6.1 Introduction

Il est nécessaire de mentionner que ce chapitre sera dédié uniquement aux principaux résultats obtenus par nos soins quant aux contributions que nous avons mentionnées dans les chapitres 3, 4 et 5. Nous allons aussi respecté l'ordre pris dans le document, qui, lui-même respecte l'ordre de la succession de traitements par lesquels passe toute image contenant du manuscrit jusqu'à la phase finale de reconnaissance.

### 6.2 Pré-traitements

#### 6.2.1 Squelettisation

Comme nous l'avons mentionné dans le chapitre dédié à la problématique de squelettisation, l'évaluation des traitements a été faite à l'aide de critères visuels où nous ne pouvons pas éliminer le caractère subjectif de l'estimation de bon ou mauvais. Nous l'avons donc agrémentée d'une évaluation à caractère objectif en utilisant des critères quantitatifs calculables.

#### Critères visuels

Les figures ?? à ?? montre quelques résultats comparatifs entre les différents algorithmes appliqués à quelques images de mots de la base IFN/ENIT. Les critères adoptés ici sont du type visuel.

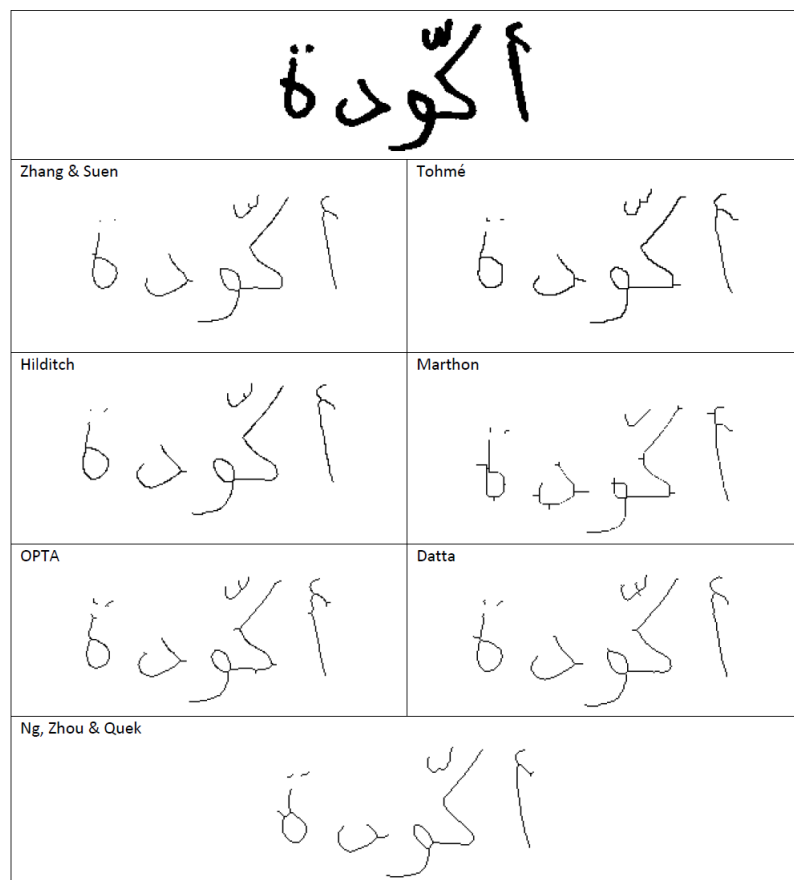


FIGURE 6.1 – Exemple 1 de comparaison visuelle entre les différents algorithmes de squelettisation.

Ce qui peut être déduit visuellement de ces figures et autres non montrées ici, est que c'est l'algorithme de Zhang & Suen qui a le mieux squelettisé les mots de la base IFN/ENIT.

1. Son squelette a présenté le minimum de barbules.
2. Il a le mieux respecté la forme initiale du mot.
3. Il a le mieux respecté les connectivités des composantes de la forme initiale.

Il est directement suivi par l'algorithme de Hilditch. En fait, les exemples ?? et ?? montrent clairement la supériorité de l'algorithme de Zhang & Suen. Ces conclusions ont été confrontées aux critères calculables et les résultats sont montrés dans le tableau ??.

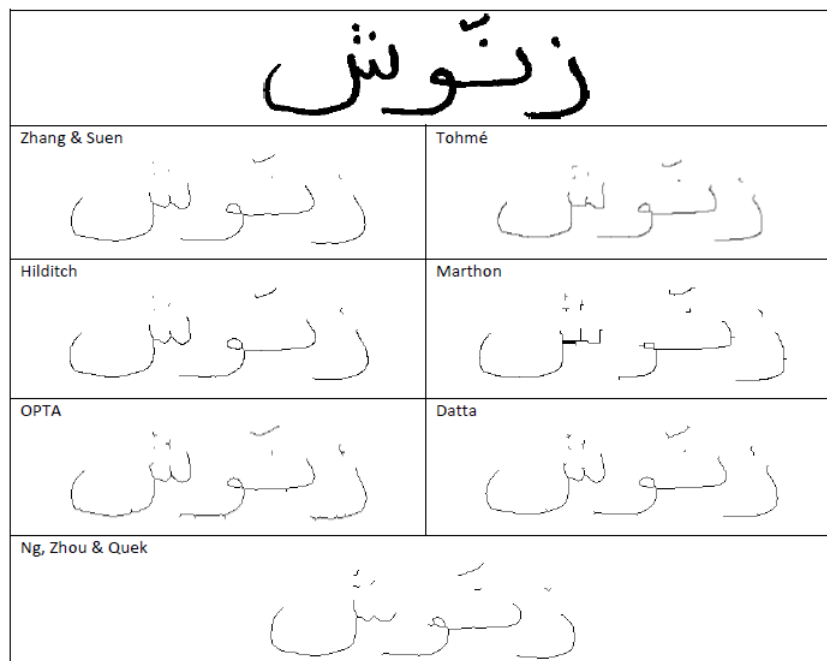


FIGURE 6.2 – Exemple 2 de comparaison visuelle entre les différents algorithmes de squelettisation.

### Critères calculables

Le tableau ?? et la figure ?? présentent les résultats obtenus avec les différents algorithmes de squelettisation implémentés en utilisant les différents critères calculables définis au chapitre 4. Les résultats figurant dans ce tableau concernent les moyennes obtenues pour chaque critère après squelettisation des 6537 images qui constituent l'ensemble set\_a de la base IFN/ENIT.

Alors que selon les critères visuels, l'algorithme de Zhang & Suen vient en première position suivi de celui de Hilditch, les critères calculables chamboulent toutes les prédictions offertes par la comparaison visuelle. En effet, les résultats dans le tableau ?? montrent plus de disparité dans les évaluations des différents algorithmes. Alors que les critères CM et SM maintiennent les résultats de l'évaluation visuelle, l'algorithme de Hilditch, perd la face devant celui de Zhou et al. quant aux critères LR et taux de squelettes acceptables et devant Datta pour le critère TM. Comme l'aspect visuel est déjà concrétisé par Zhang & Suen, nous


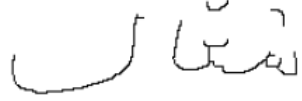





سَمَّال	
Zhang & Suen 	Tohmé 
Hilditch 	Marthon 
OPTA 	Datta 
Ng, Zhou & Quek 	

FIGURE 6.3 – Exemple 3 de comparaison visuelle entre les différents algorithmes de squelettisation.

تطارين ٦ نومبر	
Zhang & Suen 	Tohmé 
Hilditch 	Marthon 
OPTA 	Datta 
Ng, Zhou & Quek 	

FIGURE 6.4 – Exemple 4 de comparaison visuelle entre les différents algorithmes de squelettisation.

