

People's Democratic Republic of Algeria  
Ministry of Higher Education and Scientific Research  
University of 8 May 1945-Guelma-  
Faculty of Mathematics, Computer Science and Science of Matter  
Department of Computer Science



# Master Thesis

Specialty: Computer Science

## Option

Science and technology of information and communication

# Theme

---

Repérer les mots dans les images de documents

---

**Presented by:** Yaaqob Maatoug

### Jury Members:

N	Full name	Quality
1	Dr. Khaled Halimi	Chairman
2	Dr. Abderrahmane Kefali	Supervisor
3	Dr. Lazhar Farek	Examiner

June 2023.

## ملخص

يوجد عدد كبير من الوثائق حاليًا في مجالات مختلفة مثل الإدارة العامة والصناعة والبحث العلمي والتعليم وغيرها الكثير. أدى النمو المتسارع للوثائق الرقمية إلى زيادة تعقيد إدارتها واستغلالها.

في مواجهة هذه الوفرة من المعلومات النصية ، أصبح من الضروري أن تكون قادرًا على الوصول بسرعة وكفاءة إلى المعرفة الواردة في هذه الوثائق. هذا هو المكان الذي يأتي فيه اكتشاف الكلمات وتحديد الكلمات المهمة وتحديد ما ضمن مجموعات البيانات الضخمة هذه. يلعب اكتشاف الكلمات دورًا مهمًا في مجالات مثل استرجاع المعلومات وتصنيف المستندات والترجمة الآلية والعديد من التطبيقات الأخرى.

في هذا السياق تقع هذه الأطروحة. يهدف عملنا إلى المساهمة في مهمة البحث عن الكلمات في صور المستندات الرقمية ، مع التركيز على المستندات العربية. يتكامل النهج المقترح مع الأساليب التحليلية التي تتطلب تجزئة الوثائق في الكلمات لتنفيذ التعريف. إنه يشمل العديد من خطوات المعالجة التي تهدف إلى تحقيق أهدافنا.

تتكون الخطوة الأولى في نهجنا من المعالجة المسبقة لصور المستندات من أجل تحسين جودتها وتقليل المصنوعات اليدوية. بعد ذلك ، ننتقل إلى تقسيم المستندات إلى أسطر من النص ثم إلى كلمات فردية. بمجرد تقسيم الكلمات ، نقوم باستخراج مجموعة من الميزات من كل كلمة. تلعب هذه المرحلة دورًا رئيسيًا في تمثيل الكلمات والقدرة على تمييزها عن بعضها البعض. استكشفنا عائلات مختلفة من الواصفات من أجل الحصول على تمثيل ثري وتمييزي للكلمات. ثم يتم تجميع الكلمات في فئات بناءً على تشابهها. أخيرًا ، الوحدة الأخيرة في نهجنا هي وحدة البحث ، حيث يعبر المستخدم عن استفساره في شكل صورة من الكلمات ، ويقارنه النظام بالكلمات المستخرجة والمصنفة مسبقًا للعثور على الكلمات الأكثر صلة.

أظهرت التجارب أداءً واعدًا ، وبالتالي فتحت آفاقًا جديدة في مجال تحليل الوثائق والاعتراف بها.

**الكلمات الرئيسية:** صور المستند ، اكتشاف الكلمات ، استخراج الميزات ، المطابقة ، التجميع.

# Résumé

---

Un grand nombre de documents existe actuellement dans divers domaines tels que l'administration publique, l'industrie, la recherche scientifique, l'éducation, et bien d'autres encore. La croissance exponentielle des documents numériques a rendu leur gestion et leur exploitation de plus en plus complexe.

Face à cette abondance d'informations textuelles, il est devenu essentiel de pouvoir accéder rapidement et efficacement aux connaissances contenues dans ces documents. C'est là qu'intervient le repérage de mots (word spotting), qui permet de localiser et d'identifier les mots d'intérêt au sein de ces ensembles de données massifs. Le word spotting joue un rôle crucial dans des domaines tels que la recherche d'informations, la classification de documents, la traduction automatique, et bien d'autres applications encore.

C'est dans ce contexte que s'inscrit le présent mémoire. Notre travail vise à apporter une contribution à la tâche de repérage de mots dans les images de documents numérisés, en mettant l'accent sur les documents arabes. L'approche proposée s'intègre aux méthodes analytiques qui nécessitent la segmentation des documents en mots pour effectuer le repérage. Elle englobe plusieurs étapes de traitement visant à atteindre nos objectifs.

La première étape de notre approche consiste en un prétraitement des images de documents afin d'améliorer leur qualité et de réduire les artefacts. Ensuite, nous procédons à la segmentation des documents en lignes de texte, puis en mots individuels. Une fois les mots segmentés, nous extrayons un ensemble de caractéristiques de chaque mot. Cette étape joue un rôle clé dans la représentation des mots et dans la capacité à les distinguer les uns des autres. Nous avons exploré différentes familles de descripteurs afin d'obtenir une représentation riche et discriminative des mots. Ensuite, les mots sont regroupés dans des classes en se basant sur leur similarité. Finalement, le dernier module de notre approche est le module de recherche, où l'utilisateur exprime sa requête sous forme d'image de mots, et le système la compare avec les mots préalablement extraits et classifiés pour trouver les mots les plus pertinents.

Les expérimentations ont démontré des performances prometteuses, ouvrant ainsi de nouvelles perspectives dans le domaine de l'analyse et de la reconnaissance de documents.

**Mots-clés:** Images de documents, repérage de mots, extraction de caractéristiques, appariement, clustering.

# Abstract

---

A large number of documents currently exist in various fields such as public administration, industry, scientific research, education, and many more. The exponential growth of digital documents has made their management and exploitation increasingly complex.

Faced with this abundance of textual information, it has become essential to be able to quickly and efficiently access the knowledge contained in these documents. This is where word spotting comes in, locating and identifying words of interest within these massive datasets. Word spotting plays a crucial role in areas such as information retrieval, document classification, machine translation, and many other applications.

It is in this context that this thesis falls. Our work aims to make a contribution to the task of finding words in images of digitized documents, with a focus on Arabic documents. The proposed approach integrates with the analytical methods which require the segmentation of the documents in words to carry out the identification. It encompasses several processing steps aimed at achieving our goals.

The first step of our approach consists of a pre-processing of document images in order to improve their quality and reduce artefacts. Next, we proceed to segment the documents into lines of text and then into individual words. Once the words have been segmented, we extract a set of features from each word. This stage plays a key role in the representation of words and the ability to distinguish them from each other. We explored different families of descriptors in order to obtain a rich and discriminative representation of the words. Then the words are grouped into classes based on their similarity. Finally, the last module of our approach is the search module, where the user expresses his query in the form of an image of words, and the system compares it with the previously extracted and classified words to find the most relevant words.

The experiments demonstrated promising performances, thus opening up new perspectives in the field of document analysis and recognition.

**Keywords:** Document images, word spotting, features extraction, matching, clustering

# Table de figures

---

Figure I.1. Organigramme des techniques holistiques. ....	11
Figure I.2. Organigramme des techniques analytiques.....	13
Figure I.3. Architecture générale des systèmes de repérage de mots [03]. ....	14
Figure I.4. Caractéristiques SIFT denses extraites d'une image de mot [21] .....	18
Figure I.5. Histogramme PHOC aux niveaux 1, 2 et 3. L'histogramme PHOC final est la concaténation de ces histogrammes partiels [23] .....	19
Figure I.6. Représentation graphique dans l'approche [30] (a) mot original et points clés détectés sur le squelette (b) le graphe correspondant .....	20
Figure I.7. Lieu caractéristique d'un seul point du mot "page" [32] .....	21
Figure I.8. Modèles de Markov cachés conçus dans [30] : (a) lettre HMM (b) filler HMM (c) mot-clé HMM.....	23
Figure I.9. Un exemple simplifié d'un WG normalisé qui serait obtenu à partir du décodage d'une image de ligne [39].....	24
Figure I.10. L'ensemble des réseaux de neurones BLSTM [40] .....	24
Figure II.1. Les catégories des documents manuscrits [44].....	29
Figure II.2. Etapes de l'analyse de documents [47]. ....	31
Figure II.3. Squelettisation d'une image [58].....	33
Figure II.4. Exemple d'application de RLSA.....	34
Figure II.5. histogrammes de projections verticale et horizontale pour la lettre "a" [75]. ....	37
Figure II.6. Les quatre profils de projection pour la lettre "a" profils haut, bas, gauche et droit [75]. ....	38
Figure III.1. Architecture générale de l'approche proposée.....	42
Figure III.2. Transformation en niveaux de gris. ....	43
Figure III.3. Résultat de binarisation d'une image en niveaux de gris .....	45
Figure III.4. Segmentation du texte en lignes et puis en mots .....	48
Figure III.5. Identification des minimas dans la projection verticale d'une image d'un mot...49	
Figure III.6. Détection de boucles dans une image de mot.....	51
Figure III.7. Ligne de base et zone médiane détectées sur une ligne de texte .....	51
Figure III.8. Détection des hampes et jambages .....	52
Figure III.9. Deux mots ayant le même nombre de boucles, hampes et jambages mais se différencient par le nombre de leurs composantes connexes. ....	53

Figure III.10: Les points terminaux (rouge) et les points de branchements (vert) dans le squelette de l'image. ....	55
Figure IV.1. Interface principale de notre application .....	64
Figure IV.2. Interface de l'Indexation .....	65
Figure IV.3. Exemple d'une image de test. ....	66
Figure IV.4. Chargement d'une image de texte.....	66
Figure IV.5. Affichage de l'image chargée sur le panneau d'affichage.....	67
Figure IV.6. Affichage de l'image au niveau de gris dans un nouvel onglet. ....	67
Figure IV.7. Affichage de l'image binaire dans un nouvel onglet. ....	68
Figure IV.8. Affichage de l'image segmenter dans un nouvel onglet. ....	69
Figure IV.9. Résultat d'extraction des caractéristiques affiché dans un nouvel onglet .....	69
Figure IV.10. Résultats de K-means affichés dans un nouvel onglet.....	70
Figure IV.11. Résultat de classification des mots affiché dans un nouvel onglet.....	71
Figure IV.12. interface de recherche .....	72
Figure IV.13. Chargement d'une requête sous forme d'image de mot .....	72
Figure IV.14. Extraction des caractéristiques de la requête .....	73
Figure IV.15. Affichage d'une image résultante de la recherche .....	74
Figure IV.16. Exemples d'image de la base de la compétition ICFHR 2014 .....	75

# Liste de tableaux

---

Tableau IV.1. Les requêtes utilisées pour interroger la première base ainsi que leurs nombres d'occurrences respectifs.....	76
Tableau IV.2. Les requêtes utilisées pour interroger la deuxième base ainsi que leurs nombres d'occurrences respectifs.....	77
Tableau IV.3. Performances obtenues sur les deux corpus d'images en employant le premier ensemble de caractéristiques.....	78
Tableau IV.4. Performances obtenues sur les deux corpus d'images en employant le deuxième ensemble de caractéristiques.....	78
Tableau IV.5. Performances obtenues sur les deux corpus d'images en combinant toutes les caractéristiques.....	79

# Table des matières

---

ملخص .....	i
Résumé .....	ii
Abstract .....	iii
Table de figures .....	iv
Liste de tableaux .....	vi
Table des matières .....	1
Introduction générale .....	6
Chapitre I. Repérages de mots .....	8
I. 1. Introduction .....	8
I. 2. Qu'est ce que le repérage de mot? .....	8
I.2.1. Définitions .....	8
I.2.2. Utilité du repérage de mot .....	9
I.2.3. Repérage de mots vs reconnaissance de mots .....	9
I. 3. Les aspects du repérage de mot .....	9
I.3.1. Modes de saisi de la requête .....	9
I.3.1.1. Saisie par pointage .....	9
I.3.1.2. Saisie manuscrite directe .....	10
I.3.1.3. Saisie au clavier .....	10
I.3.2. Le moteur de recherche .....	10
I. 4. Les différentes approches de repérage de mot .....	10
I.4.1. Approche holistique .....	11
I.4.1.1. Approche basée sur une requête par exemple (QBE) .....	12
I.4.1.2. Approche basée sur une requête chaîne de caractères (QBS) .....	12
I.4.2. Approche Analytique .....	12
I.4.2.1. Approche basée sur une requête par exemple (QBE) .....	13
I.4.2.2. Approche basée sur une requête chaîne de caractères (QBS) .....	13
I.4.3. Discussion .....	13
I. 5. Architecture générale d'un système de repérage de mots .....	14
I.5.1. Acquisition .....	14
I.5.2. Prétraitements .....	15
I.5.3. Segmentation .....	15
I.5.4. Extraction de caractéristiques .....	15
I.5.5. Recherche .....	15
I. 6. Travaux antérieurs sur le repérage de mots .....	15
I.6.12. Un autre point de vue: Techniques avec et sans apprentissage .....	21

I.6.12.1.	Techniques sans apprentissage.....	22
I.6.12.2.	Techniques utilisant l'apprentissage .....	22
a)	<i>Techniques basées sur l'utilisation des HMMM</i> .....	22
b)	Méthodes basées sur les réseaux de neurones artificiels .....	24
I.6.13.	Critique.....	25
I.6.13.1.	Techniques basées sur l'apprentissage vs techniques sans apprentissage .....	25
I.6.13.2.	Approches basées sur la segmentation vs approches sans segmentation.....	25
I. 7.	Conclusion .....	26
<b>Chapitre II.</b>	<b>Analyse de documents et extraction de caractéristiques</b> .....	<b>27</b>
II.1.	Introduction.....	27
II.2.	Qu'est-ce qu'un document? .....	28
II.2.1.	Définition .....	28
II.1.2.	Document électronique .....	28
II.3.	Catégories de documents manuscrits.....	28
II.3.1.	Documents mono-orientés.....	29
II.3.2.	Documents multi-orientés .....	29
II.3.3.	Documents multi-scripts .....	29
II.3.4.	Documents hétérogènes.....	29
II.4.	Analyse de documents .....	30
II.4.1.	Définition .....	30
II.4.2.	Objectif de l'analyse de documents .....	30
II.5.	Étapes d'analyse de documents.....	30
II.5.1.	Prétraitement.....	31
II.5.1.1.	Lissage.....	31
II.5.1.2.	Binarisation.....	31
II.5.1.3.	Redressement .....	32
II.5.1.4.	La squelettisation .....	33
II.5.2.	Segmentation .....	33
II.5.2.1.	Les méthodes classiques .....	33
II.5.2.3.	Les méthodes de segmentation à base de texture .....	35
II.5.3.	Les niveaux de segmentation.....	35
II.6.	Extraction des caractéristiques (features extraction) .....	36
II.6.1.	Les caractéristiques structurelles .....	37
II.6.2.	Caractéristiques de contour .....	37
II.6.3.	Histogrammes des projections.....	37
II.6.4.	Valeurs ou densités de pixels.....	37
II.6.5.	Moments invariants.....	38
II.6.6.	Descripteurs de Fourier elliptiques .....	38
II.6.7.	Caractéristiques adaptatives .....	38
II.6.8.	Caractéristiques métriques .....	38

II.6.9. Caractéristiques à base du gradient .....	39
II.7. Conclusion .....	39
<b>Chapitre III. Conception</b> .....	<b>41</b>
III. 1. Introduction.....	41
III. 2. Description de l'approche proposée .....	41
III.2.1. Prétraitement d'images .....	43
III.2.1.1. Transformation en niveaux de gris .....	43
III.2.1.2. Binarisation.....	44
III.2.2. Segmentation .....	45
III.2.2.1. Segmentation en lignes.....	45
III.2.2.2. Segmentation en mots .....	47
III.2.3. Extraction de caractéristiques .....	48
III.2.3.1. Le rapport d'aspect .....	49
III.2.3.2. La densité des pixels noirs.....	49
III.2.3.3. Le nombre de minimas dans le profil de projections verticales .....	49
III.2.3.4. Le nombre de boucles.....	50
III.2.3.5. Hampes et Jambages .....	51
a) Détection de la ligne de base.....	51
b) Localisation de la zone médiane .....	52
III.2.3.6. Nombre de composantes connexes .....	52
III.2.3.7. Les moments de Hu.....	53
III.2.3.8. Les points-clés des SIFT.....	54
III.2.3.9. Nombre de points-clés de [03] .....	54
III.2.3.10. Enregistrement des vecteurs de caractéristiques .....	56
III.2.4. Regroupement des mots similaires en utilisant l'algorithme K-means.....	57
III.2.4.1. L'algorithme Kmeans .....	57
III.2.4.2. Résultats du clustering.....	58
III.2.5. La recherche .....	59
III.2.5.1. Saisie du mot et extraction des caractéristiques .....	59
III.2.5.2. Appariement des caractéristiques .....	59
III.2.5.3. Affichage des résultats .....	61
III. 3. Conclusion .....	61
<b>Chapitre IV. Implémentation et résultats</b> .....	<b>62</b>
IV. 1. Introduction .....	62
IV. 2. Outils de développement.....	62
IV.2.1. Langage et environnement de développement .....	62
IV.2.1.1. Java.....	62
IV.2.1.2. Eclipse .....	63
IV.2.2. Bibliothèques utilisées .....	63
IV.2.2.1. Jtessboxeditor.....	63

IV.2.2.2.	Weka .....	63
IV.2.2.3.	OpenCV .....	63
IV.2.2.4.	JFreeChart.....	64
IV. 3.	Présentation de l'application .....	64
IV.3.1.	Interface de "Portail de l'Application" .....	64
IV.3.2.	Interface de l'indexation .....	65
IV.3.2.1.	Chargement d'images.....	66
IV.3.2.2.	Transformation en niveaux de gris .....	67
IV.3.2.3.	Binarisation.....	68
IV.3.2.4.	Segmentation .....	68
IV.3.2.5.	Extraire des caractéristiques .....	69
IV.3.2.6.	Clustering en utilisant K-means .....	70
IV.3.2.7.	Affichage des groupes de mots .....	70
IV.3.3.	Interface de Recherche .....	71
IV.3.3.1.	Charger la requête .....	72
IV.3.3.2.	Extraire les caractéristiques .....	73
IV.3.3.3.	Recherche .....	73
IV. 4.	Expérimentations et résultats.....	74
IV.4.1.	Bases d'images utilisés .....	74
IV.4.1.1.	Base de données AHDB .....	74
IV.4.1.2.	Base de données de la compétition de word spotting ICFHR 2014 .....	74
IV.4.2.	Les mesures d'évaluation utilisées .....	75
IV.4.3.	Expérimentations .....	76
IV.4.3.1.	Expérimentation 1 .....	77
IV.4.3.2.	Expérimentation 2 .....	78
IV.4.3.3.	Expérimentation 3 .....	78
IV. 5.	Conclusion .....	79
<b>Conclusion générale et perspectives:</b> .....		80
Conclusion:.....		80
Perspectives: .....		81
<b>Bibliographie</b> .....		82



# Introduction générale

---

En effet, les documents jouent un rôle essentiel dans notre vie quotidienne. Ils constituent le moyen principal de conservation et de transmission de l'information. Ils permettent de préserver notre héritage culturel, de partager des connaissances scientifiques, de mener des transactions commerciales et bien plus encore.