TABLE 6.1 – Comparaison suivant les critères calculables entre les différents algorithmes de squelettisation implémentés

	Nb. it.	Nb. del. pxl. (%)	Time	LR	TM	CM	SM	Accept. skt. (%)
Zhou et al.	5	86.0470	0.0420	0.4306	0.999918	21.464586	7.525011	99.5258
Zhang & Suen	6	83.9404	0.0184	0.4955	0.999276	18.961603	6.899342	55.8054
Hilditch	8	82.6916	0.1026	0.5344	0.999005	20.187395	7.416093	8.7349
Tohmé	12	81.1839	0.2487	0.5806	0.999526	22.522870	8.562184	0.5507
Marthon	48	81.9466	0.1369	0.5591	0.997525	27.352455	18.215848	9.1632
OPTA	11	83.3997	0.1409	0.5122	0.999613	33.658559	22.008720	24.7208
Datta	7	83.9985	0.0506	0.4944	0.999966	27.441334	13.627046	67.5998

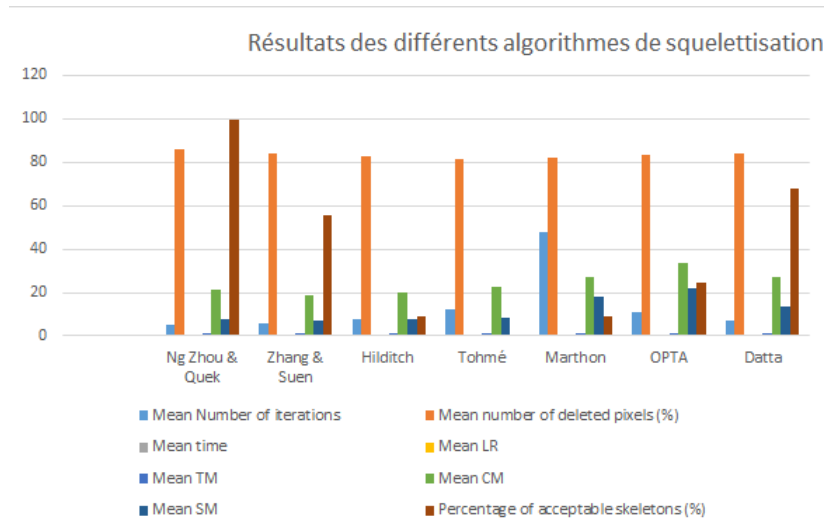


FIGURE 6.5 – Comparaison entre les différents algorithmes de squelettisation.

avons donc décidé de le comparer (après les différentes améliorations apportées) à celui qui présente les meilleurs résultats calculables c'est-à-dire l'algorithme de Zhou et al.

#### Quelques résultats du post-traitement appliqué à l'algorithme de Zhang & Suen

Les figures ?? à ?? montrent quelques résultats obtenus par la technique d'amélioration utilisée selon les critères visuels. La figure ?? montre le résultat sur un texte de la base AHDB. Les résultats selon les critères calculables sont résumés dans le tableau ??.

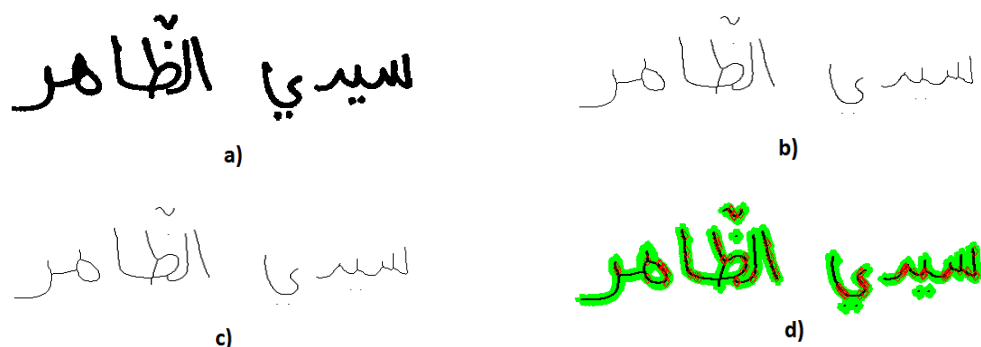


FIGURE 6.6 – Zhang & Suen Improved a) Un mot de la base IFN/ENIT b) Squelette avec Zhang & Suen c) Squelette avec Zhang & Suen Improved d) Visualisation (en rouge) des zones affectées par le post-traitement.



FIGURE 6.7 – Zhang & Suen Improved a) Latin et chiffres arabes b) Squelette avec Zhang & Suen c) Squelette avec Zhang & Suen Improved d) Visualisation (en rouge) des zones affectées par le post-traitement.

### 6.2.2 Ligne de base

De même que pour la squelettisation, et comme mentionné dans le chapitre 3, une comparaison entre les méthodes de détection de la ligne de base est faite avec des critères visuels suivis de critères calculables.

#### Critères visuels

La figure ?? montre les résultats de détection de la ligne de base sur quelques mots de la base IFN/ENIT. Bien que la méthode de squelette dans le premier mot donne une ligne de base plus proche de celle donnée

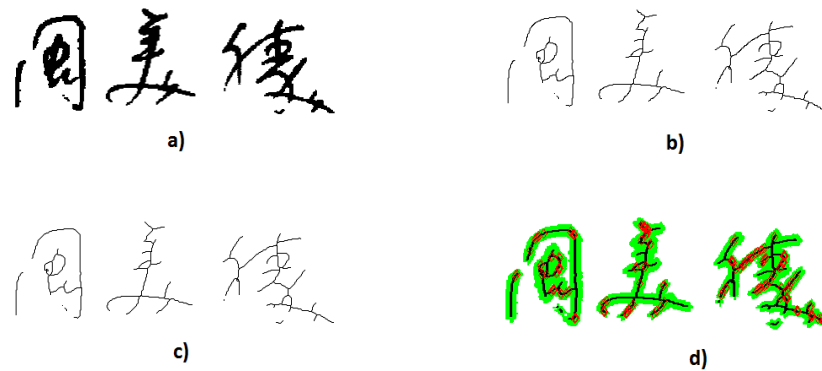


FIGURE 6.8 – Zhang & Suen Improved a) Chinois b) Squelette avec Zhang & Suen c) Squelette avec Zhang & Suen Improved d) Visualisation (en rouge) des zones affectées par le post-traitement.

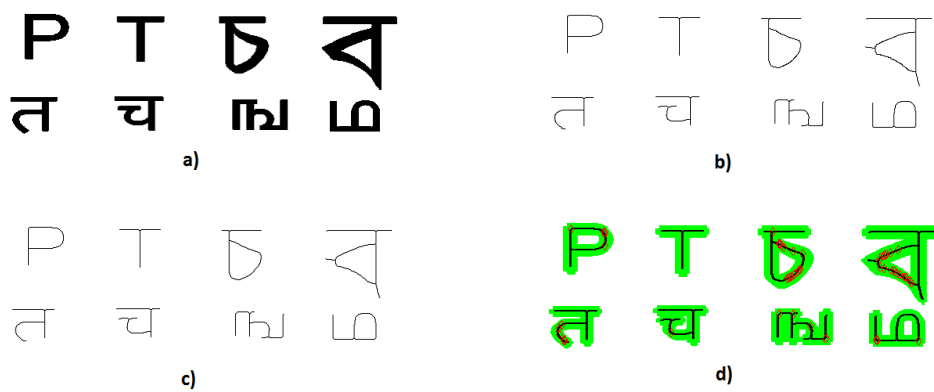


FIGURE 6.9 – Zhang & Suen Improved a) Alphabet Coréen b) Squelette avec Zhang & Suen c) Squelette avec Zhang & Suen Improved d) Visualisation (en rouge) des zones affectées par le post-traitement.

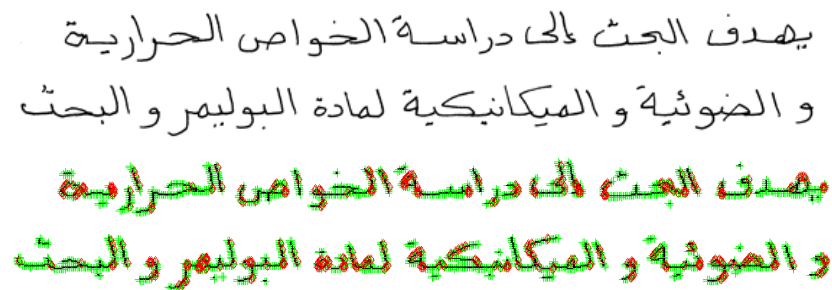


FIGURE 6.10 – Un texte de la base AHDB et l'algorithme de Zhang & Suen amélioré.