Cependant, la transition des documents physiques vers un format numérique a été un processus complexe. Cette numérisation de documents existants, en particulier ceux qui remontent à plusieurs siècles, nécessite des technologies avancées pour préserver leur intégrité tout en garantissant une reproduction fidèle de leur contenu.

Grâce aux avancées technologiques récentes dans les domaines de la numérisation et du stockage électronique, de nombreuses bases de données de documents numérisés sont désormais disponibles. Ces bases de données représentent un trésor d'informations potentielles, mais leur pleine utilisation nécessite des outils et des techniques de traitement automatique adaptés. Cela a conduit à un besoin croissant de traiter automatiquement ces documents afin de comprendre leur contenu et de répondre aux demandes des utilisateurs, telles que la recherche de mots-clés, l'indexation, l'identification des auteurs, etc.

Les premiers systèmes développés pour le traitement des images de documents étaient basés sur la reconnaissance optique de caractères (OCR), qui convertit l'image du document en texte. Cependant, les résultats de l'OCR sont souvent médiocres pour les documents anciens ou complexes, tels que les manuscrits anciens, en raison de leur format complexe et de leur détérioration au fil du temps.

Une alternative prometteuse à l'OCR est le repérage des mots (Word Spotting), une technique qui permet de repérer un mot spécifique dans une image donnée par l'utilisateur. Cette approche s'avère particulièrement utile pour les documents comportant des images ou des graphiques complexes.

Ainsi, ce mémoire de fin d'études s'inscrit dans ce contexte. Nous essayons à travers le présent travail de contribuer dans cet axe prometteur d'analyse et reconnaissance de documents, à savoir le word spotting, à travers le développement d'une application permettant le repérage de mots dans des images de documents numérisés arabes.

Ce mémoire s'articule autour de 4 chapitres comme suit:

## ***Chapitre 1. Repérage de mots***

Dans ce premier chapitre, nous nous concentrons sur l'introduction du domaine du repérage des mots. Nous passons en revue les approches couramment utilisées dans la littérature, en décrivant en détail leurs résultats et en fournissant un aperçu complet des études antérieures dans ce domaine. Nous abordons également les différents aspects du repérage des mots, y

compris les moteurs de recherche et les différentes approches, telles que les approches holistiques et analytiques, ainsi qu'une approche multi-échelle.

### ***Chapitre 2. Analyse de documents et extraction de caractéristiques***

Le deuxième chapitre se penche sur l'analyse de documents, un domaine d'étude important dans le traitement d'images. Nous explorons la structure physique des documents, la segmentation en différentes zones homogènes (texte, graphisme, etc.) et les défis associés à cette tâche. Nous discutons également des catégories de documents manuscrits et des étapes d'analyse de documents, y compris l'extraction des caractéristiques.

### ***Chapitre 3. Conception***

Ce chapitre présente la méthodologie mise en œuvre pour notre étude sur le repérage des mots. Nous détaillons les étapes de traitement des images de documents, ainsi que les choix d'algorithmes et de techniques utilisés.

### ***Chapitre 4. Implémentation et résultats***

Le quatrième chapitre se concentre sur la réalisation pratique de notre système de repérage des mots. Ce chapitre met en lumière les ressources, les outils de développement et l'architecture générale de notre système, tout en présentant les interfaces principales de l'application. Il présente finalement les différentes expérimentations effectuées et les résultats obtenus.

# Chapitre I. Repérages de mots

---

## I. 1. Introduction

Dans la vie quotidienne, les documents sont régulièrement utilisés. Ils constituent jusqu'à présent le support principal pour la conservation et la transmission de l'information. Il faudra toujours les évaluer et comprendre leur contenu. En raison de l'indisponibilité des bases de données numériques de grande taille et de haute qualité à partir desquels un processus automatique pourrait être formé, le traitement des documents était auparavant effectué manuellement puisque le processus de conversion et de stockage des images de documents du format papier au format numérique étaient une opération difficile en raison du manque de solutions de numérisation de haute qualité et de dispositifs de stockage de grande capacité.

Récemment les récentes avancées exceptionnelles dans les technologies de numérisation et de stockage électronique ont simplifié ce travail. En conséquence, il existe maintenant plusieurs bases de données de documents numérisés. Cela nécessite un traitement automatisé des documents pour appréhender leur contenu et répondre aux différentes demandes des utilisateurs, comme rechercher certains mots-clés dans le document, localiser l'auteur d'un document ambigu, indexer un document pour en faciliter l'utilisation, etc. Les premiers systèmes proposés pour le traitement d'images de documents (imprimés ou manuscrits), connus sous le nom d'OCR (Optical Character Recognition) convertit l'image du document en texte, mais le problème vient des anciens graphiques (anciens manuscrits). Les résultats de l'OCR ne sont pas bons à cause du format complexe de ces documents et la détérioration qui leur est survenue car ils sont très anciens. Il existe une solution alternative et elle est idéale pour utiliser des images de documents qui s'appelle *repérage des mots* (en anglais : *Word Spotting*). C'est une technique qui repère le mot dans une image donné par l'utilisateur comme requête.

Dans ce chapitre, nous nous intéressons à l'introduction de ce domaine interdisciplinaire et à la présentation des concepts de base qui s'y rapportent. Les différentes approches de repérage de mots dans la littérature sont décrites en profondeur dans ce chapitre, ainsi que certains de leurs résultats. Nous fournissons également un aperçu assez-complet des travaux et techniques relatifs au repérage de mots effectués dans la littérature.

## I. 2. Qu'est ce que le repérage de mot?

### I.2.1. Définitions

Selon [01] le repérage de mot ou le word spotting peut être défini comme "*une tâche qui vise à trouver plusieurs occurrences d'une requête dans des images de documents. La requête est généralement un mot représenté par une image ou fixé par l'utilisateur sous forme ASCII*".

Une autre définition proposé [02] par est la suivante: "*le repérage de mots consiste en un parcours rapide d'un support enregistré afin de localiser un mot et/ou un passage déterminé*".

*Le processus commence à partir du choix du mot (la requête) que nous souhaitons retrouver, ce mot sera proposé ou soumis à un moteur de recherche s'occupant de la fouille des données, lequel nous retournera les positions dans lesquelles se trouve notre requête."*

### **I.2.2. Utilité du repérage de mot**

En effet, le repérage de mot est considéré comme une alternative à l'OCR traditionnel pour différentes applications telles que l'indexation et la recherche dans des collections de documents numérisés.

Le repérage de mot facilite l'indexation et la recherche d'informations tout en analysant les occurrences de requêtes spécifiques dans des documents historiques ou contemporains. Il est très efficace lorsque les documents sont relativement complexes et dégradés et lorsqu'il n'est pas facile de les rétro-convertir en texte.

En outre, le repérage de mots a élargi la portée des logiciels de recherche d'information et a donné une nouvelle direction de recherche aux scientifiques du domaine d'analyse et de reconnaissance de documents.

### **I.2.3. Repérage de mots vs reconnaissance de mots**

En effet, le repérage de mots est totalement différent de la reconnaissance de mots ou de caractères. La différence réside dans la finalité et le résultats attendus de chacune des deux applications. Ainsi, un système de repérage de mots utilise une image de mot-clé comme entrée avec des images de document afin de trouver des occurrences de mot-clé dans des documents sans comprendre leur contenu. Ceci est différent de la tâche de reconnaissance de mots où le contenu de l'image du mot est reconnu

## **I. 3. Les aspects du repérage de mot**

A partir des définitions du repérage de mots précédentes, deux aspects sont à évoquer :

- Le premier consiste en la forme que pourrait prendre la requête ainsi que sa procédure de nomination ou de saisie,
- Le deuxième consiste à l'élaboration du moteur de recherche approprié.

### **I.3.1. Modes de saisie de la requête**

Nous pouvons distinguer trois modes de saisie du mot requête : la saisie par pointage sur l'image, la saisie manuscrite et la saisie par l'intermédiaire du clavier [02].

#### ***I.3.1.1. Saisie par pointage***

Dans cette option de saisie, l'utilisateur choisit l'image du mot cherché (mot requête) à partir de l'image de la page. De ce fait, le mot recherché faisant physiquement partie du support enregistré. Cette approche présente l'avantage que les images du document et du mot demandé sont extrêmement homogènes comme l'image de la requête fait partie de l'image de la page et donc ils ont été écrits par le même scripteur, ont été numérisées sous les mêmes conditions, ont subi les mêmes traitements, etc.

Le principal inconvénient de cette méthode de saisie est que l'utilisateur doit localiser la première instance du mot qu'il cherche, ce qui l'oblige à effectuer un balayage visuel préliminaire du contenu pertinent. Si le mot est fréquent, c'est aisé, au contraire, si le mot est rare (voire absent), il faut prévoir d'autres stratégies.

### ***I.3.1.2. Saisie manuscrite directe***

Pour cette approche, l'utilisateur est équipé d'un périphérique d'entrée. Ce périphérique peut être une tablette graphique, un stylo optique, ou un simple instrument d'écriture sur papier et un scanner.

L'inconvénient de cette méthode de saisi est que l'image du mot requête créée par l'utilisateur est une image étrangère par rapport au document, ce qui nécessitera une homogénéisation entre la requête et l'image du manuscrit avant de procéder l'appariement.

### ***I.3.1.3. Saisie au clavier***

L'utilisateur tape le mot qu'il souhaite rechercher au clavier lorsqu'il utilise cette technique de saisie. Cette méthode est certainement la plus facile à comprendre et à utiliser du point de vue de l'utilisateur. D'autre part, la saisi au clavier nécessite la disponibilité d'une base de données importante contenant des images modèles de lettres sous toutes leurs formes, accompagnées de leur étiquette typographié respective. Lors de la saisi du mot requête, chaque lettre de la requête nous conduira à l'étiquette de la classe manuscrite correspondante; le mot manuscrit sera recréé à partir des combinaisons de lettres. L'image du mot reconstitué sera envoyé vers le moteur de recherche qui parcourra le document afin de retrouver les occurrences possibles du mot requête. Cette approche porte alors le nom de « Word Retrieval ».

## **I.3.2. Le moteur de recherche**

Le moteur de recherche est l'outil le plus difficile à concevoir et reste le même quelle que soit la manière dont la requête est saisie. C'est l'élément essentiel du processus de repérage de mot.

Cependant, les stratégies de recherche appliquées dans les moteurs de recherches se diffèrent entre eux au niveau [02]:

- Des caractéristiques extraites,
- De l'algorithme d'appariement,
- Des permissions de recherche (filtrage des mots vides, éliminations des terminaisons et des conjugaisons, etc...).

## **I. 4. Les différentes approches de repérage de mot**

En effet, les chercheurs ont proposé plusieurs catégorisation des techniques de repérage de mots suivant divers critères [01].

Une des catégorisations sépare les techniques de repérage de mots en deux classes ou approches en fonction des techniques de mise en correspondance des images. La première classe regroupe les techniques d'appariement basées images et la deuxième les techniques basées caractéristiques.

La deuxième catégorisation considère la forme de la requête fournie au moteur de recherche. On trouve ainsi deux approches : approche basée sur une requête chaîne de caractère (QBS pour Query By String), et approche basé sur une requête image exemple (QBE pour Query By Example).

Une autre catégorisation populaire divise les méthodes de repérage de mots soit dans une catégorie basée sur la segmentation (approche analytique), soit dans une catégorie de méthodes sans segmentation (approche holistique).

Ghorbel et al. [01] ont établi une classification plus détaillée en combinant deux catégorisations de la littérature. Cependant, cette classification sépare les techniques de repérage de mots en deux grandes catégories: Analytique (basée sur une segmentation) et Holistique (sans segmentation), et dans chaque catégorie principale, les techniques sont également séparées en deux sous-classes: Les techniques basées QBE et les techniques basées QBS.

Nous survolons dans ce qui suit les différentes approches.

#### I.4.1. Approche holistique

Les techniques holistiques considèrent chaque image de mot comme une unité, et ne nécessitent pas de segmenter les mots en unités plus petites (caractères, graphèmes, etc.). Elles s'appuient principalement sur un processus de segmentation en mots qui doit être réalisée préalablement sur les documents dans lesquels se fait la recherche. Notons que les résultats dépendent de la qualité des mots segmentés dans les images de documents.

Comme nous avons avancé, cette approche peut être subdivisée en deux selon la façon dont la requête est formulée : QBE ou QBS, et chacune de ces deux sous-classes peut être séparée en des sous-sous-classes en fonction d'autres critères.

La figure suivante montre un aperçu des différentes techniques holistiques.

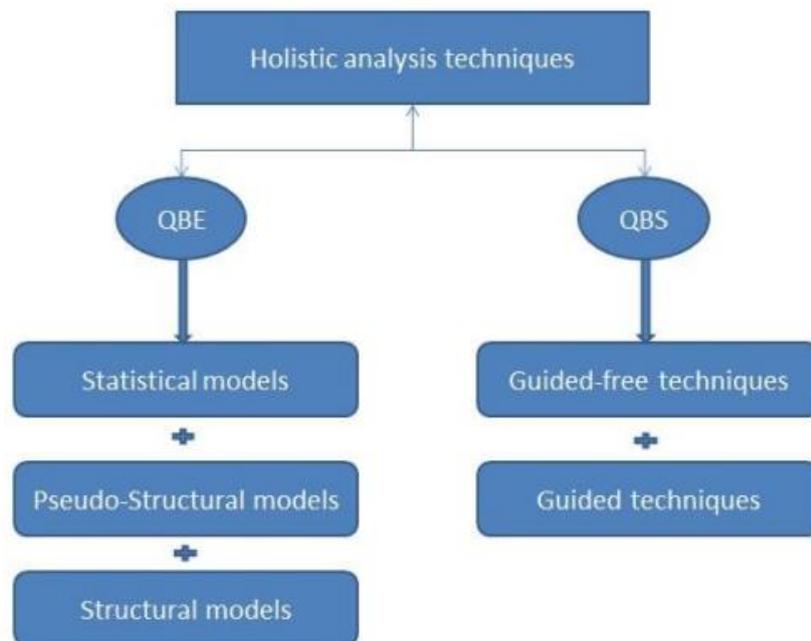


Figure I.1. Organigramme des techniques holistiques.[01]

#### ***I.4.1.1. Approche basée sur une requête par exemple (QBE)***

Les techniques de cette approche considèrent des requêtes sous forme d'images représentant un exemple du mot cherché. Ainsi, dans cette approche, chaque mot ou plus précisément chaque image de mot est représentée à l'aide de l'un des trois types de modèles, classés en modèles statistiques, modèles pseudo-structurels et modèles structurels [01].

#### ***I.4.1.2. Approche basée sur une requête chaîne de caractères (QBS)***

Dans ces techniques, l'utilisateur saisit au clavier le mot qu'il cherche. La requête prend alors la forme d'une chaîne de caractères codée en ASCII ou en un autre système de codage. Ainsi, la plupart des techniques de cette classe permettent à l'utilisateur d'intervenir après une phase automatique. L'intervention de l'utilisateur est en vue de déterminer, parmi une liste d'occurrences qui lui sont proposées, quelles sont les occurrences qui convienne au mot cherché. En fonction des commentaires de l'utilisateur, le système peut améliorer les résultats du repérage.

Certaines techniques de cette classe sont qualifiées comme approches guidées alors que d'autres comme approches non guidées.

### **I.4.2. Approche Analytique**

Dans cette approche, les images de mots et celles de documents sont segmentées en des unités plus petites et plus facilement identifiables.

Une partie des techniques essayent de segmenter les images de mots en caractères. Cependant, la segmentation de mots en caractères, une fois accomplie avec précision, permettra d'améliorer le résultat de repérage mais il n'est pas toujours facile à le faire, surtout avec des documents manuscrits ou dégradés. C'est pour ça, de nombreuses autres techniques étudient diverses hypothèses de segmentation en segmentant excessivement les images en des portions de caractères (traits, graphèmes, etc.).

D'une manière semblable aux techniques holistiques de repérage de mots, ces approches peuvent également être séparées en deux sous-groupes en fonction de la manière dont la requête est formulée: QBE ou QBS.

La figure suivante illustre l'organigramme des techniques d'analyse analytique.

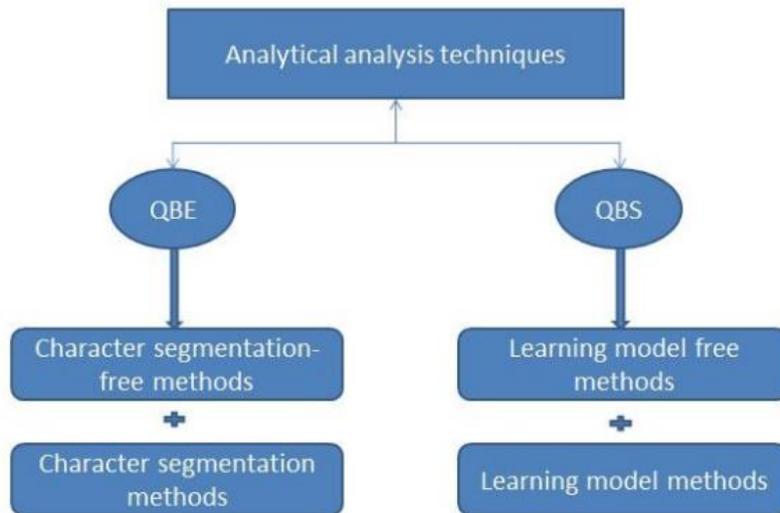


Figure I.2. Organigramme des techniques analytiques.[01]

#### I.4.2.1. Approche basée sur une requête par exemple (QBE)

Les méthodes analytiques basées sur QBE peuvent à leur tour être divisées en deux sous-classes: les techniques de segmentation en caractères, et les techniques de non segmentation en caractères.

La majorité des méthodes de la catégorie basée sur la non-segmentation en caractères utilisent des fenêtres glissantes pour extraire divers attributs, et comparent ensuite les attributs extraits des fenêtres glissantes avec ceux tirés de l'image requête en utilisant des méthodes d'appariement. La plupart des travaux qui s'appuient sur la segmentation en caractères manipulent des documents en chinois ou japonais car la segmentation de l'image de document en caractères est plus facile que dans les autres alphabets [01].

#### I.4.2.2. Approche basée sur une requête chaîne de caractères (QBS)

Les techniques de cette classe utilisent des méthodes de correspondance DTW (Dynamic Time Warping), des modèles HMM et des kNN pour résoudre les problèmes de repérage.

[01] a séparé les méthodes analytiques basées sur une requête chaîne de caractère selon l'approche d'apprentissage procédée en deux sous-classes : les techniques basées sur un apprentissage par modèle et les techniques basées sur un apprentissage sans modèle.

### I.4.3. Discussion

Ghorbel et al. [01] prétendent, à partir de l'étude qu'ils ont menée, que les approches analytiques sont très robustes et donnent de meilleurs résultats que les approches holistiques.

Ils ont également énoncé que décrire l'enquête par une séquence de caractères (l'approche QBS) permet d'utiliser le système dans toutes les situations, même si le mot cherché est absent du document.

Une autre conclusion tirée par les mêmes auteurs est à propos de la phase d'apprentissage. L'inclusion d'une phase d'apprentissage sur une base de documents rend le système de

repérage spécialisé en un type particulier de documents (documents historiques, documents imprimés, documents en langue anglaise, etc.), qui est le type avec lequel il a été entraîné.

### I. 5. Architecture générale d'un système de repérage de mots

Pour atteindre l'objectif visé, tout système de repérage de mots intègre plusieurs étapes de traitements. Ces étapes sont résumées dans la figure I.3. Nous décrivons brièvement ces étapes.

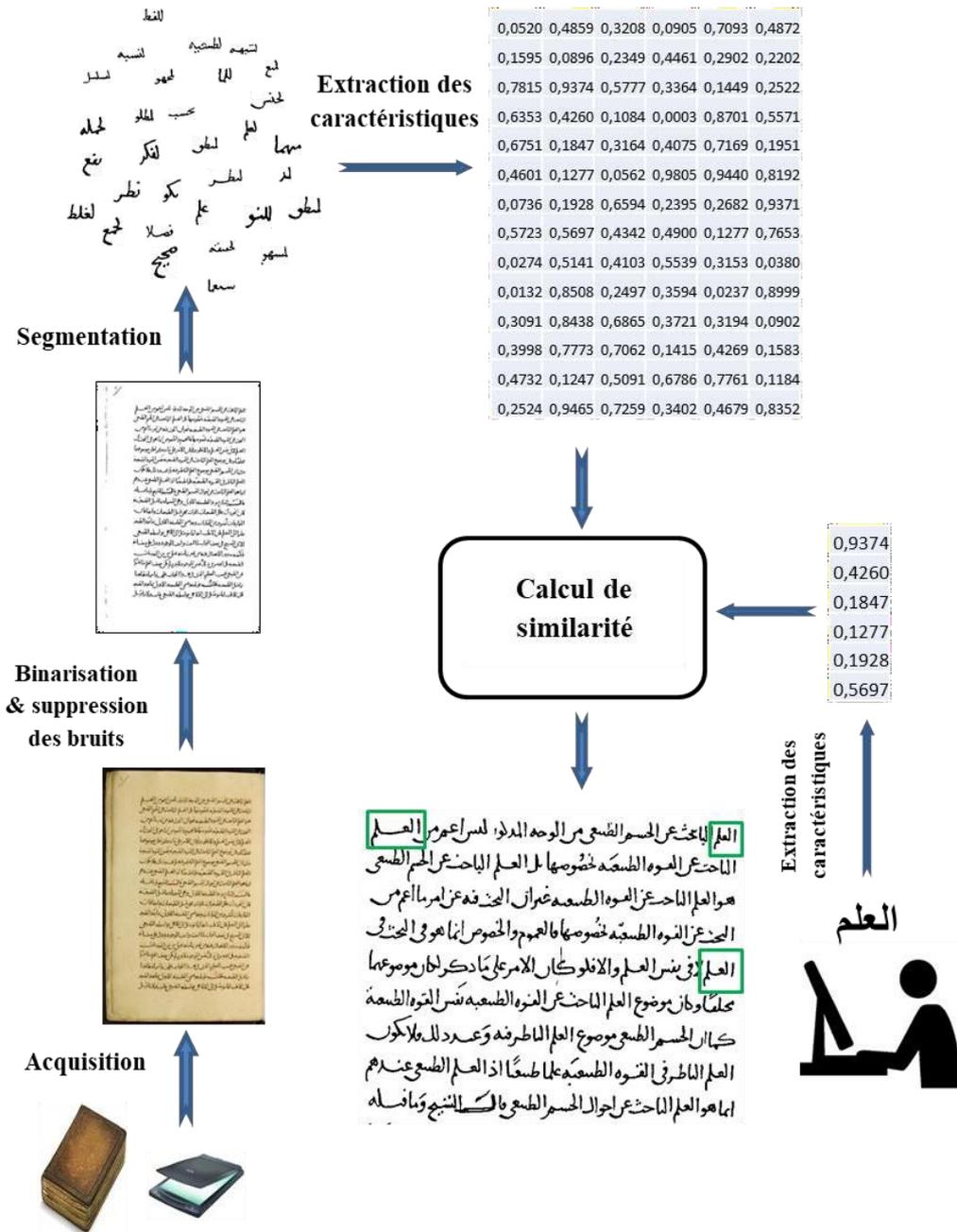


Figure I.3. Architecture générale des systèmes de repérage de mots [03].

#### I.5.1. Acquisition

La première étape est l'acquisition des images de pages en utilisant des appareils de numérisation spécialisés.

### **I.5.2. Prétraitements**

Les images reçues des capteurs obéissent ensuite à un ensemble d'opérations de prétraitement en vue de les nettoyer d'une part et afin de ne garder que les informations pertinentes de l'autre part. Les prétraitements couramment appliqués sont la binarisation, l'élimination de bruit, le filtrage, la détection de contours, la suppression des graphiques, etc.

### **I.5.3. Segmentation**

La troisième étape est la segmentation des images nettoyées, selon l'approche de recherche suivie, en lignes, en mots, en sous-mots, ou en caractères.

La segmentation est en fait est étape cruciale et les résultats finals du système de repérage de mots dépendent fortement de la qualité de la segmentation.

### **I.5.4. Extraction de caractéristiques**

Après, vient l'étape la plus importante qui est l'extraction des caractéristiques à partir des images de pages segmentées et de la requête de l'utilisateur qui peut être, comme nous avons déjà avancé, de différentes formes. Les caractéristiques extraites sont quantifiées par des vecteurs numériques.

Le choix des caractéristiques à extraire est également une tâche cruciale. Les caractéristiques choisies doivent être les plus discriminantes possibles.

### **I.5.5. Recherche**

Finalement, une technique de recherche est appliquée pour repérer le mot cherché (la requête) dans les images du document.

## **I. 6. Travaux antérieurs sur le repérage de mots**

Le repérage de mot constitue une alternative à l'OCR lorsque ce dernier ne soit pas assez robuste pour des documents dégradés ou ceux présentant des écritures manuscrites dans un contexte de vocabulaire ouvert.

En effet, plusieurs travaux ont été effectués dans la littérature sur le repérage de mots concernant divers scripts comme des scripts latins, arabes ou grecs, etc. Ces scripts diffèrent les uns des autres par la nature des alphabets, le nombre de caractères, la direction de l'écriture, la forme et la cursivité entre les lettres. Ils peuvent être soit manuscrits soit imprimés [01].

Comme nous avons avancé, il existe plusieurs catégorisation des approches de word spotting proposée dans la littérature. Nous présentons dans cette section un petit état de l'art sur les principaux travaux de repérage de mots effectués dans la littérature.

### **I.6.1. Travaux de Manmatha et son équipe**

L'équipe de Manmatha à l'Université du Massachusetts à Amherst a été pionnière dans le développement des méthodes holistiques de repérage de mots. Les recherches de cette équipe conduisent au développement d'un des premiers moteurs de recherche basé sur le repérage de mots pour identifier les mots. Cependant, à cause de leurs importances, les travaux de

Manmatha et son équipe sur la collection numérisée en niveaux de gris de George Washington sont toujours cités [04], [05], et [06] .

Ils ont utilisé des caractéristiques de profil multidimensionnelles telles que des transitions fond/encre, des profils de mots et des profils de projection. L'appariement était faite en utilisant l'algorithme DTW (Dynamic Time Warping), qui s'agit d'une technique de correspondance qui compare les attributs tirés des images de mots.

Ils ont fusionné les caractéristiques scalaires et les caractéristiques de profil pour la reconnaissance holistique des mots dans certains de leurs autres travaux plus récents [07], et ils utilisent des chaînes de Markov cachées HMM pour la correspondance (Modèles de Markov cachés). Pour une reconnaissance complète des mots, ils ont également opposé une stratégie SVM (Support Vector Machine) à un modèle Naive Bayes couplé à des indices de densité gaussiennes [08].

### **I.6.2. Méthode de Wshah et al.**

Plus récemment, une méthode de repérage de mots indépendante du script pour les documents manuscrits a été proposée par Wshah et al. [09], dans laquelle les caractéristiques de gradient et d'intensité sont dérivées d'une fenêtre glissante. Le rapport des pixels noir sur blanc sert d'indicateur d'intensité et les orientations du gradient dans huit directions différentes sont agrégées dans un histogramme.

Khayat et al. dans [10] utilisent également des caractéristiques de gradient. Pour créer les images en niveaux de gris nécessaires au calcul du gradient, un filtre moyen est appliqué aux images binaires. Le gradient est initialement quantifié dans 32 directions à l'aide de blocs de  $9 \times 9$ . Un vecteur de caractéristiques à 400 dimensions est créé après réduction de la dimensionnalité. Ces méthodes reposent sur l'apprentissage et nécessitent des données correctement étiquetées, ce qui est une opération laborieuse.

### **I.6.3. Travail de Cheriet et Moghaddam**

Cheriet et Moghaddam [11] ont extraits des caractéristiques topologiques et géométriques à partir des squelettes des composantes connexes (CC) de l'image. En utilisant des points d'extrémité, et des points de branchement, la forme du script est examinée dans le but d'extraire des descripteurs. Pour classer les CC dans le document cible en fonction de leur ressemblance avec la requête CC, une fonction de distance est proposée.

### **I.6.4. Système de Sari et Kefali**

Les caractéristiques structurelles, tels que les boucles, les ascendants et les descendants, ont été utilisés par Sari et Kefali [12] dans leur système de recherche dans les anciens documents arabes. Le document a été divisé en composantes connexes, puis il a subi une procédure de codage se terminant par la représentation du document sous forme de fichier ASCII. Le repérage du mot requête est ensuite traduit par une recherche de texte. Cette technique fonctionne bien avec les documents imprimés et avec d'autres documents de haute qualité.

### **I.6.5. Recherches de Terasawa et al.**

Les recherches de Terasawa et al. sur les anciens manuscrits japonais ont inclus des tentatives de "Word Spotting" basées sur l'espace propre (Eigen Space) [13] et [14]. Des fenêtres glissantes ont été utilisées pour extraire les signatures. Des calculs ont été effectués pour déterminer les niveaux relatifs du gradient dans les 8 directions principales. Les caractéristiques invariantes à l'échelle en sont le résultat.

En utilisant un algorithme d'appariement basé sur le DTW, les écarts morphologiques entre les mots sont surmontés. Dans [15], le DTW est également utilisé pour la correspondance des mots. [16] décrit une autre méthode sans segmentation basée sur les modèles de Markov cachés et d'autres modèles statistiques.

### **I.6.6. Travaux de Leydier et al.**

Le laboratoire LIRIS (Laboratoire d'Informatique en Image et Systèmes d'information) de Lyon fait depuis plusieurs années des recherches sur le Word Spotting en tentant de lui donner un statut, une alternative à l'OCR, un moteur de recherche ; il fournit maintenant plusieurs chemins. Leydier et al. [17] [18] ont extrait et identifié les mots sélectionnés à partir de textes latins médiévaux numérisés en niveaux de gris dans le cadre de leur enquête. Ils ont créé des fonctionnalités basées sur l'orientation du gradient pour y parvenir. Ces caractéristiques sont stables vis-à-vis des changements d'échelle et invariantes aux altérations géométriques. Une recherche est effectuée sur les nombreuses pages du document pour découvrir toutes les instances du terme de recherche après qu'il a été représenté par un vecteur de caractéristiques. Le balayage de surface est utilisé par l'algorithme d'appariement pour déterminer le chevauchement approprié entre les caractéristiques. Cependant, trois stratégies d'appariement ont été expérimentées dans ces travaux.

La première est une correspondance naïve et rigide dans laquelle l'image du mot requête scanne la page de texte, chaque pixel étant considéré comme une zone d'intérêt et la comparaison se fait pixel par pixel. Cette méthode d'appariement ne tolère aucune variation spatiale et nécessite une parfaite superposition de l'image du mot requête et celle du texte à rechercher.

La deuxième méthode, plus précise que la première mais très complexe, utilise un algorithme de correspondance élastique dans lequel un pixel de l'image de mot est comparé au gradient le plus proche de son voisinage dans l'image de texte.

La troisième méthode cherche l'information la plus représentative, que l'on peut trouver dans les lignes verticales des manuscrits latins. Il est plus acceptable d'examiner les pixels autour de ces traits plutôt que les formes complètes car la distance entre ces traits peut changer, ce qui peut entraîner un resserrement ou un espacement entre les lettres qui peut varier d'une occurrence à l'autre. Pour calculer les guides, la morphologie mathématique est appliquée aux niveaux de gris des images. Pour obtenir les zones d'intérêt, une ouverture avec un élément structurant vertical est réalisée et les boîtes définissant les guides sont agrandies.

Le mot requête est ensuite fragmenté suivant les zones d'intérêt, et la recherche de correspondance est effectuée par fragment. En utilisant cette méthode, ils ont pu retrouver toutes les occurrences de mots requêtes sur les documents Latins. Ils ont même testé cette

méthode sur la base utilisée par Rath et al. [19], et ils ont obtenu un taux de récupération de 85.18% (23 occurrences retrouvées parmi 27) alors que celui obtenu par Rath et al. était de 77.64% (21 occurrences retrouvées parmi 27).

### **I.6.7. Méthode de Ghilas**

Ghilas et al. [03] ont proposé une méthode efficace d'extraction de caractéristiques pour capturer les caractéristiques complexes de l'écriture manuscrite arabe et tolérer le bruit et la dégradation. La méthode d'extraction de caractéristiques proposée est basée sur des points-clés. Tout d'abord, un ensemble de points-clés est extrait à l'aide du squelette de l'image. Deuxièmement, un vecteur de caractéristiques est extrait à partir de chaque point clé dans l'image binarisée. Une fonction de distance est proposée pour faire correspondre deux séquences de vecteurs de caractéristiques extraits des CC.

### **I.6.8. Travail de Rodrguez**

La solution de repérage de mots sans contrainte décrite dans [20] utilise une méthode plus simple mais plus dense pour reformuler les gradients de tous les pixels de l'image de mot par rapport à la technique de Leydier. Pour créer un vecteur de caractéristiques, une fenêtre glissante centrée sur chaque colonne de l'image de mots est utilisée. La fenêtre glissante est en outre segmentée en cellules rectangulaires en utilisant trois méthodes distinctes à chaque point. La représentation finale est constituée des vecteurs caractéristiques de chaque fenêtre glissante inclus dans la même image de mot. Les résultats des expérimentations présentées dans [20] indiquent que cette stratégie fonctionne mieux que les autres approches traditionnelles. Cependant, cela se fait au détriment d'un coût de calcul beaucoup plus élevé causé par les vecteurs de caractéristiques à 128 dimensions.

### **I.6.9. Recherches M. Rusiñol**

Plus tard, le défi de repérage de mots a été modifié par M. Rusiñol et son équipe [21] pour inclure la méthode du sac de mots visuels. Ils commencent par supprimer les descripteurs SIFT (Scale-Invariant Feature Transform) des images de document, puis décrivent les correctifs à l'aide des SIFT classés. Ils utilisent également la technique de l'indexation sémantique latente pour affiner la représentation, ce qui permet d'obtenir des patches même s'ils n'ont pas exactement les mêmes caractéristiques que la sous-image de la requête. Les résultats des tests effectués par [21] sont extrêmement encourageants. La taille de la fenêtre, cependant, ne peut pas être trivialement ajustée à la longueur de la requête, ce qui est un inconvénient. Les auteurs précisent que la longueur de la requête par rapport à la taille fixe de la fenêtre a un impact significatif sur les performances de la méthode. De plus, le problème de calcul coûteux existe toujours dans cet algorithme, car il calcule de manière dense les descripteurs SIFT sur toute la page (voir la figure I.4. )

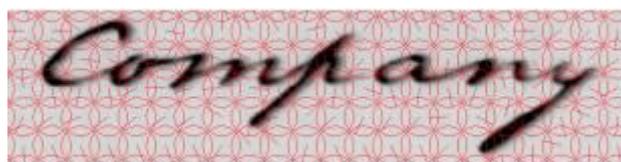


Figure I.4. Caractéristiques SIFT denses extraites d'une image de mot [21]

### I.6.10. Travaux de Almazan et al.

L'analyse d'images de documents a récemment bénéficié de l'introduction d'approches de plus en plus sophistiquées développées pour les images naturelles. J. Almazán et al. [22] implémentent la détection de mots dépendant du scripteur sans segmentation en utilisant un SVM basé sur un apprentissage non supervisé de la requête décrite par les HOG (Histogram of Oriented Gradient) quantifiés. Cependant, le problème de la taille de fenêtre fixe qui existe dans [21] est résolu par l'utilisation du descripteur HOG. En adaptant l'apprentissage non supervisé à la requête spécifique avec moins de restrictions sur les échantillons d'apprentissage, cela améliore également l'approche de détection de mots. Ces deux faits rendent cette approche faisable et flexible. Le problème que la méthode de [22] ne résout pas est de savoir comment indexer la requête lorsque plusieurs scripteurs sont impliqués.

Peu de temps après, les mêmes auteurs ont créé une stratégie basée sur les attributs pour faire face au problème de repérage de mots multi-scripteurs [23]. L'écart entre les techniques de détection de mots requête par exemple et requête par chaîne est également comblé par cette étude. Cette méthode, qui s'inspire du sac de noyaux de chaînes de caractères de l'apprentissage automatique, encode les chaînes de mots sous la forme d'un histogramme pyramidal de caractères, ou PHOC. Le mot "beyond" est construit dans le style PHOC, comme le montre la figure I.5. Au lieu d'utiliser le cadre BoVW pour la représentation d'images de mots, le cadre Fisher Vector basé sur les caractéristiques SIFT est utilisé. De plus, pour saisir la structure du mot image, une pyramide spatiale de  $2 \times 6$  est mise en place, conduisant à un descripteur final d'environ 25 000 dimensions. La partie la plus innovante de ce travail est le modèle de représentation basé sur les attributs proposés, qui réalise une représentation unifiée des images de mots et des chaînes.



Figure I.5. Histogramme PHOC aux niveaux 1, 2 et 3. L'histogramme PHOC final est la concaténation de ces histogrammes partiels [23]

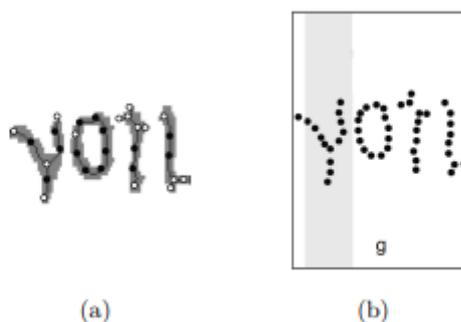
### I.6.11. Travaux reposant sur des représentations graphiques

Le modèle basé sur la structure est encore moins populaire que celui basé sur l'apparence pour l'analyse de l'écriture manuscrite. Parce que les représentations graphiques sont particulièrement efficaces pour capturer et décrire les caractéristiques structurelles des objets, elles sont utilisées dans certaines œuvres existantes. Il n'y a eu que quelques tentatives

d'utilisation de graphes ou de concepts qui leur sont liés dans la recherche sur la récupération et la reconnaissance de l'écriture manuscrite, malgré le fait que la représentation graphique est assez courante dans la reconnaissance de composés chimiques en chimie et la reconnaissance de symboles dans l'analyse de documents.