TABLE 6.2 – Résultats obtenus après application du post-traitement suivant les critères calculables

	Nb. del. pxl. (%)	Time	LR	TM	CM	SM	Accpt. skt. (%)
Zhang & Suen	83.9404	0.0174	0.4955	0,999276	18,961603	6,899342	55.8054
Zhang & Suen Improvement 1	84.6927	0.0621	0.4722	0,999698	18,961603	6,899342	91.3569
Zhang & Suen Improvement 2	85.2551	0.1589	0.4549	0,999916	18,961603	6,899342	99.3881
Zhou et al.	86.0470	0.0420	0.4306	0,999918	21,464586	7,525011	99.5258

dans la vérité terrain (ground truth) donnée dans la base, nous remarquons bien comment agit la présence des descendants sur la position de la ligne de base extraite par la méthode de projection et entropie, et la sensibilité de la méthode d'entropie liée à l'angle  $\theta$  choisi (l'intervalle  $[-\theta, +\theta]$ ). Quant au deuxième mot, et en l'absence des jambes, les deux premières méthodes donnent de bons résultats, exceptée la méthode de Hough qui donne dans certains cas des lignes plus inclinées et ceci est lié à la plus grande valeur du pic dans l'accumulateur. Dans le troisième mot où le mot est incliné, toutes les méthodes donnent des lignes proches de la ligne citée dans la base IFN/ENIT, et ceci dépend aussi de l'intervalle de variation de l'angle de rotation choisi quant-à la méthode de projection et l'entropie.

L'avantage majeur de la méthode Det-Boucle est la non sensibilité à l'inclinaison de l'écriture (figure ??), ni à la présence fréquente des signes diacritiques (figure ??), ni à la présence des descendants et ascendants (figure ??).

#### Critères calculables

La table ?? montre les résultats obtenus pour les différentes méthodes de détection de la ligne de base sur l'ensemble set\_a de la base IFN/ENIT (tableau ??).

Chapitre 6. Résultats expérimentaux

Méthodes d'extraction	Présence des descendants	Lettres isolées	Mot incliné
Ligne de base IFN/ENIT			
Projection			
Entropie			
Squelette			
Hough			
Cont-inf			
Det-Boucle			

FIGURE 6.11 – Lignes de base obtenues par les différentes méthodes.

السؤال      السؤال

FIGURE 6.12 – Ligne de base par la méthode de projection en vert, et ligne de base par la méthode Det-Boucle en rose.

أكورة      أكورة  
(a)                      (b)

FIGURE 6.13 – (a) Ligne de base par la méthode de projection (b) ligne de base par la méthode Det-Boucle.

زريق      زريق

FIGURE 6.14 – Ligne de base par la méthode de projection en vert, et ligne de base par la méthode Det-Boucle en rose.

TABLE 6.3 – Statistiques de l’ensemble `set_a` de la base IFN/ENIT, sur lequel on a évalué nos résultats

Taille du lexique	Nombre de scripteurs	Nombre d’échantillons
436	411	6537

TABLE 6.4 – Résultats obtenus par les différentes méthodes de détection-redressement de la ligne de base sur l’ensemble `set_a` de la base IFN/ENIT

	Projection	Entropie	Squelette	Hough	Cont-Inf	Det-boucles
Taux (%)	66,5	63,5	56,06	64	67,6	76,93

### 6.3 Segmentation

Dans ce qui suit, nous allons donner quelques résultats concernant les trois algorithmes de segmentation que nous avons implémentés. Nous tenons à préciser que, depuis l’acquisition de la base de données IFN/ENIT, nous nous focalisons surtout sur le dernier algorithme parce qu’il est plus général et n’est pas limité à une écriture particulièrement contrainte.

#### 6.3.1 Écriture contrainte

Nous avons implémenté deux algorithmes que nous avons décrit dans le chapitre ???. Nous avons effectué les tests sur un ensemble de textes écrits par quelques étudiants. Nous avons, pour chaque caractère, encodé les caractéristiques décrites dans la section ?? du chapitre ?? dans un vecteur binaire : le vecteur caractéristique. Ces vecteurs ont servi à l’apprentissage de quatre réseaux de Kohonen [TamenDrias2007] qui ont donné, pour un ensemble de tests indépendant de caractères pré segmentés, un taux de reconnaissance moyen de 92,43%. Le tableau ?? donne des détails sur les bases utilisées dans ce but.

Les résultats obtenus sur les parties segmentées issues de textes dont

TABLE 6.5 – Bases utilisées pour l’apprentissage des réseaux de Kohonen

Début	Milieu	Fin	Isolés
230	194	152	227

TABLE 6.6 – Segmentation avec et sans squelettisation sur écriture contrainte

Algorithme	Taux de reconnaissance (%)
Sans squelettisation	85.36
Avec squelettisation	88.31

la taille en mots est de 90 et en caractères de 267 sont exprimés dans le tableau ??.

### 6.3.2 Écriture non contrainte

La figure ?? montre le résultat de notre segmentation sur un mot de la base IFN/ENIT. Pour peu que le mot soit assez bien écrit, sans caractères susceptibles d'être sur-segmentés, notre segmentation est presque parfaite.



FIGURE 6.15 – Le résultat de segmentation de notre algorithme sur un mot bien écrit sans caractères d'être susceptible d'être sur-segmentés.

Le résultat de notre algorithme de segmentation sur une ligne de texte de la base AHDB est montré en figure ?. Le texte original ainsi que le résultat de la segmentation en PAWs sont donnés dans la figure ?.

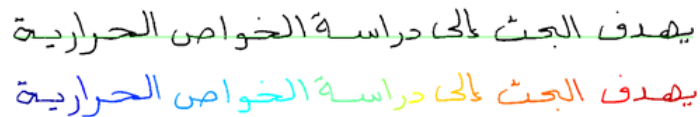


FIGURE 6.16 – En haut : une ligne de texte de la base AHDB. En bas, segmentation de la ligne en PAWs.

Nous remarquons que la lettres "sin" dans le mot dans le premier mot de la phrase a été segmentée en trois entités différentes.

Les résultats de segmentation en PAW et en caractères obtenus dans

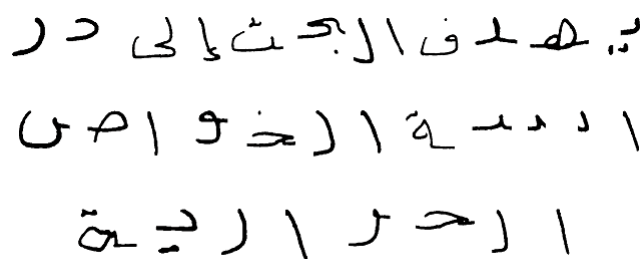


FIGURE 6.17 – La collection de caractères tirés de la segmentation de la ligne de texte de la figure ??.

TABLE 6.7 – Segmentation en PAW sur la base IFN/ENIT

nb words	nb of paws	correct	over	under
200	879	85,8931	8,8737	5,2332
2000	8485	81,8975	10,3359	7,7666

les deux bases IFN/ENIT et AHDB sont montrés dans les tableaux ?? à ??.

La figure ?? compare nos résultats à ceux trouvés dans [Elzobietal2014]. Il faut dire que ces résultats sont justes à titre indicatifs et ne peuvent pas prétendre être rigoureux.

Pour être rigoureux, il faut :

1. Prendre les mêmes ensembles d’images de comparaison.
2. Utiliser les mêmes algorithmes de squelettisation.
3. Nous avons comparé les résultats de squelettisation de l’algorithme de Zhang & Suen amélioré avec les résultats de squelettisation offerts par la base de données IESK-arDB. Nos résultats sont nettement meilleurs.