Les premières études utilisant des représentations graphiques pour les chiffres solitaires ont été publiées dans la littérature [24, 25], ainsi que dans le contexte de l'identification d'un seul caractère (26, 27, 28, 29). Bien que le modèle hiérarchique proposé dans [26] capture la nature hiérarchique des caractères manuscrits - en particulier les caractères chinois, qui ont une structure plus complexe que les lettres latines - un mécanisme de représentation plus sophistiqué est toujours nécessaire pour distinguer les caractères similaires en général.

Plus tard, en utilisant le mot squelette d'image, qui ne contient que des sommets et pas d'arêtes comme dans l'exemple illustré à la Figure I.6, Fischer et al.[30] créé un modèle de représentation graphique pour la reconnaissance de l'écriture manuscrite. Les informations structurelles sont toujours conservées et la représentation graphique résiste à la mauvaise qualité du squelette en ajoutant suffisamment de points d'intersection entre les points clés. Cependant, cela entraîne également la possibilité que les réseaux incluent un nombre excessif de sommets, ce qui augmente le calcul.



Figurier I.6. Représentation graphique dans l'approche [30] (a) mot original et points clés détectés sur le squelette (b) le graphe correspondant

Dernièrement, la conversion de graphes en vecteurs de caractéristiques a gagné en popularité en raison de la complexité de calcul coûteuse de l'appariement de graphes. Dans [31], le squelette est initialement disséqué pour récupérer les informations topologiques et géométriques des formes de sous-mots. Chaque sous-mot reçoit alors un graphe orienté acyclique, qui est transformé en un vecteur de signature topologique tout en conservant la structure du graphe. Une autre méthode de détection de mots basée sur la structure qui s'inspire des traits distinctifs de Loci est le travail de D. Fernández et al. [32]. Pour chaque point clé, les comptages d'intersection le long de huit directions sont déterminés et quantifiés, comme illustré à la Figure I.7. L'image squelette est utilisée pour extraire la fonctionnalité Loci mise à jour, et les pixels d'arrière-plan ou les pixels de premier plan peuvent servir de points clés. La dimension de l'espace des caractéristiques est de  $3^8$  (=6561) puisque les caractéristiques des lieux sont représentées en base 3. La représentation d'une image de mot est un histogramme de nombres de lieux si nous traitons la caractéristique décimale des lieux

à 4 dimensions comme des nombres de lieux. L'utilisation d'une table de hachage pour stocker les signatures condensées des images présente l'avantage de rendre l'indexation des mots rapide et simple, ce qui permet d'économiser du temps et de l'espace.

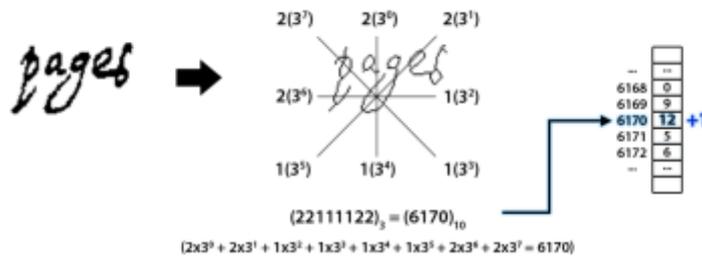


Figure I.7. Lieu caractéristique d'un seul point du mot "page" [32]

De plus, l'approche de sérialisation de graphes est appliquée à l'appariement de graphes pour la détection de mots manuscrits par Lladós et ses collègues [33]. Bag-of-pathways (BoP), un descripteur unidimensionnel créé en extrayant et en regroupant les chemins acycliques des graphes, est utilisé pour décrire les mots. Bien que cette tentative soit intrigante, les performances ne sont pas du tout satisfaisantes.

Dans le cadre de l'application de repérage de mots dans des documents historiques, J. Lladós et al. dans [33] ont mené une comparaison et une étude de deux modèles de représentation de l'écriture manuscrite basés sur la structure et de deux modèles basés sur l'apparence. Il s'agit d'œuvres de D. Fernández [32], Dutta [34], Rath [19] et Rusiñol [21], respectivement. Selon [33], l'approche basée sur l'apparence [21] obtient les meilleures performances, tandis que les techniques structurales, en particulier l'approche basée sur les sacs de chemins, ne sont pas aussi performantes que les méthodes basées sur l'apparence. En effet, pour créer un graphe représentant les mots, il faut une étape préalable de transformation de l'image raster en image vectorielle. Lorsque l'on travaille avec des documents anciens, cette méthode de vectorisation est très sensible à de modestes dégradations. Les méthodes basées sur la structure et les méthodes basées sur l'apparence, cependant, sont au coude à coude si nous prenons en compte les développements récents dans les approches de détection de mots basées sur la structure et incluons ces études [30] dans la comparaison. Chacun d'eux présente des avantages et des inconvénients particuliers. En général, les approches basées sur l'apparence nécessitent beaucoup d'espace mémoire et ont un haut niveau de complexité temporelle, ce qui les rend inadéquates lorsque le temps et l'espace sont rares. D'autre part, la binarisation, qui est utilisée comme phase préparatoire, présente un défi à la plupart des approches structurales. La perte d'informations après la binarisation peut entraîner une réduction significative des performances en fonction de l'état des images du document. En ce sens, les approches basées sur l'apparence qui ne nécessitent pas de binarisation et extraient l'information directement des images en niveaux de gris sont moins contraintes.

### I.6.12. Un autre point de vue: Techniques avec et sans apprentissage

Sous un angle différent, les méthodes utilisées pour résoudre le problème de reconnaissance de mots manuscrits peuvent généralement être classées en deux classes en fonction de l'utilisation de l'apprentissage. Les approches basées sur l'apprentissage [30, 21, 22, 23]

utilisent des techniques d'apprentissage automatique pour former des modèles des caractères ou des mots recherchés, alors que l'alternative est qualifiée comme des techniques sans apprentissage) [05, 20, 17, 32].

#### ***1.6.12.1. Techniques sans apprentissage***

La technique d'indexation sémantique latente (LSI : Latent semantic indexing) [35] est utilisée dans la recherche de [21] pour récupérer les correctifs pertinents même s'ils n'ont pas exactement les mêmes caractéristiques. LSI a été initialement créé pour le travail de récupération de texte. Son idée de base est de trouver des modèles dans les relations entre les phrases et les concepts présents dans une collection non structurée de textes en utilisant une technique mathématique appelée "décomposition en valeurs singulières (SVD : singular value decomposition)". Le scénario de M. Rusiñol et alwork. [21] fait l'hypothèse que les patches qui surviennent dans des situations similaires sont susceptibles d'avoir des significations connexes. En d'autres termes, l'espace des descripteurs a une certaine structure sémantique sous-jacente. En conséquence, ils transforment d'abord toutes les fonctionnalités de patch en un espace où les patches avec des sujets similaires mais des descripteurs différents sont proches et calculent la distance cosinus comme mesure de similarité dans cet espace plutôt que de comparer directement les fonctionnalités de patch après les avoir obtenues avec le modèle BoVW et Stratégie SPM (spatial pyramid matching). Les mots sont modélisés en relation avec les sujets de chaque document individuel de cette manière. Une idée similaire est également utilisée pour résoudre le problème de repérage de mots requête par chaîne [36]. Un extrait de mot a deux représentations : une représentation textuelle et une représentation visuelle. Ces deux représentations sont combinées en un seul espace en utilisant l'analyse sémantique latente (LSA). Chacun des sujets abstraits suggérés représente une distribution représentative des caractéristiques visuelles et texturales. Le descripteur textural et le descripteur visuel sont combinés en une seule description pour fournir la matrice de transformation. Une matrice de descripteur par mot est créée en utilisant tous les mots de l'ensemble d'apprentissage, et le SVD la décompose pour produire la matrice de transformation. Dans la zone où la matrice de transformation se projette, la distance de similarité finale est déterminée. En conclusion, l'emploi de LSA permet le repérage de mots requête par chaîne des langues latines. Les résultats expérimentaux présentés dans [36] illustrent sa force par rapport à d'autres approches de pointe. Cependant, nous pouvons également constater que les performances de cette approche diminuent beaucoup lorsqu'elle fait face à une requête qui n'apparaît pas dans l'ensemble d'apprentissage.

#### ***1.6.12.2. Techniques utilisant l'apprentissage***

##### ***a) Techniques basées sur l'utilisation des HMMM***

Les HMM, ou modèles de Markov cachés, sont une méthode éprouvée et fiable fréquemment utilisée pour la reconnaissance de l'écriture manuscrite. Il sert de base à de nombreux efforts sur la reconnaissance de mots manuscrits [30, 37, 38]. Par exemple, trois HMM distincts sont créés (Figure I.8) dans [30] pour récupérer des mots-clés dans une situation impliquant plusieurs scripteurs d'écriture manuscrite. La représentation de chaque ligne de texte est une série de modèles de lettres. Chaque ligne de texte reçoit un score basé sur le rapport de vraisemblance entre un modèle de mot-clé et un modèle de remplissage. La ligne de texte est

classée comme une correspondance positive ou une correspondance négative à un seuil spécifique.

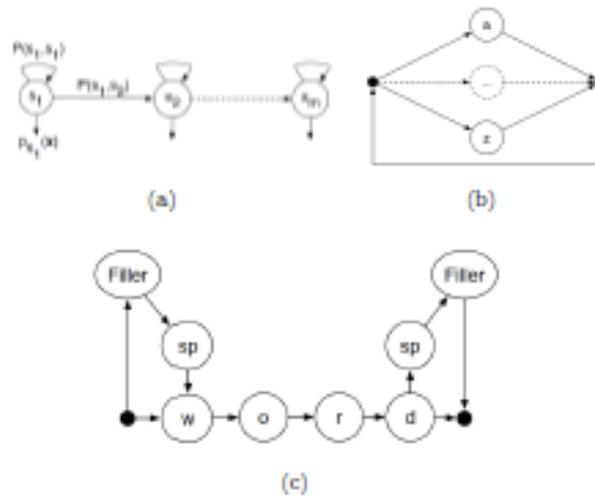


Figure 1.8. Modèles de Markov cachés conçus dans [30] : (a) lettre HMM (b) filler HMM (c) mot-clé HMM

Le travail de [39] est un autre exemple d'utilisation de HMM pour le repérage de mots manuscrits. Dans cette étude, les caractéristiques morphologiques séquentielles des lignes de texte sont extraites et un modèle HMM est utilisé pour prédire les mots en travaillant avec un modèle de langage N-gramme et en estimant la probabilité des caractères. Le graphe de mots (WG) est modifié pour exprimer des segmentations de mots alternatives, des probabilités de décodage de mots et des hypothèses de séquence de mots (voir Figure I.9). Contrairement à d'autres stratégies d'apprentissage, HMM nécessite un nombre important d'échantillons d'apprentissage afin d'estimer les modèles HMM, en particulier dans le cas de nombreux styles d'écriture.

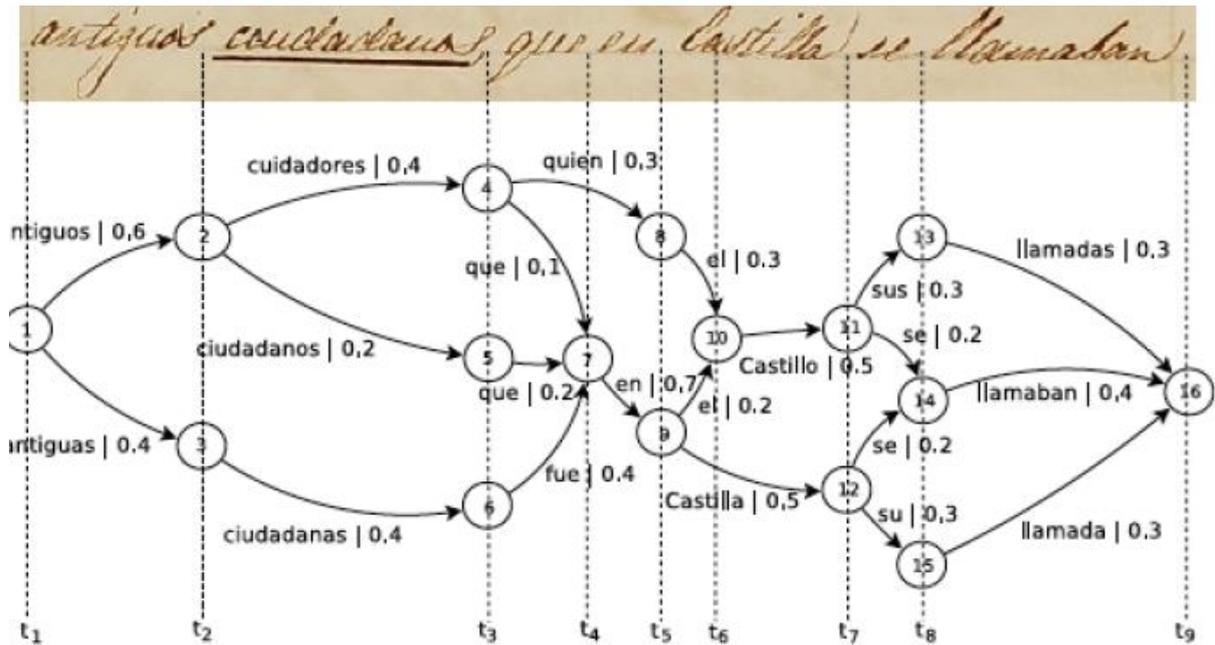


Figure I.9. Un exemple simplifié d'un WG normalisé qui serait obtenu à partir du décodage d'une image de ligne [39]

**b) Méthodes basées sur les réseaux de neurones artificiels**

Dans le domaine de l'apprentissage automatique, les réseaux de neurones ont gagné en popularité en tant que méthode d'apprentissage. Le repérage de mots manuscrits n'est qu'une des nombreuses applications de vision par ordinateur qui l'ont adopté en raison de sa polyvalence. Le réseau neuronal bidirectionnel à mémoire longue et courte durée (BLSTM), un réseau neuronal récurrent récemment créé, a été utilisé comme outil de reconnaissance dans les travaux de [40]. Chaque lettre et chaque place dans la ligne de texte reçoivent un ensemble de probabilité. La Figure I.10 montre ce réseau dans une représentation visuelle.

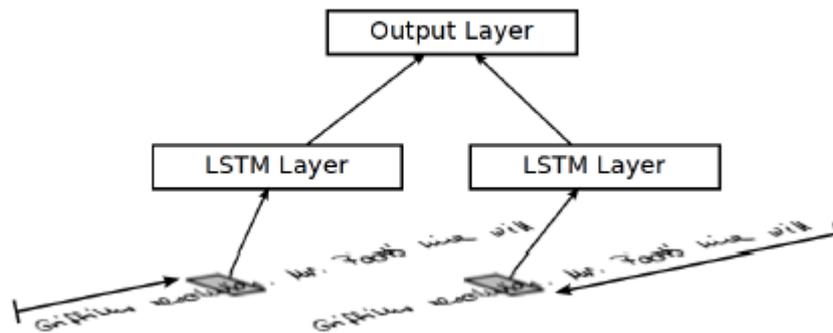


Figure I.10. L'ensemble des réseaux de neurones BLSTM [40]

**c) Autres techniques**

Les approches d'apprentissage mentionnées ci-dessus ne parviennent souvent pas à répondre à de nouvelles questions qui ne font pas partie de l'ensemble de données de formation. J. Almazán et al. [22, 23] utilisent la technique d'apprentissage exemplaire pour les images de

mots de requête pour résoudre ce problème en mettant en évidence les zones des images qui sont pertinentes pour la requête. Une nouvelle représentation de chaque requête est créée à l'aide d'un exemple de framework SVM [41]. La nouvelle représentation peut être considérée comme donnant plus de poids aux dimensions des descripteurs de région qui se rapportent à la requête.

### **1.6.13. Critique**

#### **1.6.13.1. Techniques basées sur l'apprentissage vs techniques sans apprentissage**

En un mot, il y a des avantages et des inconvénients aux approches d'apprentissage et sans apprentissage. Bien que les approches basées sur des exemples soient largement utilisées dans la littérature, leur efficacité dans des situations avec plusieurs scripteurs est faible. Cependant, du point de vue des applications, de telles techniques sont insuffisantes si l'utilisateur souhaite simplement mettre le terme au clavier et rechercher l'existence de ce mot.

Néanmoins, par rapport aux approches basées sur l'apprentissage, les approches basées sur des exemples ne nécessitent pas le processus de formation, qui peut être effectué même avec une très petite quantité d'images et sans vérité terrain. Étant donné que la majorité des méthodologies basées sur l'apprentissage reposent sur l'extraction ou l'estimation de caractères [30, 36, 22, 39], cette extraction de caractères est effectuée manuellement [30, 39] ou en utilisant une stratégie de clustering [23]. En ce sens, l'absence d'échantillons d'apprentissage affecte l'efficacité de ces techniques lorsque certains caractères particuliers ou N grammes n'apparaissent que sporadiquement dans la collection. De plus, l'approche basée sur l'apprentissage ne peut pas être utilisée si les mots ne sont pas annotés. De plus, parce que les langages de glyphes ont tellement de caractères, il est difficile de leur appliquer le cadre de ce type d'approche dans un cadre sans restriction. Par exemple, il faut un gros effort pour créer les prototypes des caractères chinois, puisqu'il y a environ 50 000 caractères au total.

#### **1.6.13.2. Approches basées sur la segmentation vs approches sans segmentation**

Deux catégories principales d'algorithmes de repérage de mots ont été développées au fur et à mesure que la recherche sur le repérage de mots progresse. L'une est l'approche basée sur la segmentation, qui a été l'une des premières approches du problème de repérage des mots et reposait sur une étape préliminaire d'analyse de la mise en page consacrée à la segmentation des mots. Il n'existe actuellement aucune technique efficace de segmentation des mots car il existe encore plusieurs problèmes liés à la division d'une ligne en mots distincts. En conséquence, des techniques de repérage de mots sans segmentation sont présentées. L'approche sans segmentation tente d'effectuer simultanément la segmentation et le repérage. L'image de la page entière sert d'entrée plutôt qu'une image de mot candidat.

Les algorithmes de repérage de mots incluent fréquemment une technique pour décomposer les lignes en mots. Ces mots sont codés avec des signatures de forme après segmentation afin qu'ils puissent être traités comme des signaux unidimensionnels. Rath et Manmatha ont publié les premières recherches sur le repérage de mots basé sur la segmentation dans [05]. L'image du mot de requête est finalement comparée à l'ensemble du corpus de mots à l'aide d'une distance basée sur la déformation temporelle dynamique (DTW). Le fait que cette stratégie dépende de manière significative de la segmentation précise des mots est l'un de ses

inconvenients. Ses performances sont limitées par sa capacité à tolérer les erreurs de segmentation. La segmentation précise des mots est également cruciale pour la méthode utilisée par D. Fernández et al. dans leur article de 2011, qui est basé sur des caractéristiques de lieux caractéristiques enregistrées dans une structure de hachage.