TABLE 6.8 – Segmentation en caractères sur la base IFN/ENIT

Algorithme	nb words	nb of chars	correct	over	under
Zhang & Suen	2000	15972	82,9389	12,4593	4,6018
Improved Zhang & Suen	2000	15972	83,8845	8,2049	7,9106

TABLE 6.9 – Segmentation en caractères sur la base AHDB

Algorithme	nb words	nb of chars	correct	over	under
Zhang & Suen	1995	15972	77,6751	10,0934	12,2316
Improved Zhang & Suen	1995	15972	77,4883	13,6695	8,8422

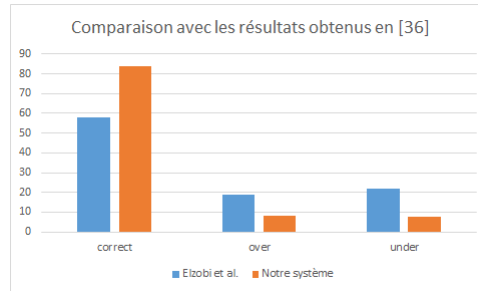


FIGURE 6.18 – Comparaison des résultats de segmentation avec ceux de [Elzobietal2014] sur la base IFN/ENIT.

## 6.4 Extraction de caractéristiques

### 6.4.1 Analyse en ondelettes

#### Base TBK

Le tableau ?? montre les meilleurs résultats obtenus pour la base TBK. Nous remarquons que pour tous les ensembles 'début', 'fin', 'isolé' et 'milieu', tous les paramètres réunis donnent le meilleur résultat avec la dimension [160,160] et la famille *Coiflet1* avec un niveau de décomposition 6 et un nombre de valeurs propres égal à 5.

Dans le tableau ??, 'All' signifie que tous les paramètres cités dans la section ?? sont pris en considération pour avoir le résultat.

TABLE 6.10 – Taux de classification de la base (TBK)

Ensemble	Paramètres Statistiques	Size	Wname	Dec	K	Régression
Début	All	[160,160]	coif1	6	5	91.9334
Fin	All	[160,160]	coif1	6	5	91.8482
Isolés	All	[160,160]	coif1	6	5	92.8837
Milieu	All	[160,160]	coif1	6	5	91.1191

TABLE 6.11 – Architecture du réseaux de neurones

Nb. de couche cachée	Nb. de neurones	Fonction d'apprentissage	Fonction de transfert
1	27	Trainlm	Sigmoid

TABLE 6.12 – Meilleurs Taux de classification (IFN/ENIT)

Paramètres Statistiques	Size	ch	Wname	Dec	K	Régression
123	[32,128]	2	coif1	7	0	0.936741
1234	[32,128]	2	coif1	6	5	0.935858
23	[64,256]	1	coif1	7	0	0.927057
12	[64,256]	1	coif1	7	0	0.919200
13	[32,128]	2	coif1	5	0	0.903371
3	[64,256]	1	coif1	6	0	0.895360
4	[32,128]	1	coif1	7	5	0.889385
2	[32,128]	2	coif1	7	0	0.862360
1	[64,256]	2	sym4	6	0	0.758937

### Base IFN/ENIT

Dans le tableau (Tableau ??), on montre le taux de classification par ordre décroissant de classification. Le taux le plus élevé est atteint avec la famille *Coiflet* avec un niveau de décomposition qui varie entre 6 et 7 avec les paramètres moyenne-écart-type et énergie avec la configuration décrite dans le tableau ??.

Dans le tableau ??, on montre les configurations testées pour réaliser ce système de reconnaissance pour chaque base de données. La conclusion globale que l'on peut tirer de toutes ces expérimentations pour les deux bases de données peut être énoncée comme suit :

- Concernant les deux bases de données, la famille d'ondelettes *Coiflet* a présenté les meilleurs résultats quels que soient les paramètres pris en considération.
- Le meilleur paramètre pour les deux bases est celui prenant en compte les valeurs propres des images de décomposition.
- La meilleure combinaison pour la base TBK est celle qui englobe tous les paramètres étudiés.
- Pour la base IFN/ENIT, la meilleure combinaison est la combinaison

TABLE 6.13 – Les différentes configurations du système

Prétraitement	
Dimension de l'image (Bases TBK)	[64] [96] [128] [160] [192]
Dimension de l'image (Bases IFN/ENIT)	[32 128] [64 128]
Ondelettes	
Niveau de décomposition	[1,2,3,4,5,6]
Famille d'ondelettes	Daubichies, Symlet, Coiflet
Paramètres Statistiques	Moyenne, Écart Type, Énergie, Valeurs Propres
Standarisation	MapMinmax, MapStd
Réseaux de neurones	
Nombre de couche	Une seule couche
Nombre de neurones	[20-50] neurones
Fonctions d'apprentissages	trainlm, trainbr, trainscg
Fonctions de transfert	tansing, logsig

[moyenne, écart-type, énergie].

#### 6.4.2 DCT VS. SVD

Nous avons comparé entre les caractéristiques basées sur la DCT (Discrete Cosine Transform) avec la décomposition en valeurs singulières (SVD).

#### DCT

Les résultats sur la DCT sont exposés dans les tableaux ?? à ??.

TABLE 6.14 – Résultats de la DCT sur le taux de reconnaissance (%)-Méthode zigzag sur toute l'image.

Taille des images/Taille des vecteurs	20	60	144
Sans normalisation	94.88	<b>97.85</b>	97.82
Taille 32 × 128	94.64	<b>97.84</b>	97.74
Taille 64 × 256	94.47	97.76	<b>98.19</b>
Taille 80 × 80	93.96	97.63	<b>97.85</b>
Taille 160 × 160	94.66	<b>97.76</b>	97.24

## Chapitre 6. Résultats expérimentaux

TABLE 6.15 – Résultats de la DCT sur le taux de reconnaissance (%)- DCT Upper Left Corner (ULC) coefficients

Taille des images/Taille des vecteurs	48	81	144
Sans normalisation	97.48	97.66	<b>97.90</b>
Taille 32 × 128	97.49	97.74	<b>97.97</b>
Taille 64 × 256	97.62	97.64	<b>97.77</b>
Taille 80 × 80	97.17	97.61	<b>97.85</b>
Taille 160 × 160	97.34	<b>97.58</b>	<b>97.58</b>

TABLE 6.16 – Résultats de la DCT sur le taux de reconnaissance (%) - Block Based DCT ULC coefficients

Taille des images/Taille des vecteurs	96	144
Taille 32 × 128	97.47	<b>97.91</b>
Taille 64 × 256	97.84	<b>97.91</b>
Taille 80 × 80	97.40	<b>97.97</b>
Taille 160 × 160	97.56	<b>97.63</b>

TABLE 6.17 – Résultats de la DCT sur le taux de reconnaissance (%) - Block Based DCT Zigzag Coefficients

Taille des images/Taille des vecteurs	96	144
Taille 32 × 128	97.79	<b>97.88</b>
Taille 64 × 256	97.47	<b>97.60</b>
Taille 80 × 80	97.65	<b>97.74</b>
Taille 160 × 160	97.74	<b>98.10</b>

## SVD

Le tableau ?? résume les résultats obtenus avec la SVD.

TABLE 6.18 – Résultats de la SVD sur le taux de reconnaissance (%).

Taille des images	Taux (%) / Taille des vecteurs
Sans normalisation	45.51 / 24 <b>52.02 / 48</b>
Taille 80 × 80	51.96 / 60 <b>53.06 / 80</b>
Taille 160 × 160	54.72 / 120 <b>55.37 / 160</b>

### Combinaison DCT-SVD

Les tentatives de combinaisons entre les deux méthodes d'extraction de caractéristiques et les résultats obtenus sont exprimés dans le tableau ??.

L'approche hybride est appliquée dans notre système selon différentes techniques :

- DCT en premier : Dans cette technique nous présentons d'abord les coefficients de la DCT puis ceux de la SVD.
- SVD en premier : Ici, nous commençons par présenter les coefficients de la SVD puis ceux de la DCT.
- ALTERNÉE : La dernière technique sert à prendre un coefficient de la DCT puis un autre de la SVD dans le vecteur de caractéristique.

Nous respectons les règles suivantes :

- Pour chacune des techniques, le nombre de coefficients présenté dans la DCT est égal à ceux de la SVD.
- Nous avons fixé le nombre de neurones dans la couche cachée à 20 neurones.
- La taille des images est fixée à  $160 \times 160$ .

TABLE 6.19 – Résultats de la combinaison DCT-SVD sur le taux de reconnaissance (%).

Méthode / Nombre de coefficients	80	160
DCT en premier	96.80	<b>97.57</b>
SVD en premier	97.43	<b>97.54</b>
ALTERNÉE	96.99	<b>97.52</b>

Nous remarquons que :

- Les trois techniques d'hybridation donnent des moyennes de taux de reconnaissance élevés.
- La technique SVD en premier a donné les meilleurs résultats.
- Une petite amélioration dans la moyenne du taux de reconnaissance est constatée en utilisant 160 coefficients par rapport à 80 coefficients.

TABLE 6.20 – L’effet du redimensionnement dans le taux de reconnaissance

	CM	KM	LM	ZM
$32 \times 128$	<b>94.49</b>	84.40	87.32	80.90
$80 \times 80$	<b>94.41</b>	84.24	85.49	74.89

TABLE 6.21 – Les résultats des classifieurs K-NN versus RF pour les quatre OM

	CM	KM	LM	ZM
Order	10	10	10	25
Rec. Rate (%) (K-NN)	<b>94.49</b>	84.40	87.32	80.90
Rec. Rate (%) (RF)	94.08	89.49	80.90	74,56

### 6.4.3 Moments orthogonaux : comparaison entre les quatre moments

Dans le Tableau ??, nous résumons les meilleurs résultats obtenus avec deux tailles des images  $32 \times 128$  et  $80 \times 80$ . Comme les meilleurs résultats obtenus sont ceux avec la taille  $32 \times 128$ , cette dernière a été maintenue dans le reste des expérimentations.

Le taux de reconnaissance avec les classificateurs KNN et RF est donné dans le tableau ??.

Comme nous pouvons le remarquer, les résultats obtenus par les moments de Chebyshev ont été les meilleurs, suivis de ceux des moments de Legendre. La première amélioration que nous avons tentée a consisté en différentes combinaisons des deux OM (CM et LM). Les résultats obtenus sont présentés dans le Tableau ?? où  $f(C \text{ ou } L)$  et  $l(C \text{ ou } L)$  signifient respectivement *premiers* et *derniers* 10 ordres des moments de Chebyshev ou Legendre. fC-fL signifie les 10 premiers ordres CM suivis des 10 premiers ordres LM.

TABLE 6.22 – Les résultats obtenus par le classifieur K-NN et les combinaisons entre OM(CM & LM)

Sets	CM	fC-fL	fL-fC	fC-lL	fL-lC
set1(%)	<b>94.49</b>	<b>94.49</b>	<b>94.49</b>	<b>94.49</b>	46.54
set2(%)	<b>92.42</b>	<b>92.42</b>	<b>92.42</b>	<b>92.42</b>	31.03

TABLE 6.23 – Les meilleurs résultats obtenus en fonction des ordres des CM versus le temps d'exécution

CM	7	8	9	10	11	12	13
R	92.74	93.74	93.66	<b>94.49</b>	94.24	93.49	93.66
t(s)	0.0091	0.0106	0.0156	0.0203	0.0255	0.0299	0.0386

Comme nous pouvons le voir dans la Table ??, aucune amélioration n'a été obtenue en utilisant toutes les combinaisons entre les moments de Chebyshev avec ceux de Legendre. Plus encore, dans certains cas, le taux de reconnaissance a diminué. Les résultats impliquent que les moments OM fournissent des informations redondantes lorsqu'elles sont combinées.

Nous avons ensuite raffiné le processus de recherche pour trouver le meilleur ordre dans les CM qui assure le compromis entre la minimisation du temps de traitement et l'obtention de bons résultats. Comme nous avons déjà remarqué que cet ordre était aux environs de 10, nous avons calculé les caractéristiques CM aux ordres : 7-8-9-10-11-12-13. Les résultats sont présentés dans le Tableau ?. Le temps de traitement est calculé comme le temps moyen utilisé pour estimer pour les 1188 images de test les  $K$  plus proches voisins avec les quatre distances "Euclidienne", la distance de "Manhattan", la distance en "Cosinus" et la "Corrélation" pour  $K = 1, \dots, 10$ . Une question intuitive peut venir à l'un d'entre nous : comme les moments sont des fonctionnalités basées sur le contenu, si nous les appliquions à l'image de contour au lieu de l'image entière, cela améliorerait-il les résultats obtenus ? Nous avons testé les caractéristiques CM appliquées aux images de contour des deux ensembles et les résultats sont montrés dans le Tableau ?. Comme nous pouvons le voir, les résultats ne sont pas meilleurs. Ainsi, dans le reste des tests, seuls les moments CM appliqués à des images entières jusqu'à l'ordre 10 sont conservés avec les caractéristiques SCF.

TABLE 6.24 – Les résultats obtenus par le classifieur KNN avec les CM appliqués aux images d’origine (CM(Or)) et CM appliqués aux contours des images (CM(Ct))

Sets	CM(Or)	CM(Ct)
set1(%)	<b>94.49</b>	92.41
set2(%)	<b>92.42</b>	91.79

#### 6.4.4 Impact des caractéristiques basées sur le contour et les statistiques SCF

Dans ce qui suit, nous donnons les résultats obtenus lorsque la technique d’extraction de caractéristiques SCF proposée est utilisée. Comme nous l’avons déjà mentionné, la largeur des fenêtres, les chevauchements et le nombre de blocs à prendre dans chaque fenêtre ont été variés afin de parvenir à un compromis entre les détails des informations traitées et les pouvoirs de convergence et de généralisation des classificateurs. Trop de détails peuvent donner lieu à un processus d’apprentissage parfait mais long. Cependant, cela se traduit généralement par une très faible capacité de généralisation du système. Les différents paramètres ont été modifiés comme suit :

1. La largeur de la fenêtre a été variée entre 9 et 32 pixels avec un chevauchement allant jusqu’à environ la *largeur/3*.
2. Le nombre de blocs dans chaque fenêtre a pris une de ces valeurs :  $4 \times 4$ ,  $3 \times 4$ ,  $2 \times 4$  et  $2 \times 3$ .

Les résultats obtenus avec CM et SCF (séparément) sur set1 et set2 sont montrés dans le Tableau ???. Les derniers ont été obtenus avec 4 fenêtres glissantes avec un chevauchement de 3 pixels et des blocs de  $2 \times 3$  dans chaque fenêtre. Les informations sauvegardées dans chaque fenêtre ont été réduites au nombre de points d’intersection entre le contour de l’écriture et les droites allant du centre de gravité dans seulement quatre directions, avec l’exclusion des directions verticales et horizontales. L’exclusion de ces quatre directions (Nord, Sud, Est et Ouest) a été faite après plusieurs tests qui ont prouvé que ces informations n’améliorent pas le taux de reconnaissance. La taille du vecteur caractéristiques est avec cette configuration de 116.

TABLE 6.25 – Les meilleurs taux de CM et SCF sur set1 et set2 avec le classifieur K-NN

Sets	CM	SCF $4 \times (2 \times 3)$
set1	<b>94.49%</b>	93.91%
set2	<b>92.42%</b>	86.67 %

TABLE 6.26 – Les meilleurs taux (%) des six classificateurs sur l’ensemble dans [Khemirietal2014]

Khémiri	$ELM_{CM}$	$ELM_{SCF}$	$MLP_{CM}$	$MLP_{SCF}$	$SVM_{CM}$	$SVM_{SCF}$
92.19	89.10	<b>92.78</b>	89.15	88.12	89.09	87.90

#### 6.4.5 Comparaison entre MLP, SVM et ELM classifieurs avec CM et SCF caractéristiques

Afin de faire une comparaison finale avec certains résultats de l’état de l’art, nous avons pris l’ensemble des images utilisées dans [Khemirietal2014]. Tout d’abord, nous avons entraîné un réseau MLP avec une couche cachée de 30 neurones avec l’algorithme BP traditionnel sur l’ensemble d’apprentissage. Le deuxième classifieur consistait en un ensemble de SVM utilisant la méthode One-Against-All (OAA), avec un noyau RBF. Enfin, nous avons utilisé un ELM avec noyau RBF et 12 neurones aléatoires cachés. Ces classifieurs ont été entraînés séparément à l’aide des moments CM. Trois autres classificateurs (MLP, SVM et ELM) ont été entraînés avec les caractéristiques SCF. Les résultats de ces six classifieurs individuellement sont résumés dans le tableau ??.