En ce qui concerne les performances, les méthodes de segmentation jouent un rôle important dans les approches de repérage de mots basées sur la segmentation. Cependant, il n'existe aucune méthode de segmentation existante qui puisse atteindre une précision de 100 % pour tous les types d'écriture manuscrite. Dans les approches de répétition de mots basées sur la segmentation, les méthodes de segmentation ont un rôle important en termes de performances. Cependant, aucune méthode de segmentation n'est actuellement disponible qui puisse atteindre une précision de 100 % pour t

outes les variétés d'écriture manuscrite. Une méthode de segmentation des lignes est proposée dans la thèse de doctorat de V. Malleron [41]. Après avoir séparé les composants liés, ils calculent un éventail de voisinage pour chaque composant qui contient la distance euclidienne, l'orientation et les voisins les plus proches dans 18 directions de l'espace afin de comprendre la structure de voisinage du composant. Les composants connectés adjacents sont assemblés pour construire la ligne de texte à l'aide de l'analyse de l'extraction des coins et des bordures et de l'estimation de l'orientation de la ligne.

## **I. 7. Conclusion**

Dans ce chapitre, nous avons introduit le domaine du repérage de mots dans les images documents. Nous avons souligné l'importance croissante du traitement automatisé des documents à l'ère de la numérisation et du stockage électronique. Alors que les anciens systèmes OCR ont des limites pour traiter les documents anciens ou complexes, le repérage de mots offre une solution alternative en permettant aux utilisateurs de localiser des mots spécifiques dans une image de document.

Nous avons défini le repérage de mots comme une technique qui repère un mot donné par l'utilisateur dans une image de document, offrant ainsi une approche plus flexible pour l'extraction d'informations pertinentes. Nous avons discuté de l'utilité du repérage de mots et souligné les différences entre le repérage de mots et la reconnaissance de mots.

Ensuite, nous avons exploré les différents aspects du repérage de mots, en mettant l'accent sur le moteur de recherche et les approches holistiques et analytiques.

Enfin, nous avons examiné l'architecture générale d'un système de repérage de mots et passé en revue les travaux antérieurs réalisés dans ce domaine. Ce chapitre fournit une base solide pour comprendre les concepts clés et les approches utilisées dans le repérage de mots, ce qui sera utile pour la suite de notre étude.

# Chapitre II. Analyse de documents et extraction de caractéristiques

---

## II.1. Introduction

Il y a actuellement beaucoup de papiers stockés dans les bibliothèques, les musées, les organisations, etc. Outre l'énorme volume de documents créés quotidiennement dans les entreprises, les bureaux de poste, les banques et autres institutions, cette énorme quantité de documents stockés met en évidence de nouveaux besoins, notamment pour leur préservation numérique et l'utilisation la plus efficace possible sans la nécessité d'une saisie manuelle.

Un domaine d'étude important dans le domaine du traitement d'images, à savoir l'*analyse et la reconnaissance de documents*, a émergé suite au besoin croissant de développer des méthodes pour les traiter et les analyser automatiquement. Afin d'identifier toutes les zones homogènes du document (texte, graphisme, etc.) et sa structuration hiérarchique, l'analyse du document, ou reconnaissance de la structure physique du document, est une démarche difficile (paragraphe, mot, lettre.). Malgré les efforts des chercheurs, il reste encore un certain nombre de défis dans le domaine de l'analyse de documents, qui est encore très vaste et complexe.

Une fois le document segmenté, et la structure physique du document extraite, vient la dernière phase avant l'étape de décision ou de reconnaissance, qui est la phase d'extraction de caractéristiques. Cette dernière vise à identifier les informations clés contenues dans le document segmenté et à les représenter sous forme de caractéristiques exploitables.

Dans ce chapitre, nous effectuerons un survol sur ce domaine passionnant, en explorant les différentes facettes de l'analyse de documents et de l'extraction de caractéristiques. Ensemble, nous serons en mesure d'approfondir notre compréhension des méthodes avancées utilisées dans ce domaine et d'appréhender leur importance dans des applications pratiques.

Nous commencerons par définir ce qu'est un document et examinerons les différentes catégories de documents manuscrits. Ensuite, nous nous pencherons sur l'analyse de documents, en discutant de ses objectifs et de ses étapes essentielles. Nous aborderons en détail le prétraitement, la segmentation et l'extraction des caractéristiques, en explorant les méthodes classiques ainsi que les avancées récentes dans ces domaines. Nous présenterons les différentes méthodes de segmentation, les niveaux de segmentation, en soulignant leur importance dans la compréhension de la structure du document. Dans la section consacrée à l'extraction des caractéristiques, nous passerons en revue un large éventail de techniques utilisées pour représenter les documents de manière quantitative.

## II.2. Qu'est-ce qu'un document?

### II.2.1. Définition

Les questions liées aux documents sont traitées dans un certain nombre d'articles scientifiques, mais peu d'entre eux s'efforcent de définir le terme au sens large. D'autre part, les dictionnaires, les encyclopédies et les répertoires contiennent un grand nombre de définitions. Dans cette section, nous en montrons quelques-uns.

Selon le dictionnaire LE PETIT ROBERT, "*Un document est tout ce qui sert à instruire, y compris tout écrit servant de preuve ou d'information, tout ce qui sert de preuve de témoignage, et tout document permettant d'identifier les marchandises dans les cours de transport*".

Une définition plus large donnée par Karim HADJAR [42] dans sa thèse de doctorat annonce qu'en fonction du support choisi, un document peut être de différentes natures (textuel, sonore, vidéo, graphique, etc.). Selon lui, *le support physique de stockage et de transmission des connaissances* est un document.

Selon Bachimont [43], un support physique et le papier sont indissociables. Un document est en effet « une chose matérielle représentant un contenu ». Le support d'enregistrement utilisé pour véhiculer un contenu est une chose matérielle. Le matériel ou l'ensemble de connaissances à exprimer est appelé *le contenu*.

### II.1.2. Document électronique

Un document électronique est une représentation d'un document qui peut être transmise entre ordinateurs sous la forme d'une structure de données conservée en mémoire ou sur un support informatique. La principale caractéristique d'un document électronique est sa simplicité de modification. Tout document informatique peut être édité ; diverses opérations y sont utilisables ; parmi eux, nous citons l'impression, l'édition et la copie d'un support à un autre (diffusion illimitée) [42].

L'authenticité du document électronique est donc une source constante de discorde. Le travail que nous faisons avec des documents manuscrits est beaucoup plus intéressant pour nous.

## II.3. Catégories de documents manuscrits

Indépendamment de la langue, les papiers manuscrits peuvent généralement être divisés en quatre catégories en fonction de leur structure [44]. Ces catégories comprennent :

### II.3.1. Documents mono-orientés

Les lignes dans cette classe sont orientées suivant (en adoptant) une seule direction. Voir la figure II.1.a.

### II.3.2. Documents multi-orientés

Lorsqu'elles sont disposées en blocs, les lignes de ces documents remplissent ou non toute la largeur du document. Ils procèdent de diverses manières. Regardez la figure II.1.b.

### II.3.3. Documents multi-scripts

Ce sont des documents rédigés par de nombreux auteurs, donnant naissance à de nombreux écrits ou scripts. Dans le passé, il était courant que des individus travaillent en tandem pour terminer le même document ou co-écrire le même livre. Les papiers actuels comme les cartes de vœux, les écrits pour diverses nationalités, etc., peuvent parfois être multiscrit. Voir la figure II.1.c.

### II.3.4. Documents hétérogènes

Du texte et des images ou des illustrations sont tous deux présents dans ce type de documents. Ces types de ces derniers sont courants dans les domaines cartographiques, mécaniques et architecturaux. Ces types de documents manuscrits peuvent avoir des orientations différentes qui correspondent aux lignes de cote ou à l'illustration des dessins. Voir la figure II.1.d



Figure II.1. Les catégories des documents manuscrits [44].

## **II.4. Analyse de documents**

### **II.4.1. Définition**

L'analyse de documents, l'analyse d'images de documents ou la reconnaissance de la structure physique de documents est un domaine d'étude dans le domaine du traitement d'images. Son objectif principal est de modifier visuellement, archiver, rechercher, réutiliser et transmettre les informations contenues dans les images de documents [45].

L'analyse de documents est décrite comme "une théorie et une pratique de reconstruction de la structure symbolique des images numériques directement produites par l'ordinateur ou simplement numérisées à partir du papier" [46].

### **II.4.2. Objectif de l'analyse de documents**

Afin de séparer l'image du document en lignes de texte et les caractères qu'elles contiennent, l'analyse de l'image du document la divise d'abord en régions ou blocs homogènes (texte, graphiques, photos, etc.). La structure physique du texte est constituée de ces composants.

## **II.5. Etapes d'analyse de documents**

L'analyse de documents est cruciale pour le processus d'identification et de compréhension des documents. Cependant, à partir d'un document papier, une version électronique utile est créée grâce à la reconnaissance. Il faut plusieurs étapes pour atteindre cet objectif.

La figure II.2 illustre de manière plus détaillée ces étapes :

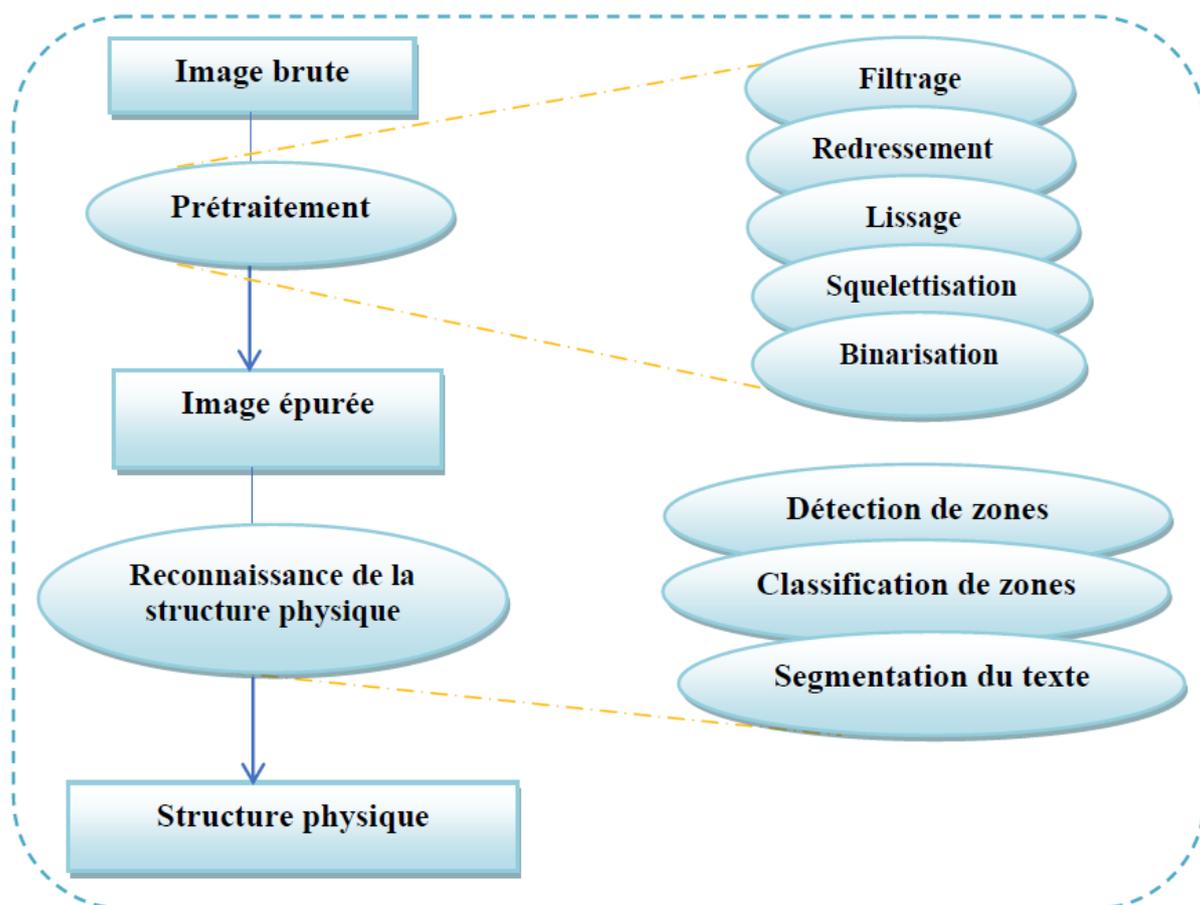


Figure II.2. Etapes de l'analyse de documents [47].

### II.5.1. Prétraitement

Dans tous les systèmes de traitement, d'analyse et de reconnaissance de documents, le prétraitement est une étape cruciale et nécessaire. Il consiste à supprimer les défauts de l'image numérisée afin d'accélérer l'étape d'identification. Ces défauts peuvent appartenir à l'une des deux catégories suivantes : ceux liés au processus de numérisation (tels que l'inclinaison, la luminosité, le bruit, etc.) ou ceux liés à la qualité inhérente du document (tels que les zones humides, l'aspect du verso, trous, etc.) [48]. En effet, plusieurs traitements peuvent être inclus dans le prétraitement. Nous citons dans cette section quelque uns :

#### II.5.1.1. Lissage

Le lissage local (ou filtrage) s'agit de supprimer le bruit, ou les petites variations, présent dans une image. L'intensité d'un pixel est transformée en fonction des intensités sur un petit voisinage du pixel [49].

Pour éliminer les points parasites, [50] utilisent un filtrage passe bas, d'autres utilisent un filtre passe haut ou un filtrage morphologique ou encore des ondelettes [51] [52].

#### II.5.1.2. Binarisation

La binarisation est une opération indispensable dans presque tous les systèmes d'analyse et de reconnaissance de document. Elle permet de séparer le premier plan du fond de l'image, ce

qui donne deux classes de pixels : une classe représentant la scène de l'image (l'avant-plan), en noir, et l'autre classe représentant le fond de l'image (l'arrière plan), en blanc.

En effet, un grand nombre de techniques de binarisation ont été proposées dans la littérature. La majorité de ces techniques utilisent un ou plusieurs seuils pour séparer les pixels en deux classes: noir et blanc, c'est pourquoi ces méthodes sont appelées méthodes de seuillage. Ainsi, le seuillage peut être global, local, ou mixte.

Les méthodes de seuillage global calculent un seuil unique dans toute l'image. Parmi les méthodes de seuillage global les plus connues il y a: la méthode d'Otsu [53], de Kapur [54], et ISODATA [55].

Par opposition aux méthodes globales, les méthodes de seuillage local calculent un seuil différent pour chaque pixel ou région de l'image. Parmi les méthodes de cette catégorie nous citons: la méthode de Niblack, et Sauvola.

Les méthodes de seuillage mixte combinent des informations globales et locales pour effectuer la binarisation.

### **II.5.1.3. Redressement**

Le redressement est une opération courante dans l'analyse de documents, souvent à la suite d'un positionnement incorrect du document sur le scanner, ce qui a pour effet d'incliner l'image. Au cours des deux dernières décennies, un certain nombre de techniques de redressement ont été proposées, et parce que ces opérations sont si fréquentes, la majorité des systèmes de reconnaissance optique de caractères (OCR) en sont actuellement équipés.

Les meilleurs algorithmes qui ont été proposés sont sans conteste ceux qui sont le moins impactés par la présence de graphismes, de zones noires dans le texte avec des inclinaisons variables, ou de zones d'ombre proches des marges causées par un phénomène de renflement qui se produit lors de la saisie de livres ou de périodiques.

Ici aussi, il existe plusieurs méthodes de redressement dans la littérature exploitant diverses informations et employant différentes techniques. Il existe ainsi des:

- Méthodes basées sur la transformée de Hough
- Méthodes basées sur le profil de projection
- Méthodes basées sur le regroupement
- Méthodes basées sur la morphologie

Une méthode basée sur l'analogie entre les lignes textuelles et les antennes radar a été proposée par Agajan et Kailath [56]. La distance de chaque pixel d'arrière-plan à une ligne de référence est convertie en une phase d'un complexe sinusoïdal d'onde. L'algorithme de détection détermine la cohérence spatiale entre les différentes entrées à l'aide de différentes lignes. Bien que le concept de la transformée de Hough soit similaire, cette approche semble plus efficace, selon ses créateurs. Plus récemment, Chaudhuri & Pal ont proposé une méthode pour les papiers indiens multi-scripts (Devanagari et Bangla).

#### **II.5.1.4. La squelettisation**

L'image binaire est une série de lignes d'épaisseurs variables, généralement avec du bruit. L'application d'une étape de squelettisation à l'image est parfois nécessaire. Ainsi, la squelettisation est appliquée aux images suite à la binarisation, et en concevant sa continuité, elle vise à limiter l'épaisseur du contour du caractère à un pixel. Le processus implique l'exécution de procédures d'érosion conditionnelle supplémentaires jusqu'à ce que le résultat souhaité soit atteint. Le squelette résultant doit conserver la forme du personnage et s'abstenir d'ajouter du bruit [57]. Figure II.3. illustre un exemple de résultat de cette opération.

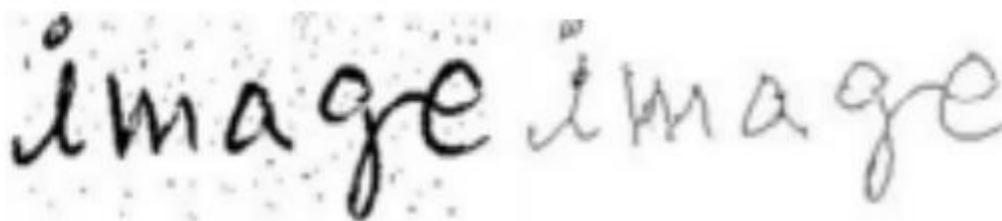


Figure II.3. Squelettisation d'une image [58]

#### **II.5.2. Segmentation**

C'est l'analyse de la mise en page du document afin de trouver sa structure physique. Il s'agit de segmenter l'image de document en composantes homogènes et de classifier chaque zone en texte, image, graphique, etc. [59].

Selon [59] Les deux familles de la méthodes de segmentation sont les méthodes classiques (*approche descendante, approche ascendante et approche mixte.*) et les méthodes basées sur la texture.

##### **II.5.2.1. Les méthodes classiques**

###### **a) Approche ascendante (Bottom-up)**

Les méthodes ascendantes reposent sur l'analyse pixel par pixel d'une image. Selon les principes de base de l'approche ascendante, chaque composante connexe est d'abord combinée en un mot, qui est ensuite combiné en une ligne, qui est ensuite combinée en un bloc, et ainsi de suite, jusqu'à ce que la page entière ait été reconstruite.

Il existe de nombreuses techniques ascendantes, notamment celles qui utilisent l'algorithme RLSA (*Run Length Smoothing Algorithm*), les composantes connexes, le filtrage par fenêtre, la technique docstrum, etc.

###### **a.1) Méthodes utilisant l'algorithme de lissage RLSA**

Le RLSA (*Run Length Smoothing Algorithm*) [60] consiste à relier les pixels noirs d'un document entre eux si leurs distances est inférieure à un certain seuil. Suivant le seuil choisi,

cela permet de segmenter une lettre, un mot, une ligne ou un paragraphe. La figure suivante illustre ce principe.