### 6.5 Combinaison de classifieurs

Dans ce qui suit, nous allons décrire les résultats du travail que nous avons mené sur la combinaison de classifieurs dans le but d’améliorer la performance du système de reconnaissance [Tamenetal2017].

TABLE 6.27 – Les meilleurs taux (%) sur l’ensemble dans [Khemirietal2014] de la combinaison avec MLP et ELM

Khémiri	ELM	MLP
92.19	<b>94.66</b>	<b>94.83</b>

TABLE 6.28 – Résultats (%) de la combinaison basée sur les règles de combinaison par rapport au système dans [Khemirietal2014]

Khémiri	Borda	GWAR	ICWAR	ARM	MAR	MR	AR	DS	$AR_{GWAR,Borda,DS}$
92.19	96.53	96.14	96.08	95.40	95.85	94.49	96.14	96.08	<b>96,82</b>

### 6.5.1 Résultats des règles de combinaison proposées des classifieurs MLP, SVM et ELM

Bien que les résultats globaux utilisant le classificateur  $ELM_{SCF}$  soient meilleurs que ceux cités dans [Khemirietal2014], les résultats par classe ne sont pas très satisfaisants. Afin d’améliorer les taux de reconnaissance intra-classe, les premières combinaisons tentées étaient celles utilisant un MLP et un ELM. Les résultats obtenus sont présentés dans le Tableau ???. De plus, nous avons proposé plusieurs règles déjà définies dans la section ??? pour combiner les classifieurs. Les résultats de ces règles de combinaison sur l’ensemble de données utilisé dans [Khemirietal2014] sont présentés dans le tableau ???.

Nous notons que chacune des règles de combinaison a permis une amélioration importante que soit sur le taux global, ou sur les taux intra-classe et a permis de bien meilleurs résultats que ceux de [Khemirietal2014] (voir le tableau ???). Borda compte a permis les meilleurs résultats dans le premier niveau de combinaison. Il est suivi par les règles AR et GWAR. La combinaison Dempster-Shafer est arrivée en troisième position avec un taux de 96,08 %.

### 6.5.2 Combinaison à un niveau supérieur

La règle de combinaison à un niveau supérieur qui est la règle moyenne (AR) est appliquée sur les résultats donnés par les règles GWAR, Borda

TABLE 6.29 – Résultats détaillés (%) des meilleures règles par rapport à [Khemirieta2014]

City Name	Class number	Khémiri	Our System			
			GWAR	Borda	DS	$AR_{GWAR,Borda,DS}$
الرضاع	1	97.60	97.60	<b>99.20</b>	95.20	<b>99.20</b>
الخليج	2	88.69	98.26	<b>100</b>	98.26	<b>100</b>
نقعة	3	95.61	96.49	98.25	96.49	<b>98.25</b>
شعال	4	95.57	96.46	96.46	95.57	<b>97,34</b>
مارث	5	91.15	<b>99.11</b>	<b>99.11</b>	<b>99.11</b>	<b>99.11</b>
شماخ	6	90.65	97.20	97.20	<b>99.06</b>	98,13
زنوش	7	93.40	96.23	<b>98.11</b>	96.23	97,17
الدخانية	8	92.45	97.17	98.11	97.17	<b>99,06</b>
الفايض	9	87.50	<b>94.23</b>	92.31	93.27	92.31
أكودة	10	89.90	98.99	98.99	<b>100</b>	<b>100</b>
سبعة آبار	11	94.90	<b>98.98</b>	<b>98.98</b>	97.96	<b>98.98</b>
سيدي ابراهيم الزهار	12	90.72	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
المرناقية ٢٠ مارس	13	94.50	98.90	<b>100</b>	98.90	<b>100</b>
شتاوة صحراوي	14	96.34	<b>100</b>	98.78	98.78	<b>100</b>
الفكة	15	92.98	91.23	<b>94.74</b>	89.47	92,98
أوتيك	16	<b>91.11</b>	86.96	89.13	89.13	86,96
الفحص	17	<b>91.30</b>	89.13	<b>91.30</b>	84.78	89,13
الشرايع	18	<b>95.55</b>	75.55	75.55	77.78	77,78
حي الإنطلاقة	19	90.00	97.50	<b>100</b>	97.50	97,50
شواط	20	86.11	<b>88.89</b>	75.00	97.22	86,11
حي التضامن	21	<b>90.00</b>	85.00	85.00	<b>90.00</b>	85,00
<b>Global rates (%)</b>		92.19	96.14	96.53	96.08	<b>96.82</b>

et Dempster-Shafer. Cette combinaison a surpassé toutes les autres règles. Les résultats détaillés sont résumés dans le tableau ???. Les résultats comprennent aussi ceux de GWAR, Borda, Dempster-Shafer. Ici, nous devons rappeler que :

1. Tous les classifieurs utilisés dans notre système exigent des ensembles de données bien équilibrés. L'ensemble utilisé dans [Khemirieta2014]

TABLE 6.30 – Résumé des taux de reconnaissance (%) obtenus

Methods	Min	1st Qu	Median	Mean	3rd Qu	Max
Khemiri	<b>86.11</b>	90.00	91.30	92.19	94.90	97.60
GWAR	75.55	91.23	97.17	94.47	98.90	<b>100.00</b>
Borda	75.00	<b>92.31</b>	98.11	94.58	99.11	<b>100.00</b>
DS	77.78	93.27	97.17	94.85	98.78	<b>100.00</b>
$AR_{GWAR,Borda,DS}$	77.78	<b>92.31</b>	<b>98.13</b>	<b>95.00</b>	<b>99.20</b>	<b>100.00</b>

est très déséquilibré. La représentativité de ses classes varie de 60 à 374 avec une variabilité de 0,84 ce qui est trop pour de tels systèmes [Huang2014].

2. Malgré cela, notre système a réalisé beaucoup mieux que celui cité dans [Khemirietal2014].
3. La plupart des noms de villes qui n'étaient pas bien reconnus par notre système sont ceux avec un petit nombre d'images représentatives.

Pour mettre en valeur la performance de la méthode proposée, nous donnons quelques statistiques sommaires dans le tableau ???. Nous donnons le minimum (Min), le maximum (Max), la moyenne (moyenne), la médiane (médiane), le premier quartile (1er Qu) et le troisième quartile noté (3ème Qu).

On peut dire qu'en général notre système  $AR_{GWAR,Borda,DS}$  a bien répondu. Les résultats sont très intéressants et démontrent les avantages de la classification proposée dans la reconnaissance de l'écriture manuscrite arabe.

### 6.5.3 Comparaison supplémentaire

Nous allons dans ce qui suit comparer notre méthode avec quelques techniques bien connues pour la reconnaissance de mots manuscrits arabes. Notre objectif est de montrer l'efficacité de notre méthode par rapport à d'autres méthodes bien connues dans la littérature. La table ??? résume la comparaison avec certains des articles de pointe cités dans [ParvezMahmoud2013]. Le système proposé réussit à trouver des résultats avec un taux de reconnaissance global égal à **96.82%** dans la

TABLE 6.31 – Comparaison avec certains des résultats cités [ParvezMahmoud2013]

Author(s)	Classifier	Data	Accuracy
Ben Cheikh <i>et al.</i> 2008 [BenCheikhetal2008]	Transparent Neural Networks	2,250 words	91%
Haboubi <i>et al.</i> 2009 [Haboubi2009]	ANN	16,107 words	87.1 %
Azizi <i>et al.</i> 2010 [Azizi2010]	Multi-classifiers (SVM, K-NN, ANN, HMM)	10,000 words	93.96%
Chen <i>et al.</i> 2010 [Chenetal2010]	SVM	7,346 PAWs	82.7 %
Chergui, 2013 [Chergui2013]	MLP, RBF, Fuzzy ART	3,000 words	90.1
Khémiriet <i>et al.</i> 2014 [Khemirietal2014]	PGM	5,279 words	92.19%
Nemmour and Chibani 2017 [NemmourChibani2017]	AIRS	6439	85.2
<b>Our system</b>	Multi-classifier (MLP, SVM, ELM)	5,279 words	<b>96.82%</b>

base de données IFN/ENIT.