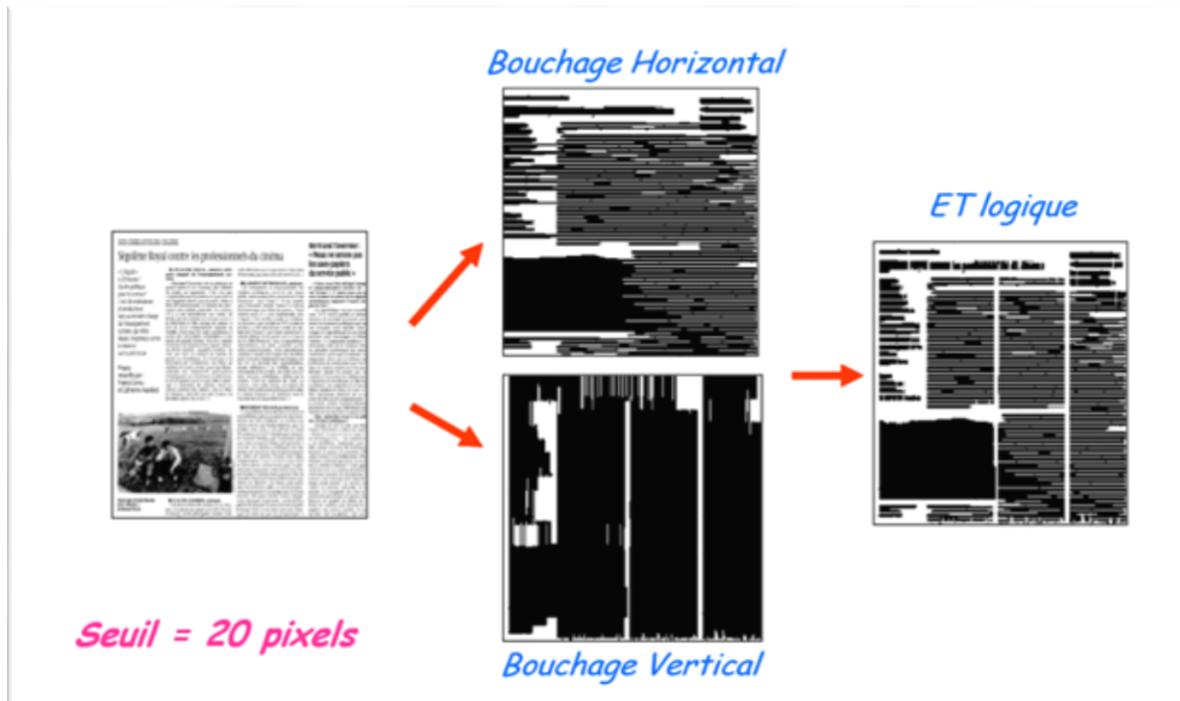


Figure II.4. Exemple d'application de RLSA[61].

Wang [62] propose une méthode basée sur la combinaison entre l'algorithme de lissage RLSA et l'algorithme de découpage X-Y récursif pour accomplir la segmentation d'image en blocs. Cette segmentation est suivie par une étape d'analyse de la texture pour classer les blocs résultants.

### ***b) Approche descendante (top down)***

Lors de la création de documents avec une structure clairement spécifiée, la méthode descendante est fréquemment utilisée. Ces approches commencent au niveau le plus élevé, qui est la page, et descendent jusqu'au niveau des composantes connexes ou au niveau du pixel (le niveau le plus bas). L'algorithme de découpage d'arbre X-Y, et les méthodes basées sur une analyse du fond blanc de l'image sont quelques techniques descendantes qui ont été suggérées dans la littérature.

#### ***b.1) L'algorithme de découpage X-Y***

L'algorithme de découpage en X-Y a été introduit par Nagy et al dans [63]. Le principe consiste à découper un document binaire horizontalement et verticalement en plusieurs rectangles. Le découpage continue dans chaque rectangle d'une manière récursive jusqu'à une condition soit satisfaite. La condition d'arrêt est définie selon l'application souhaitée. Dans ce travail, les profils de projection horizontaux et verticaux sont étudiés pour définir les conditions d'arrêt afin d'extraire les lignes. La méthode a été appliquée sur des articles latins imprimés. Les auteurs donnent quelques résultats d'extraction de lignes sans toutefois

mentionner le taux d'extraction. L'algorithme de découpage en XY a été utilisé pour des documents qui ne contiennent pas beaucoup des variations.

b.2) Méthodes basées sur l'analyse de fond de l'image

*c) Approche mixte*

Les approches mixtes résultent de la combinaison des méthodes descendantes et ascendantes ou de l'utilisation conjointe d'une de ces dernières avec une autre méthode [42].

**II.5.2.3. Les méthodes de segmentation à base de texture**

La discipline de l'analyse d'images a servi de modèle à ces techniques. Les méthodes classiques ayant montré leurs insuffisances, elles n'utilisent qu'un minimum d'informations a priori.

La méthode de segmentation de texture combine une variété de méthodes. Ces méthodes cherchent à identifier les traits visuels propres au texte. L'image est ensuite transformée en une représentation qui met l'accent sur ces qualités à l'aide d'une variété de filtres.

**II.5.3. Les niveaux de segmentation**

Le document cible doit être divisé en lignes, en mots ou en composantes connexes selon la technique de suivi des mots utilisée. Du fait de l'uniformité des espaces entre les mots et des distances entre les lignes dans les imprimés, la segmentation est plus simple. Lorsqu'il s'agit de documents manuscrits, la segmentation est plus difficile car la taille de l'écriture est irrégulière et les mots et les lignes peuvent se chevaucher.

*a) Segmentation en lignes*

La division à l'aide d'un profil de projection horizontale [64]. Les lignes de texte et les espaces entre elles sont représentés par les pics et les minima dans l'histogramme de la projection horizontale. Ces techniques, mieux adaptées aux documents imprimés, permettent de catégoriser les papiers manuscrits convertis en lignes de texte.

La technique de lissage [65] consiste à noircir les minuscules espaces entre des pixels noirs consécutifs dans un plan horizontal, qui les relieront. Les lignes sont obtenues en étiquetant les composantes connexes à partir de l'image résultante du noircissement.

La segmentation par regroupement [66] crée des lignes de texte à partir de composants de base comme les pixels et les parties liées. Afin d'identifier des lignes de texte, des techniques basées sur la transformée de Hough projettent un ensemble de coordonnées dans l'espace de Hough.

*b) Segmentation en mots*

La même idée s'applique à la segmentation des mots dans certaines techniques de division de lignes de texte à l'aide de projections horizontales. Les bordures de mots sont localisées à l'aide de profils verticaux. Les profils de projection horizontale sont utilisés par Rodriguez-

Serrano et Perronin [67] pour générer des lignes de texte. Ils déterminent les boîtes de composants pour chaque ligne.

### *c) Segmentation en sous-mots*

Dans les textes arabes, la segmentation des sous-mots est fréquemment utilisée dans les techniques de recherche de mots. La reconnaissance visuelle des bordures de mots est impossible car elles reposent sur la sémantique du texte. La majorité des techniques de recherche de mots dans les textes arabes segmentent le document en parties liées, appelées sous-mots ou PAW en anglais (pour Part of Arabic Word).

Une méthode pour extraire les PAW d'une ligne de texte a été proposée par Khayat et al [10]. Ils croient qu'un composant principal lié et quelques composants mineurs constituent une partie d'un mot arabe (points diacritiques). Avant de faire correspondre les composantes connexes (CC) mineurs à leurs principaux CC voisins les plus proches, l'algorithme sépare d'abord les principaux CC en fonction de leur taille et de leur emplacement.

## **II.6. Extraction des caractéristiques (features extraction)**

Avant la classification, l'extraction de caractéristiques (ou primitives) est une étape critique dans tout système de reconnaissance de forme ou d'analyse de document [68].

Son but est d'utiliser un vecteur de caractéristiques pour caractériser une donnée. Ensuite, en utilisant les vecteurs caractéristiques des caractères, des mots, etc., la ressemblance ou la différence entre eux est calculée.

Pour un système de repérage, de reconnaissance et de classification fiable, la phase d'extraction de caractéristiques sur les images pour représenter l'information est cruciale. Pour l'analyse de l'écriture manuscrite, des images bidimensionnelles de mots, de lignes ou de blocs sont converties en vecteurs de caractéristiques, qui sont des ensembles de nombres qui reflètent chacun une partie de l'image. Les algorithmes sont alimentés avec ces données [69].

Le défi d'extraire d'une image les données les plus pertinentes pour un problème de classification particulier, c'est-à-dire celui qui minimise la variabilité intra-classe et qui maximise la variabilité inter-classe, est appelé extraction de caractéristiques par Devijver et al. [70].

Cependant, les exigences suivantes doivent être satisfaites par les caractéristiques extraites : compacité, résistance au bruit, résistance à la déformation et pouvoir discriminant [68].

L'extraction de caractéristiques à partir d'images de documents peut s'intéresser à la fois au codage des parties graphiques en utilisant typiquement des caractéristiques visuelles de l'image, ou elle ne peut concerner que la partie textuelle du document, notamment à des fins de reconnaissance de caractères, de recherche, d'étude, etc. [68].

Généralement les caractéristiques les plus utilisées pour le codage du texte et du graphique en Parties textuelles on peut citer : 1) Valeurs ou densités de pixels, 2) Histogrammes des projections, 3) Caractéristiques de contour, 4) Caractéristiques structurales, 5) Caractéristiques métriques, 6) Caractéristiques morphologiques, 7) Moments invariants, 8)

Descripteurs de Fourier elliptiques et 9) Caractéristiques adaptatives. En revanche dans les parties graphiques les caractéristiques visuelles les plus utilisées sont: la couleur, la texture et la forme [68].

### II.6.1. Les caractéristiques structurelles

La disposition des pixels les uns par rapport aux autres est étudiée par des caractéristiques structurelles [69]. En fonction des caractéristiques topologiques et géométriques du caractère, telles que les lignes de référence, les ascendantes et les descendantes, les points de croisement, les primitives et leurs orientations, l'inflexion entre deux points, etc., elles sont utilisées [71]. Rendre ces traits insensibles aux diverses variations que doivent subir les personnages est le premier enjeu de ces approches [68].

### II.6.2. Caractéristiques de contour

Une liste de contours, qui peuvent inclure des contours extérieurs ou intérieurs, est utilisée pour représenter l'image d'un mot dans le traçage de contour d'image. Les composantes connexes principales du mot ou ses composantes connexes secondaires (points diacritiques) correspondent aux contours externes, tandis que les occlusions correspondent aux contours internes [72]. Les caractéristiques peuvent être dérivées du codage de Freeman, déduites de composants particuliers (tels que des points de rebroussement), ou utilisées en conjonction avec un contour qui a été codé (par des moments de Fourier ou de Zernike).

### II.6.3. Histogrammes des projections

L'histogramme de projection horizontale (histogramme horizontal) ou l'histogramme de projection verticale (histogramme vertical) mesure le nombre total de pixels noirs (en tant que partie de caractère ou pixels de premier plan) dans chaque ligne ou colonne d'une image. Les directions diagonales (diagonale gauche ou diagonale droite) peuvent également être utilisées pour calculer l'histogramme de projection [74]. Les caractéristiques peuvent être extraites directement des valeurs de l'histogramme ou peuvent être extraites de ces histogrammes comme, par exemple, les pics [68]. Une illustration d'histogramme de projection pour la lettre "a" est illustrée à la figure suivante.

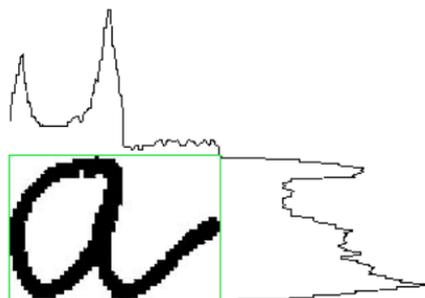


Figure II.5. histogrammes de projections verticale et horizontale pour la lettre "a" [75].

### II.6.4. Valeurs ou densités de pixels

L'idée derrière les valeurs de pixel est très simple ; cela implique de diviser l'image en boîtes NM et de déterminer le nombre de pixels dans chacune d'elles. Le nombre de pixels n'a pas

besoin d'être traité. Ces caractéristiques peuvent être utilisées dans les techniques d'appariement de modèles comme des caractéristiques simples mais communes [68].

### II.6.5. Moments invariants

Parce qu'ils sont invariants en translation, en taille et en rotation, les moments invariants sont considérés comme des caractéristiques importantes [68]. Les moments, qui indiquent la répartition des pixels autour du centre de gravité du personnage, fournissent des détails extrêmement importants sur la forme générale du personnage [74].

Par rapport aux autres techniques d'extraction de caractéristiques, les moments de Hu ne prennent en compte que 7 valeurs, ce qui accélère le calcul [71]. Les moments invariants actuellement les plus utilisés sont ceux dérivés des polynômes de Zernik [69], qui sont robustes au bruit et aux petites variations, invariables à la rotation, et contiennent également moins d'informations redondantes [68] [71][76].

### II.6.6. Descripteurs de Fourier elliptiques

Cette méthode est construite sur la représentation fréquentielle de l'image [71]. L'une des techniques les plus populaires pour l'identification des motifs et des caractères est la transformée de Fourier [77]. Les descripteurs de Fourier, qui sont basés sur les coefficients complexes de la série de Fourier, sont ce qui est réellement extrait en tant que caractéristiques. Ces traits peuvent être identifiés par leur non-variation aux rotations et aux changements d'échelle [68].

### II.6.7. Caractéristiques adaptatives

Les fonctionnalités adaptatives nécessitent une phase d'apprentissage et sont directement extraites de l'image. En d'autres termes, l'extracteur de caractéristiques doit être construit et optimisé par le système lui-même [68] puisqu'il travaille sur une représentation qui ressemble beaucoup à l'image originale.

### II.6.8. Caractéristiques métriques

En fonction des dimensions physiques de l'image, cette classe divise les entités en catégories. Les traits les plus simples, comme la hauteur, la largeur et la proportion entre ces deux tailles, se retrouvent dans ce groupe. De plus, des traits plus complexes comme le codage de profil [68] peuvent être utilisés. Ceux-ci déterminent le nombre de pixels entre le contour extérieur du caractère et les bordures gauche, droite, supérieure et inférieure de l'image du caractère (Figure II.6). Ces traits sont employés car ils capturent avec précision la forme extérieure des personnages [71] [78].



Figure II.6. Les quatre profils de projection pour la lettre "a" profils haut, bas, gauche et droit [75].

### II.6.9. Caractéristiques à base du gradient

Le gradient est une mesure de la variation de l'intensité des pixels dans une image. Il permet de capturer les changements d'intensité et d'orientation des traits présents dans une image, y compris dans le contexte de l'écriture. C'est pourquoi les caractéristiques basées sur le gradient sont largement utilisées dans le domaine du traitement d'images et de la vision par ordinateur pour l'analyse de l'écriture manuscrite.

Les caractéristiques basées sur le gradient fournissent des informations précieuses sur la direction et la magnitude des contours et des lignes présents dans une image. Elles permettent de quantifier la texture, la forme et la structure des objets, y compris les traits d'écriture. En particulier, ces caractéristiques sont utiles pour représenter et extraire des informations discriminantes des caractères, des mots ou des lignes d'écriture.

Les exemples typiques de ce type de caractéristiques sont le HOG (Histogram of Oriented Gradient) et le SIFT (Scale Invariant Feature Transform).

HOG est une technique qui permet de représenter l'information de gradient dans une image en calculant et en quantifiant les orientations des gradients locaux. Cette technique a été initialement proposée pour la détection d'objets et s'est avérée efficace pour la reconnaissance de formes, y compris dans le contexte de l'analyse de l'écriture manuscrite. HOG divise l'image en cellules, calcule les gradients dans chaque cellule, et crée un histogramme des orientations de ces gradients. Ces histogrammes sont ensuite normalisés pour obtenir une représentation robuste des caractéristiques de l'image.

SIFT est une méthode de détection et de description de points d'intérêt dans une image, qui est conçue pour être robuste aux changements d'échelle, de rotation et d'éclairage. SIFT identifie des points-clés caractéristiques de l'image, tels que des coins ou des zones distinctives, et génère des descripteurs uniques pour chacun de ces points-clés. Ces descripteurs sont basés sur des informations locales, telles que les orientations des gradients, et sont invariants par rapport aux transformations géométriques. SIFT est largement utilisé pour l'appariement d'images, la reconnaissance d'objets et la correspondance d'images, et peut également être appliqué à l'analyse de l'écriture manuscrite.

Ainsi, l'algorithme du SIFT fonction en deux étapes:

- Détection de points clés: L'image est filtrée à différentes échelles en utilisant une série de filtres Gaussiens pour détecter des caractéristiques à différentes tailles. Cela permet de détecter des points-clés à différentes échelles.
- Description des points d'intérêt : Pour chaque point-clé, une orientation dominante est déterminée. Cela est réalisé en calculant les orientations des gradients locaux dans une région autour du point-clé.

### II.7. Conclusion

L'analyse de documents joue un rôle crucial dans le traitement d'images en répondant à la nécessité croissante de gérer efficacement le volume massif de documents stockés dans différentes institutions et organisations. Son objectif principal est d'automatiser la tâche

laborieuse de la saisie manuelle en développant des méthodes pour traiter et analyser les documents de manière automatique.

L'analyse de documents vise à reconnaître la structure physique des documents en identifiant les différentes zones homogènes telles que le texte, les graphismes, etc. Cela permet de mieux comprendre la hiérarchie du document, allant des paragraphes aux mots et aux lettres. Malgré les progrès réalisés par les chercheurs, ce domaine complexe et vaste présente encore des défis à relever.

Ce chapitre a jeté les bases de notre exploration de l'analyse de documents et de l'extraction de caractéristiques. Nous avons examiné les différentes catégories de documents manuscrits et identifié les défis spécifiques associés à chacune d'entre elles. De plus, nous avons compris les objectifs et les étapes clés de l'analyse de documents, en mettant l'accent sur le prétraitement, la segmentation et l'extraction des caractéristiques.

Au cours de notre parcours, nous avons découvert une multitude de techniques et d'approches permettant de segmenter les documents d'une manière efficace selon plusieurs niveaux. Nous avons également exploré différents descripteurs et caractéristiques utilisées pour représenter les documents de manière quantitative. Ces caractéristiques permettent de capturer les informations essentielles contenues dans les documents.

# Chapitre III. Conception

---

## III. 1. Introduction

Dans ce troisième chapitre, nous détaillons la méthodologie mise en œuvre pour réaliser notre application de repérage de mots (word spotting). Ce chapitre présente les différentes étapes et procédures que nous avons suivies pour atteindre nos objectifs de recherche.

Ce chapitre constitue ainsi un guide détaillé de la conception de notre application, fournissant les informations nécessaires pour reproduire nos expériences et évaluer la validité de nos résultats.

## III. 2. Description de l'approche proposée

Dans cette section, nous détaillons l'approche que nous avons proposé pour le repérage de mots dans les images de documents.

Notre approche s'inscrit dans la catégorie des méthodes analytiques de repérage de mots, qui se concentre sur la segmentation de l'image de documents en mots individuels. Dans notre proposition, la requête est formulée sous forme d'image requête (QBE), où l'utilisateur fournit une représentation visuelle du mot recherché. Cette approche nous permet d'effectuer une recherche précise en comparant les caractéristiques visuelles de l'image requête avec celles des mots présents dans la base de données. En utilisant cette méthode, nous visons à faciliter et optimiser le processus de recherche de mots dans les documents.

Pour atteindre notre objectif, nous avons conçu une application qui intègre plusieurs étapes de traitement scindées en cinq parties ou phases principales.

- **Prétraitements:** comprend des opérations visant à améliorer la qualité des données et à les rendre plus adaptées aux analyses ou aux algorithmes qui seront utilisés.
- **Segmentation:** pour diviser l'image en un ensemble de mots.
- **Extraction de caractéristiques :** afin de représenter les mots par un vecteur de caractéristiques distinctives.
- **Clustering:** pour regrouper des formes similaires en ensembles.
- **Recherche:** consiste à comparer les caractéristiques des mots de la requête avec celles des mots de la base de données afin de retourner les images pertinentes.

La figure III.1 décrit l'architecture générale de notre approche.

Notre travail porte principalement sur l'extraction de caractéristiques, le regroupement des formes similaires, et le repérage de mots (ou l'appariement des caractéristiques de mots) plutôt que sur le prétraitement et la segmentation. Le prétraitement et la segmentation sont des problèmes de recherche en eux-mêmes, qui peuvent constituer des sujets de mémoires et de thèses indépendants. Ces domaines exigent des investigations approfondies et des contributions significatives pour résoudre les défis qui leur sont propres.

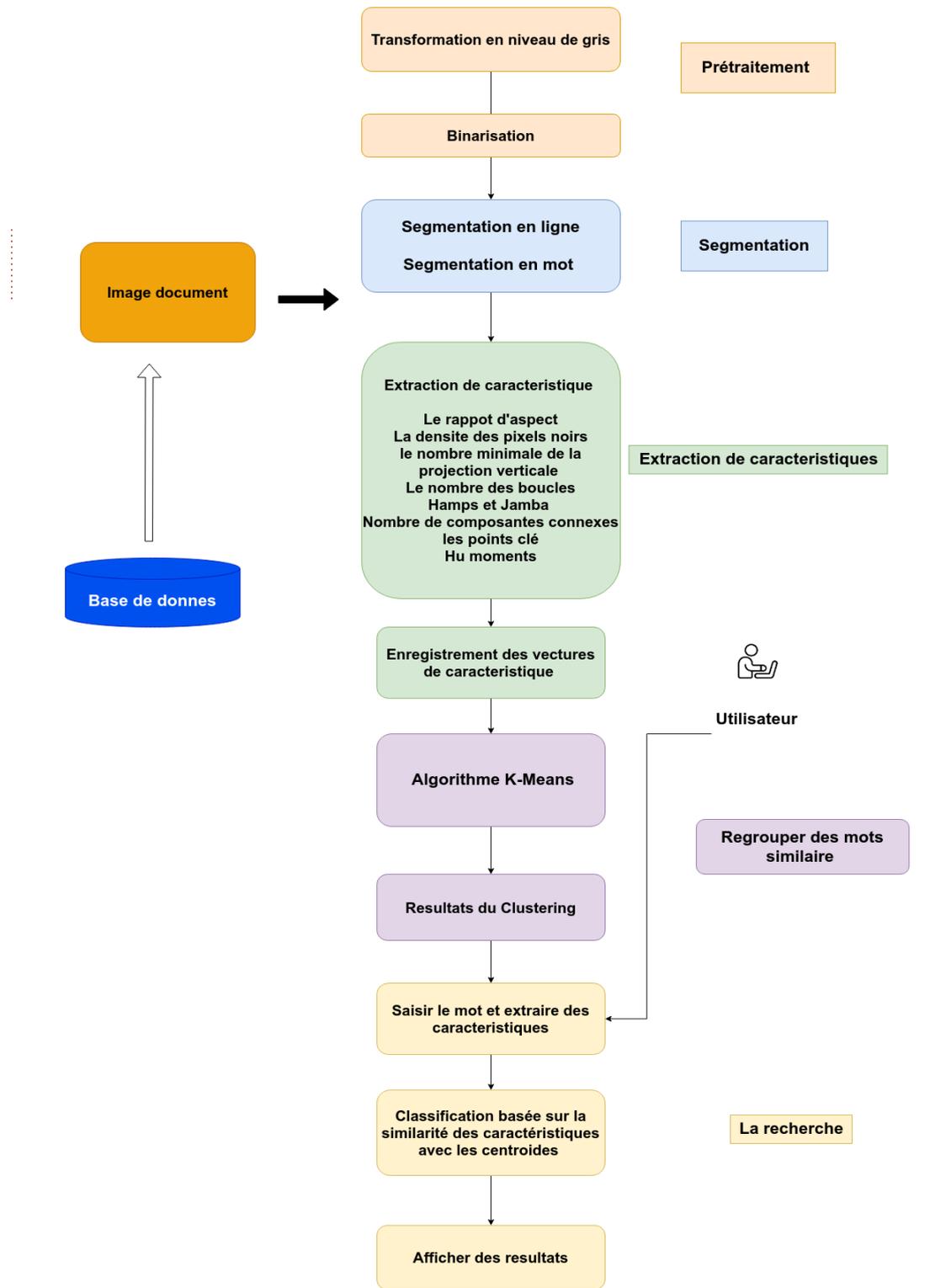


Figure III.1. Architecture générale de l'approche proposée

Nous allons maintenant détailler chacune des étapes de notre approche proposée. En ce qui concerne le prétraitement et la segmentation, nous avons opté pour des techniques simples et populaires afin de clarifier les concepts. Cependant, notre principal axe de travail a été axé sur l'extraction de caractéristiques et l'identification des mots. Nous avons consacré nos efforts à ces aspects spécifiques de notre méthode.

### III.2.1. Prétraitement d'images

#### III.2.1.1. Transformation en niveaux de gris

La première étape dans notre approche est la conversion des images couleur en niveaux de gris afin de simplifier le traitement et réduire les dimensions des données.

La transformation d'une image couleur en une image en niveaux de gris est une étape préalable essentielle pour réaliser la binarisation, la deuxième étape dans notre approche.

Elle pourra être accomplie en utilisant le pseudo-code suivant:

Algorithmme

Entrée : Image couleur ( $I$ )

Sortie : Image en niveau de gris ( $I_g$ )

Début

Pour chaque pixel  $(x,y)$  de l'image  $I$  faire

$R \leftarrow$  la quantité de la couleur rouge de  $I(x,y)$ ;

$V \leftarrow$  la quantité du couleur vert de  $I(x,y)$ ;

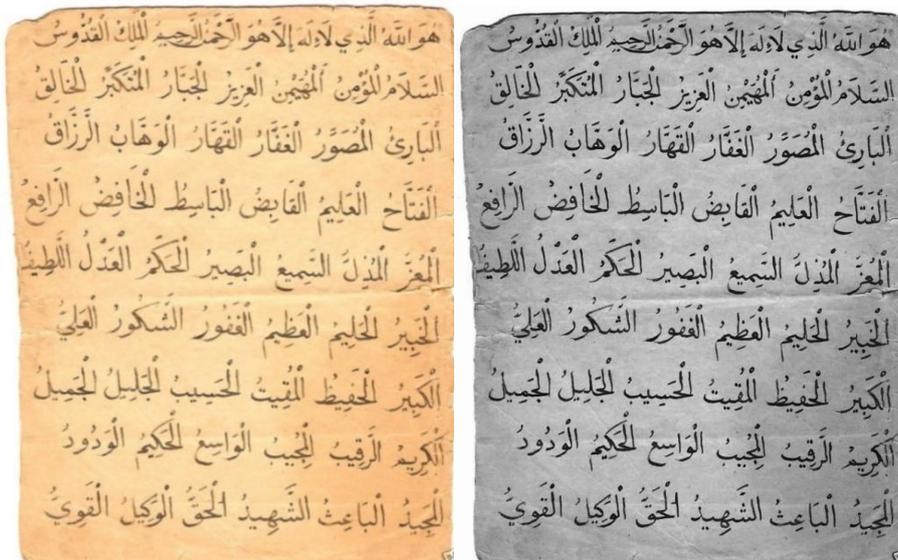
$B \leftarrow$  la quantité du couleur bleu de  $I(x,y)$ ;

$I_g(x,y) \leftarrow (R+V+B)/3$ ;

Fin pour;

Fin.

Un exemple d'une telle transformation est illustré par la figure suivante:



(a) Image couleur

(b) Image en niveaux de gris

Figure III.2. Transformation en niveaux de gris.

### III.2.1.2. Binarisation

Le deuxième prétraitement procédé dans notre approche est la binarisation. Cette étape consiste à transformer l'image en niveaux de gris en une image binaire contenant uniquement deux couleurs: noir et blanc. Cette opération sert à séparer l'information du texte du reste de l'image ce qui permettra de mettre en évidence les caractères et les mots présents dans le document(voir la figure III.3).

La méthode la plus couramment utilisée pour effectuer cette transformation est la méthode d'Otsu. C'est une méthode de seuillage global, qui permet de déterminer automatiquement un seuil optimal pour binariser l'image et séparer les pixels en deux classes : les pixels de l'avant-plan (texte ou objet) et les pixels de l'arrière-plan (fond).Le seuil d'Otsu est choisi comme étant le niveau de gris qui maximise la variance inter-classes ou bien qui minimise la variance intra-classes). Le calcul de la variance est basée sur l'histogramme normalisé  $H2 = [h_0 \dots h_{255}]$  de l'image d'où  $\sum h_i = 1$ .

La variance inter-classes pour chaque niveau de gris  $t$  est donnée par :

$$\sigma_{inter}^2 = q_1(t) \times q_2(t) \times [\mu_1(t) - \mu_2(t)]^2.$$

Tel que:

$$q_1(S) = \sum_{i=0}^{S-1} H2(i) ; q_2(S) = \sum_{i=S}^{255} H2(i) ;$$

$$\mu_1(S) = \frac{1}{q_1(S)} \sum_{i=0}^{S-1} H2(i) \times i ; \mu_2(S) = \frac{1}{q_2(S)} \sum_{i=S}^{255} H2(i) \times i ;$$

La binarisation en utilisant l'algorithme d'Otsu peut être résumée par le pseudo-code suivant:

Algorithme

Entrée : Image en niveau de gris ( $Ig$ )

Sortie : Image binarisée ( $Ib$ )

Début

$S \leftarrow$  le seuil de binarisation optimal obtenu par la méthode d'Otsu.

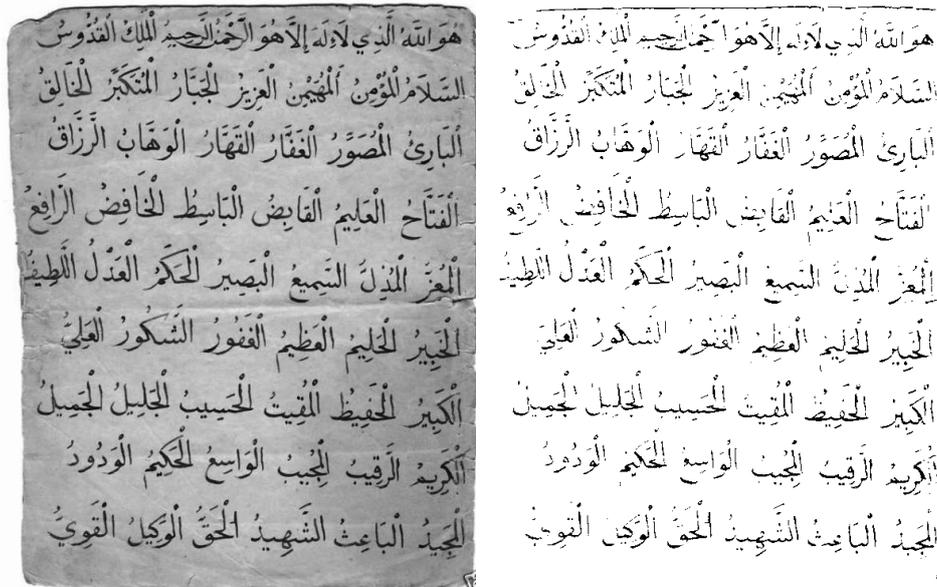
Pour chaque pixel  $(x,y)$  d'image  $Ig$  faire

Si  $Ig(x,y) \geq S$  alors  $Ib(x,y) \leftarrow$  blanc

Sinon  $Ib(x,y) \leftarrow$  noir

Fin pour;

Fin.



(a) Image en niveau de gris

(b) Image binaire

Figure III.3. Résultat de binarisation d'une image en niveaux de gris

### III.2.2. Segmentation

La deuxième phase dans notre proposition est la segmentation. Comme nous avons avancé dans le chapitre 1, la majorité des travaux de repérage de mots proposés dans la littérature reposent sur la segmentation.

La segmentation, dans le contexte des images de documents, est le processus de division d'une image de documents en unités plus petites et significatives, afin de faciliter leur analyse et leur traitement ultérieurs. Ces unités peuvent être graphiques (comme les figures, les schémas,...) ou textuelles (paragraphe, lignes de textes, mots, caractères,...)

La segmentation constitue en fait une partie essentielle dans les systèmes de repérage de mots. La recherche d'une requête dans les images de documents se fera en fait par appariement de la requête et ces unités extraites.

Dans notre proposition, l'unité de base considéré est le mot. Ainsi, nous nous limitons à la segmentation en lignes et puis en mots.

#### III.2.2.1. Segmentation en lignes

La segmentation en lignes, ou l'extraction de lignes, est une étape spécifique de la segmentation de texte où l'objectif est de diviser un paragraphe ou un bloc de texte en lignes individuelles. Cela permet de traiter chaque ligne séparément, ce qui est utile dans des tâches telles que la reconnaissance optique de caractères (OCR) ou l'analyse de documents.

Dans la littérature, on recense diverses approches pour segmenter le texte en lignes. Ces approches incluent des techniques ascendantes, descendantes et mixtes, et chacune d'entre elles englobe plusieurs techniques spécifiques.

Comme nous avons dit dès le départ, la contribution dans l'étape de segmentation n'est pas notre objectif ultime. Nous implémentons dans notre approche une technique simple de segmentation en lignes, à savoir l'analyse de profils de projection.

Cette technique simple consiste à projeter les valeurs des pixels noirs, dans la direction horizontale de façon à obtenir un histogramme: l'histogramme de projections horizontales. Ce dernier possède des maxima qui représentent les centres des lignes et des minima qui délimitent les bords inférieurs et supérieurs des lignes. L'analyse de cet histogramme nous permet de séparer les lignes du texte.

L'algorithme suivante récapitule la segmentation en lignes d'une image à l'aide de l'analyse de profils de projections horizontales:

Algorithme

Entrées :  $I$ : l'image binaire d'hauteur  $Haut$  et de largeur  $Larg$

Sortie:  $LL[]$ : liste contient des couples  $(x_d, x_f)$  exprimant le début et le fin des lignes de texte

Variables utilisées:

$H[]$ : Tableau représentant l'histogramme de projections horizontales de taille  $T=Haut$

Début

//Calcul de l'histogramme de projections horizontales

Pour  $i$  allant de 0 à  $Haut$  faire

Pour  $j$  allant de 0 à  $Larg$  faire

Si  $I(i, j) = 0$  alors  $H[i] \leftarrow H[i] + 1$  ;

//Extraire les lignes de texte

$i \leftarrow 0$ ;

Tant que  $i < Haut$  Faire

Début

Tant que  $i < T$  et  $H[i] = 0$  Faire  $i \leftarrow i + 1$ ;

Si  $i < T$  alors

Début

$x_d \leftarrow i$ ;

Tant que  $i < T$  et  $H[i] > 0$  Faire  $i \leftarrow i + 1$ ;

$x_f \leftarrow i - 1$ ;

Ajouter le couple  $(x_d, x_f)$  à la liste de lignes  $LL$ ;

Fin;

Fin;

Retourner  $LL$ ;

Fin.

### III.2.2.2. Segmentation en mots

Une fois les lignes de texte sont obtenu, on procède à l'extraction des mots en segmentant ces lignes en unités distinctes représentant chaque mot.

Cette étape est essentielle car dans le module de recherche de notre application, la requête est introduite sous forme de mot, est donc l'appariement doit se faire avec chaque mot des documents de la base.

La séparation des mots se fait en détectant les espaces entre les mots par analyse des profils de projections comme dans l'étape précédente, mais cette fois les projections verticales. Les entrées nuls dans l'histogramme correspondent aux zones de segmentation.

La segmentation en mots peut être résumée par l'algorithme suivant:

Algorithme

Entrées :  $I$ : l'image binaire d'hauteur  $Haut$  et de largeur  $Larg$

$LL[]$ : liste des couples  $(x_d, x_f)$  exprimant le début et le fin des lignes de texte

Sortie:  $LM[]$ : liste des quadruplet  $(x_d, y_d, x_f, y_f)$  exprimant la rectangle englobante des mots

Variables utilisées:

$H[]$ : Tableau représentant l'histogramme de projections verticales de taille  $T = x_f - x_d + 1$

Début

Pour chaque ligne  $L$  de  $LL$  Faire

Début

//Calcul de l'histogramme de projections verticales

Pour  $j$  allant de  $0$  à  $Larg$  faire

Pour  $i$  allant de  $x_d$  à  $x_f$  faire

Si  $I(i, j) = 0$  alors  $H[j] \leftarrow H[j] + 1$  ;

//Extraire les mots

$i \leftarrow 0$ ;

Tant que  $i < T$  Faire

Début

Tant que  $i < T$  et  $H[i] = 0$  Faire  $i \leftarrow i + 1$ ;

Si  $i < T$  alors

Début

$y_d \leftarrow i$ ;

Tant que  $i < T$  et  $H[i] > 0$  Faire  $i \leftarrow i + 1$ ;

$y_f \leftarrow i - 1$ ;

Ajouter le quadruplet  $(x_d, y_d, x_f, y_f)$  à la liste de mots  $LM$ ;

Fin;

Fin;

Fin;

Retourner *LM*;

Fin.

La figure suivante illustre le résultat final de la phase de segmentation.