#### 6.5.4 Analyse statistique ANOVA

L'analyse ANOVA (Analyse de la variance) est un outil statistique utilisé pour analyser l'effet des variables qualitatives sur une variable quantitative. Dans notre cas, nous utilisons les taux de reconnaissance intra-classe pour comparer l'effet des classifieurs dans le domaine de la reconnaissance des mots manuscrits arabes. Nous donnons les résultats de l'analyse ANOVA lorsque nous comparons les quatre classifieurs proposés  $GWAR$ ,  $DS$ ,  $Borda$  et  $AR_{GWAR,Borda,DS}$  et celui décrit dans [Khemirietal2014]. Les tables de ?? à ?? décrivent les quatre tests ANOVA où la colonne  $SS$  est la somme des carrés. La colonne  $df$  représente le degré de liberté. La colonne  $MS$  est le carré moyen. La valeur en gras représente la valeur qui nous intéresse (p-value) pour pouvoir donner une interprétation et une analyse des résultats. La p-valeur est d'environ 0,019 et 0,059 pour les méthodes  $GWAR$  et  $DS$ . De plus, la p-valeur des deux dernières tables ANOVA est autour de 0.0001. Ce

TABLE 6.32 – ANOVA test : GVAR vs. Khémiri

SS	df	MS	F-Statistic	p-value
90.923	1	90.923	5.992	<b>0.019</b>

TABLE 6.33 – ANOVA test : DS vs. Khémiri

SS	df	MS	F-Statistic	p-value
119.859	1	119.859	3.775	<b>0.059</b>

résultat indique que les valeurs des taux de reconnaissance intra-classe produits par les méthodes Borda et  $AR_{GVAR,Borda,DS}$  sont significativement différentes des valeurs de reconnaissance par la méthode Khémiri et al. Cela signifie que la qualité des résultats utilisant notre système est statistiquement meilleure que celle de [Khemirietal2014].

### 6.5.5 Discussion

Ce travail propose un système de classification efficace pour la reconnaissance des mots manuscrits arabes. Le système est validé sur la base de données IFN/ENIT. Les résultats obtenus sont compétitifs. Cela s'explique par : (1) La combinaison avec les méthodes de Machine Learning en utilisant les classificateurs MLP et ELM. Dans cette combinaison, on apprend à la machine à éviter la mauvaise classification des classifieurs utilisés. Nous utilisons les résultats obtenus lors de l'étape d'apprentissage, nous enseignons à la machine la vraie classe pour chaque combinaison de résultats des classifieurs. (2) Les règles que nous avons utilisées pour combiner les classifieurs telles que les règles AR et GPAR sont mathématiquement prouvées pour donner les meilleurs résultats. Dans la

TABLE 6.34 – ANOVA test : BORDA vs. Khémiri

SS	df	MS	F-Statistic	p-value
214.114	1	214.114	23.460	<b>0.0001</b>

TABLE 6.35 – ANOVA test :  $AR_{GWAR,Borda,DS}$  vs. Khémiri

SS	df	MS	F-Statistic	p-value
247.525	1	247.525	29.830	<b>0.0001</b>

première règle, nous calculons la moyenne et, dans la seconde, nous calculons la moyenne pondérée avec les taux de reconnaissance réalisés par les classifieurs lors de l'étape d'apprentissage. Ces paramètres donnent plus de poids aux classifieurs qui ont mieux appris les classes et, par conséquent, la combinaison surpasse les meilleurs résultats individuels. (3) Le méthode Borda est une règle largement utilisée dans le domaine de la combinaison de plusieurs sources d'information et a déjà donné de bons résultats dans différents domaines. (4) La règle de Dempster-Shafer est également bien connue dans le domaine de la combinaison de plusieurs sources d'information, et pallie même aux cas où il y a un manque dans les informations.

Dans ce travail, nous avons d'abord, comparé les moments de Legendre, Krawchouk, Zernike et Chebyshev dans leurs capacités discriminatoires pour décrire des mots arabes manuscrits. Les expériences montrent que les moments de Chebyshev (CM), suivis des moments de Legendre (LM), ont donné les meilleurs résultats concernant la base de données IFN/ENIT. Certaines combinaisons ont été évaluées entre les CM et les LM. Aucune amélioration n'a été réalisée confirmant que ces moments donnent des informations redondantes lorsqu'ils sont combinés. Les caractéristiques statistiques basées sur le contour (SCF) ont également été utilisées pour améliorer les fonctions CM pour discriminer nos images. Deuxièmement, nous avons utilisé trois classifieurs : MLP, SVM et ELM. Chacun de ces classifieurs a été entraîné avec CM et SCF individuellement, ce qui a engendré six classifieurs :  $MLP_{CM}$ ,  $MLP_{SCF}$ ,  $SVM_{CM}$ ,  $SVM_{SCF}$ ,  $ELM_{CM}$  Et  $ELM_{SCF}$ . Dans cette étape, nous avons utilisé un ensemble de données non homogène afin de faire une comparaison avec certains travaux de l'état de l'art. Plusieurs règles de combinaison des six classifieurs ont obtenu des résultats de haute qualité. Dans l'ensemble, le système proposé a réussi

à trouver des résultats concurrentiels. Comme déjà dit, cette supériorité est due à la combinaison de caractéristiques appropriées avec des classifieurs appropriés.

## 6.6 Conclusion

Ce chapitre a été dédié aux différents résultats obtenus dans quelques unes des contributions que nous avons apportées dans le domaine de la reconnaissance du manuscrit arabe. Le choix des contributions a été dur. En effet, durant ces années de travail dans le domaine, les champs d'investigation ont été très diversifiés. Ceci, à notre avis, a été le premier obstacle à une finalisation moins tardive des travaux entrepris. Nous travaillons actuellement sur un axe très différent et prometteur qui concerne la normalisation du manuscrit, nous allons en donner un aperçu dans les perspectives de ce travail.

# Conclusion Perspectives

## Conclusion

Les résultats présentés dans ce document représentent le fruit de plusieurs années de travail dans le domaine de la reconnaissance du manuscrit arabe. Notre contribution peut se focaliser en différents axes essentiels :

- La reconnaissance holistique de mots arabes manuscrits. Dans ce cadre, nous avons proposé des règles de combinaison de classifieurs pour une reconnaissance plus effective de mots arabes manuscrits issus de différentes bases de données. Nous avons pour ce faire, utilisé les moments de Chebyshev (CM) soutenus par des primitives métriques et statistiques basées sur le contour de l'écriture (CSF). Nous avons utilisé lors de la reconnaissance trois classifieurs communs dans le domaine qui sont le MLP, les SVM et les ELM. Ces classifieurs ont été entraînés avec les CM et les CSF individuellement donnant naissance à six classifieurs que nous avons combinés au niveau décisionnel. Bien que les classifieurs pris dans ce type de combinaison soient très exigeants en termes d'homogénéité des ensembles de données utilisés dans l'apprentissage, notre système a été testé sur un ensemble très hétérogène et a donné des résultats qui ont été compétitifs par rapport à ceux de l'état de l'art [**Tamenetal2017**].
- Dans le but d'assurer une bonne segmentation en caractères de l'écriture.

ture arabe manuscrite, nous avons implémenté et effectué une étude comparative entre sept des algorithmes de squelettisation les plus connus dans le domaine. Nous avons proposé une étape de post-traitement pour celui qui a montré les meilleurs résultats suivant les différents critères d'évaluation. Ceci a donné naissance à un algorithme de squelettisation (TZS) qui a révolutionné les algorithmes étudiés.

- La segmentation de l'écriture arabe manuscrite en caractères a été abordée sous plusieurs aspects et des algorithmes de segmentation et de concaténation ont été proposés suivant les contextes.
  - Concernant la reconnaissance de l'écriture manuscrite contrainte deux algorithmes de segmentation ont été proposés :
    - Avant squelettisation.
    - Après squelettisation.

L'extraction de caractéristiques discriminantes dans le cas de l'écriture manuscrite contrainte a été mimée du comportement humain quand il a la tâche de lire un tel script. Les primitives ont donc été du type visuel, et respectent les concepts inhérents à l'écriture contrainte. Il est à noter dans ce cas, que cette approximation en prenant une écriture contrainte est très plausible dans le cadre de la communication homme-machine parce qu'il a été démontré dans plusieurs recherches que quelle que soit l'aisance que peut ressentir l'être humain quand il est en face de la machine, celui-ci n'est jamais aussi libre dans son comportement que s'il était en face d'un confrère. En effet, ce dernier fait toujours plus d'efforts pour faire comprendre la machine qu'il ne le fait pour un vis-à-vis humain.

- La concaténation a été réalisée au niveau vecteurs caractéristiques où une addition spéciale de ces derniers a été mise en place. Cette addition spécifique des vecteurs de primitives a été réalisée moyennant un ensemble de règles découlant des règles d'écriture du script arabe.
- La validation de la segmentation a été assurée par deux classifieurs différents :
  - Les réseaux de Kohonen [**TamenDrias2007**].

- Les réseaux de neurones multi-couches [].
- La base IFN/ENIT représente une base de données où les images écrites présentent beaucoup de liberté d'écriture. Pour assurer la segmentation de cette dernière en caractères, une bonne squelettisation s'est avérée nécessaire. Les étapes qui ont permis cette segmentation peuvent être résumées par :
  - L'algorithme de Zhang & Suen avec le post traitement proposé a été utilisé pour la squelettisation de l'écriture.
  - La détection et le redressement de la ligne de base d'écriture s'est aussi avérée nécessaire dans ce cas. En effet, le filtrage des points candidats à la segmentation a été en particulier réalisé sur le critère de proximité de ces derniers par rapport à la ligne de base.
- La concaténation des parties segmentées a été effectuée au niveau images. Donc, dans ce cas-là, une réelle reconstitution des images concaténées a été réalisée. Ceci a induit une obligation de recalculer les vecteurs caractéristiques sur les images reconstituées.
- Pour pouvoir assurer un apprentissage du système basé segmentation et en l'absence d'une base de données gratuite disponible pour la recherche, on s'est vue obligée d'en réaliser une qu'on a nommée TBK. Plusieurs techniques de classification ont aussi été testées allant des réseaux de neurones de type perceptron multi-couches aux SVM (Support Vector machines), aux ELM (Extreme Learning Machine). Les réseaux LVQ (Learning Vector Quantization) et les réseaux RBF (Radial Basis Function) ont aussi été utilisés. Une étude comparative intensive a été menée pour décider du mieux qui a été vu jusque-là.

Nous ne pouvons pas finir ce document sans donner un ensemble de perspectives et en fait, nous allons même décrire plus loin, la nouvelle vision sur laquelle nous travaillons depuis quelques années mais qui n'a pas été citée plus avant dans ce document parce qu'elle est toujours à l'état expérimental.

## Perspectives

Concernant les perspectives, nous allons axer nos propositions sur le système complet assurant la segmentation et la reconnaissance du manuscrit arabe. Celles-ci peuvent être résumées comme suit :

1. Concernant l'extraction des caractéristiques, nous proposons d'extraire le maximum de caractéristiques statistiques, topologiques et autres et d'utiliser des méta-heuristiques pour optimiser soit :
  - (a) la sélection,
  - (b) la combinaison de caractéristiques de types différents ou
  - (c) la sélection et la combinaison de caractéristiques.
2. Utiliser des méta-heuristiques pour optimiser le choix des points de segmentation effectifs.
3. Chaque choix d'un ensemble de points de segmentation définira un chemin de lecture pour l'entité segmentée (PAW, mot etc.). Cet état est décrit dans la figure ?? qui montre les différents chemins de lecture d'une PAW.
4. Définir un coût du chemin. Pour chaque ensemble de points de segmentation, il s'agira de minimiser la fonction de coût du chemin pour assurer la meilleure segmentation de l'entité.

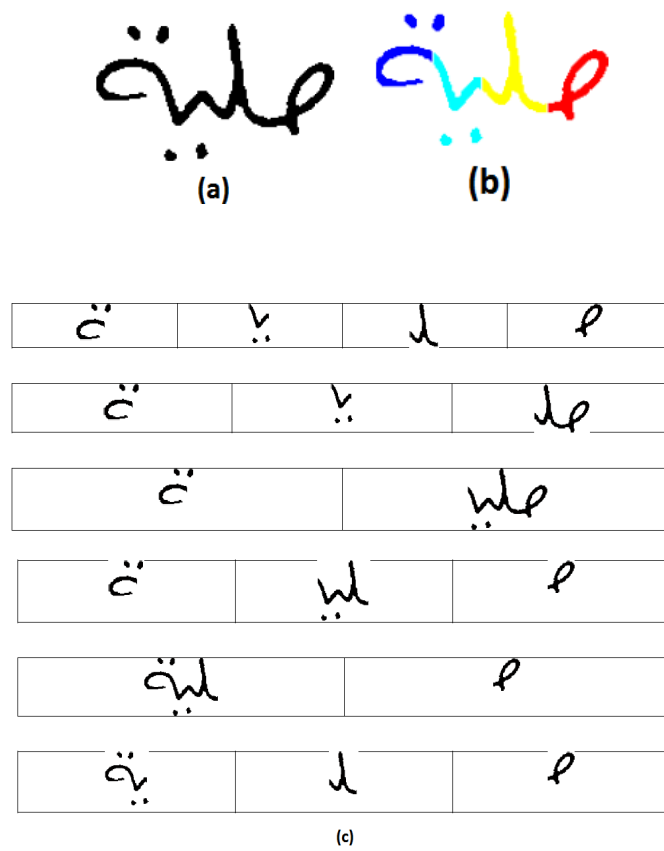


FIGURE 1 – Différents chemins de lecture d'une PAW (a) PAW originale (b) PAW segmentée par notre algorithme (c) Les différents chemins de lecture des parties segmentées

## Table des figures

## Liste des tableaux

# Table des matières

---

## Vers une reconnaissance efficace du manuscrit Arabe

---

### Résumé

On s'intéresse aux processus de segmentation, d'extraction de caractéristiques et de classification dans un système de reconnaissance du manuscrit arabe. Deux algorithmes de segmentation ont été proposés avec possibilité de concaténation des parties segmentées afin de reconstituer les caractères sur-segmentés. L'absence d'une base de données de caractères pré segmentés nous a amenée à en créer une. Les caractéristiques extraites sont du type statistique et structurel. Nous avons utilisé différents classifieurs tels que les réseaux de neurones, les SVM, les ELM et les HMM. Un processus de combinaison des résultats des classifieurs est proposé dans le but d'améliorer les résultats obtenus par les différents classifieurs individuellement.

---

**Mots-clefs** manuscrit Arabe, RN, SVM, ELM, HMM

---

## Towards an effective recognition of Arabic handwriting

---

### Abstract

We describe here the segmentation, feature extraction and classification of an Arabic handwriting recognition system. During the segmentation process, two algorithms have been implemented with possibility of concatenating the segmented portions in order to restore the characters that are over-segmented. The absence of a pre segmented characters database has led us to create one. The features extracted from the images of words or characters are of statistical and structural types. The classification process involves the use of different classifiers such as neural networks, SVM, ELM and HMM. A method of combination of classifiers' results is proposed in order to improve the results obtained by different classifiers individually.

---

**Keywords** Arabic manuscript, RN, SVM, ELM, HMM

---

---

Université des Sciences et de la Technologie Houari  
Boumediene USTHB

---