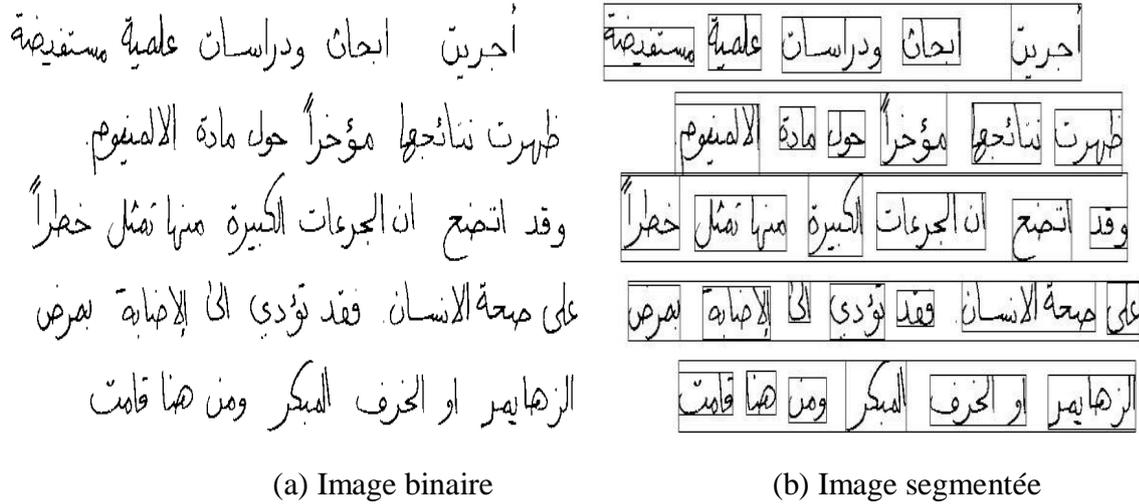


Figure III.41. Segmentation du texte en lignes et puis en mots

### III.2.3. Extraction de caractéristiques

Dans le domaine de l'analyse d'images, l'extraction de caractéristiques implique l'identification et l'extraction de mesures ou de descripteurs spécifiques à partir d'une image. L'objectif est de représenter l'information visuelle de manière compréhensible et exploitable pour des tâches telles que la classification, la détection d'objets, la reconnaissance de formes, etc.

Dans notre proposition, l'extraction de caractéristiques joue un rôle primordial. Elle vise à représenter sous une forme numérique tous les mots de notre corpus, résultants de la phase de segmentation, afin de les comparer et de les reconnaître efficacement. En se basant sur les caractéristiques extraites, notre approche permet deux principales actions. Premièrement, elle regroupe les mots similaires, qui présentent des caractéristiques similaires ou proches, en des classes homogènes, facilitant ainsi la recherche en réduisant le temps requis grâce à la comparaison avec un seul représentant sélectionné par classe. Deuxièmement, ces caractéristiques extraites sont utilisées pour la recherche du mot requête et la comparaison avec les mots du corpus. Grâce à cette approche, nous pouvons identifier les mots similaires ou identiques au mot recherché.

En effet, un des problèmes fondamentaux de l'analyse d'images est de déterminer quelles caractéristiques à employer pour avoir un bon résultat. D'après l'étude bibliographique que nous avons effectuée sur les caractéristiques, nous avons opté pour l'extraction de différents descripteurs afin de caractériser les images de mots de notre corpus. Ces descripteurs incluent le rapport d'aspect, la densité des pixels noirs, le nombre de minimas dans le profil de

projections verticales, le nombre de boucles, le nombre d'ascendants et de descendants (hampes et jambages), le nombre de composantes connexes, les moments de Hu, le nombre de points clés dans le mot, et les distances entre les points clés des SIFT (Scale-Invariant Feature Transform) au centre du mot..

Nous décrivons dans ce qui suit chacune de ces caractéristiques et sa méthode d'extraction.

### III.2.3.1. Le rapport d'aspect

Le rapport d'aspect (aspect ratio) est une caractéristique structurelle utilisée pour décrire les proportions d'une image ou d'un objet. Il est généralement calculé en divisant la hauteur de l'image par sa largeur ou l'inverse.

### III.2.3.2. La densité des pixels noirs

La densité des pixels noirs est une mesure qui permet de quantifier la quantité de pixels noirs présents dans une image. Elle peut être déterminée en calculant le ratio entre le nombre de pixels noirs et le nombre total de pixels dans l'image.

Pour calculer la densité des pixels noirs, on compte le nombre de pixels noirs dans l'image binaire de mot, et on divise ce nombre par le nombre total de pixels dans l'image. Ce dernier est égal à la hauteur de l'image multipliée par sa largeur.

Soit  $I$  l'image binaire de mot contenant deux niveaux (0: noir, et 255: blanc). Soient  $H$  la hauteur de cette image, et  $L$  sa largeur. La densité des pixels noirs de l'image  $I$  est notée  $D(I)$  et elle est calculé à l'aide de la formule suivante :

$$D(I) = \frac{\sum_{x=0}^{H-1} \sum_{y=0}^{L-1} \frac{I(x, y)}{255}}{H \times L}$$

### III.2.3.3. Le nombre de minimas dans le profil de projections verticales

Le nombre de minimas dans la projection verticale est une caractéristique structurelle utilisée pour analyser des images. La projection verticale consiste à projeter l'intensité des pixels d'une image sur un axe vertical pour obtenir un profil d'intensité.

Pour calculer le nombre de minimas dans la projection verticale, on identifie les points où l'intensité de la projection change de croissance positive à croissance négative. Ces points indiquent la présence de creux ou de changements dans le profil d'intensité verticale de l'image, comme le montre la figure suivante:

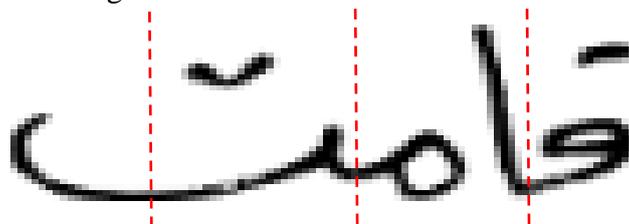


Figure III.5. Identification des minimas dans la projection verticale d'une image d'un mot

L'algorithme suivant récapitule les étapes d'identification et de calcul des minimas dans le profil de projections verticales:

Algorithme

Entrée: Image binaire de mot  $I$  de  $H$  lignes et  $L$  colonnes;

Sortie: le nombre de minimas ( $nbMin$ );

Utilisez: tableau ( $P$  de  $L$  éléments) pour stocker les valeurs de la projection verticale

Début

Pour  $i$  allant de 0 à  $L-1$  Faire

$P[i] \leftarrow 0$ ;

Pour  $j$  allant de 0 à  $L-1$  Faire

Pour  $i$  allant de 0 à  $H-1$  Faire

Si  $I(i,j)=255$  alors  $P[j] \leftarrow P[j]+1$ ;

$nbMin \leftarrow 0$ ;

Pour  $i$  allant de 1 à  $L-2$  Faire

Si  $P[i]<P[i-1]$  et  $P[i]<P[i+1]$  alors  $nbMin \leftarrow nbMin+1$ ;

Retourner  $nbMin$ ;

Fin.

#### **III.2.3.4. Le nombre de boucles**

Une boucle est formée par la présence de courbes ou de formes circulaires dans l'image du mot. Les boucles sont utiles pour l'identification de certains caractères c'est pourquoi nous avons décidé de les inclure parmi nos caractéristiques.

Le nombre de boucles dans une image peut être calculé en effectuant un processus de remplissage des trous et en soustrayant l'image originale de l'image remplie. Ensuite, nous comptons le nombre de composantes connexes dans l'image résultante, ce qui correspond au nombre de boucles.

Le remplissage de trous est une procédure qui consiste à combler les régions de l'image qui sont entourées par des pixels de couleur opposée. L'objectif est de rendre toutes les régions d'une même couleur connectées en éliminant les espaces vides à l'intérieur de ces régions.

En résumé, pour calculer le nombre de loupes dans une image on procède comme suit :

- 1- Effectuer un remplissage des trous dans l'image de mot en utilisant une technique appropriée.
- 2- Soustraire l'image originale de l'image remplie pour obtenir une image résultante.
- 3- Compter le nombre de composantes connexes dans l'image résultante. L'algorithme permettant l'extraction des composantes connexes va être présenté dans la section III.2.3.6.
- 4- Le nombre obtenu correspond au nombre de loupes dans l'image.

La figure suivante illustre le résultat de cette étape:



Figure III.6. Détection de boucles dans une image de mot

### III.2.3.5. Hampes et Jambages

Les ascendants et les descendants (ou hampes et jambages) sont des caractéristiques structurelles largement utilisées dans l'analyse de l'écriture, spécialement, manuscrite. Ils fournissent des informations importantes sur la forme et la structure des lettres.

Le nombre d'hampes et de jambages dans une image peut être déterminé en analysant les structures verticales présentes. Les hampes correspondent aux parties supérieures verticales des caractères ou des objets, tandis que les jambages correspondent aux parties inférieures verticales.

En effet, hampes et les jambages sont toujours en dehors de la zone médiane. Cette dernière est la région centrale où les traits principaux des lettres se situent, et elle est étroitement liée à la ligne de base. C'est pourquoi l'extraction donc des hampes et des jambages se fait dans notre proposition après la détection de la ligne de base et la localisation de la zone médiane.

Il est important de souligner que la détection de la ligne de base et la localisation de la zone médiane doivent être réalisées sur l'image de la ligne de texte entière plutôt que sur l'image de chaque mot individuellement. En considérant l'image de la ligne de texte dans son ensemble, nous sommes en mesure de prendre en compte le contexte global et d'obtenir une estimation plus précise de la ligne de base et de la zone médiane.

#### a) Détection de la ligne de base

La ligne de base est la ligne sur laquelle repose les caractères qui ne possèdent pas des descendants. La méthode la plus répandue pour détecter les lignes de base est la projection horizontale de l'image. La ligne de base correspond à la ligne dont la projection contient le plus grand nombre de pixels noirs (la ligne rouge dans la figure III.7).

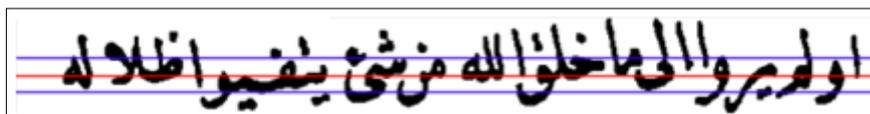


Figure III.7. Ligne de base et zone médiane détectées sur une ligne de texte

### ***b) Localisation de la zone médiane***

Dans notre application, nous détectons la zone médiane en traçons deux frontières, haute et basse par rapport à la ligne de base. La zone médiane sera donc l'espace compris entre ces deux frontières. Ces deux frontières

Dans notre approche, nous tranchons l'espace situé au-dessus de la ligne de base en trois parties égales pour déterminer la frontière supérieure de la zone médiane. Plus précisément, nous plaçons cette frontière à un tiers de la hauteur totale de l'espace au-dessus de la ligne de base. Même chose pour la frontière inférieure de la zone médiane.

Dans la figure III.8, la zone médiane est entre les deux lignes bleues.

### ***c) Extraction des composantes connexes***

Les extensions situées au-dessus de la zone médiane correspondent aux hampes, tandis que les extensions situées en-dessous de la zone médiane représentent les jambages.

Pour extraire les hampes, nous effectuons un étiquetage des composantes connexes dans la partie située au-dessus de la zone médiane. De manière similaire, pour extraire les jambages, nous procédons à un étiquetage des composantes connexes dans la partie située en-dessous de la zone médiane.

Le processus d'étiquetage de composantes connexes sera détaillé dans la section suivante.

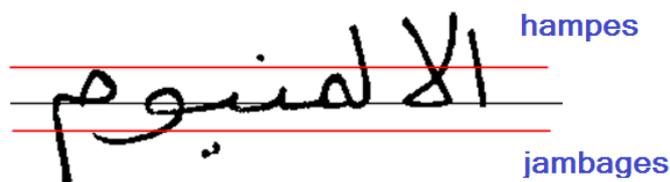


Figure III.8. Détection des hampes et jambages

### ***III.2.3.6. Nombre de composantes connexes***

En traitement d'images, une composante connexe fait référence à un ensemble de pixels adjacents qui partagent une propriété commune dans une image. Ces pixels sont généralement regroupés en fonction de leur similarité dans leurs valeurs d'intensité, de couleur ou d'autres caractéristiques.

L'identification des composantes connexes dans une image est souvent utilisée pour segmenter l'image en régions distinctes, ce qui permet d'extraire des objets ou des éléments d'intérêt de manière individuelle.

Dans notre application, le nombre de composantes connexes peut être utilisé comme une caractéristique discriminante pour différencier les mots différents qui ont des caractéristiques structurelles similaires. Par exemple, les deux mots de la figure ci-dessous ont le même nombre de boucles, d'hampes et de jambages mais ils se différencient par le nombre de leurs composantes connexes.

Figure III.9. Deux mots ayant le même nombre de boucles, hampes et jambages mais se différencient par le nombre de leurs composantes connexes.

Les algorithmes de détection de composantes connexes sont basés sur la connectivité spatiale des pixels. Les pixels adjacents dans les directions horizontale, verticale ou diagonale sont considérés comme connectés. Différentes techniques sont utilisées pour identifier les composantes connexes.

La méthode que nous utilisons pour l'extraction des composantes connexes est la méthode d'agrégation de pixel en version récursive. Cette méthode peut être décrite par l'algorithme suivant:

Algorithme

**Entrées :**  $I$ : l'image binaire de mot d'hauteur  $Haut$  et de largeur  $Larg$

**Sortie:** le nombre de composantes connexes ( $nbCC$ )

**Utilisez:**  $CC[]$ : liste contient les pixels d'une composante connexe

Début

$NbCC \leftarrow 0$ ;

Pour  $i$  allant de 0 à  $Haut$  faire

Pour  $j$  allant de 0 à  $Larg$  faire

Si  $I(i, j) = 0$  alors

$CC \leftarrow$  regrouper les pixels noirs voisins de  $(i, j)$  ;

$NbCC \leftarrow NbCC + 1$  ;

Fin Pour ;

Fin Pour ;

Retourner  $nbCC$ ;

Fin.

Cet algorithme identifie les pixels noirs non visités et les explore pour trouver tous les pixels adjacents noirs. Chaque exploration crée une nouvelle composante connexe. Le processus se répète jusqu'à ce que tous les pixels noirs aient été visités, et toutes les composantes connexes sont stockées dans une liste.

### III.2.3.7. Les moments de Hu

Nous avons mentionné dans le chapitre 2 que les moments invariants constituent des descripteurs de forme largement utilisés en traitement d'images. Parmi les moments invariants

les plus populaires, on trouve les moments de Hu. Ces derniers sont des valeurs numériques qui peuvent être utilisées comme caractéristiques pour la reconnaissance d'objets, la détection de formes, la comparaison d'images, etc. Les moments de Hu sont un ensemble de sept moments invariants qui captent des informations cruciales sur la forme et la symétrie d'un objet ou d'une région d'intérêt. Ils sont robustes aux variations de taille, d'orientation et de position de l'objet, ce qui les rend utiles dans de nombreuses applications de vision par ordinateur.

Ces moments sont calculés à partir des moments géométriques, qui sont des quantités statistiques calculées à partir des positions relatives des pixels de l'objet.

Les sept moments de Hu sont définis mathématiquement comme suit:

$$H1 = \eta_{20} + \eta_{02}$$

$$H2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$H3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$H4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$H5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$H6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$H7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

### **III.2.3.8. Les points-clés des SIFT**

L'algorithme SIFT (Standard Feature Transformation) est fréquemment utilisé dans le domaine de la vision par ordinateur pour détecter et décrire les points d'intérêt présents dans les images.

Dans notre approche, nous adoptons l'algorithme SIFT pour détecter les points-clés dans chaque image de mot en mettant l'accent sur la distance entre chaque point-clé et le centre de l'image de mot. Cette mesure de distance nous permet de caractériser la position relative des points-clés par rapport au centre, ce qui offre une information pertinente sur la répartition spatiale des caractéristiques dans l'image de mot. L'utilité de cette approche réside dans le fait qu'elle permet de différencier les mots en se basant sur la disposition spatiale de leurs caractéristiques. Les mots qui partagent des points-clés proches du centre de l'image de mot peuvent présenter des similarités structurelles ou des propriétés communes, tandis que les mots dont les points-clés sont plus éloignés peuvent indiquer des différences significatives dans leur apparence visuelle.

Ainsi, pour chaque mot, nous avons extrait 50 points clés en utilisant l'algorithme SIFT et calculé leur distance Euclidienne au centre de l'image du mot. Ces caractéristiques ont été spécifiquement conçues pour maintenir leur stabilité face aux variations de taille, de rotation et d'éclairage.

### **III.2.3.9. Nombre de points-clés de [03]**

A ces caractéristiques, nous avons ajouté une autre caractéristique qui ont prouvé leur efficacité dans un travail antérieur, à savoir: le nombre des points-clés de [03]

L'extraction de ces points clés passe par plusieurs étapes comme suit:

**a) Extraction des points clés**

Les points clés, également connus sous le nom de *keypoints* sont des points spéciaux dans une image qui présentent des caractéristiques distinctives et uniques. Ces points sont sélectionnés en fonction de leur capacité à représenter des informations importantes de l'image, tels que les coins, les bords ou les régions texturées.

Dans notre approche, les points clés sont de deux types, et ils sont extraits à partir du squelette de l'image:

- Les points terminaux: un point terminal est un pixel situé sur le squelette de l'image qui a seulement un voisin.
- Les points de branchement: un point de branchement est un pixel situé sur le squelette de l'image qui représente une intersection entre plusieurs traits. Un point de branchement a trois voisins dans le squelette de l'image.

Dans la figure suivante, les points terminaux sont en rouge alors que les points de branchement sont en vert.

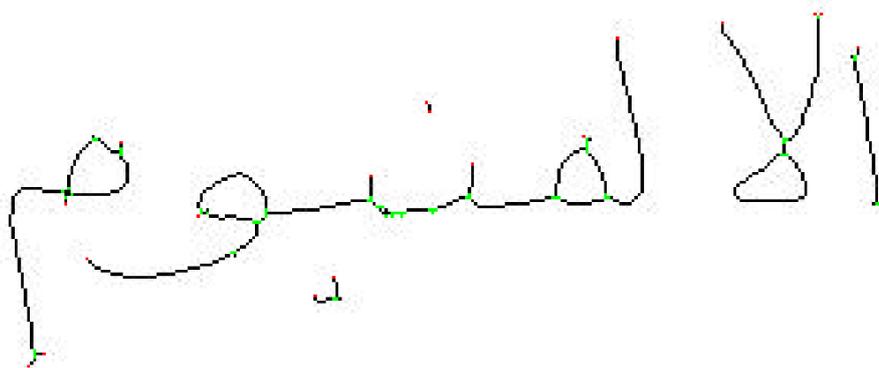


Figure III.10: Les points terminaux (rouge) et les points de branchements (vert) dans le squelette de l'image.

**b) Nettoyage des points-clés**

Lors de l'extraction des points clés, on obtient un grand nombre de points, dont certains peuvent être considérés comme du bruit ou ne pas contenir d'informations utiles. Par conséquent, il est nécessaire de procéder à un nettoyage ou à une sélection des points clés extraits afin de ne conserver que ceux qui sont pertinents.

Le nettoyage repose sur deux paramètres: la largeur moyenne du trait (ASW pour average stroke width) et la distance  $d$  dans le squelette. La distance  $D_s$  entre deux points-clés est le nombre de pixels qui les séparent dans l'image squelette. On suppose que la distance entre deux points-clés doit être supérieure à ASW pour être considérée valide et non bruitée.

**b.1) Estimation de la largeur moyenne du trait:**

Pour chaque image de mot de l'ensemble de données, le profil de projection verticale est calculé et les minima locaux sont extraits à partir du profil de projection. La moyenne de ces minima locaux a été considérée comme la largeur du trait moyenne (ASW).

**b.2) Nettoyage des points terminaux:**

La distance de squelette  $D_s$  est calculée entre chaque point terminal  $EP_i$  et tous les points-clés  $KP_j$  connectés à  $EP_i$  par des pixels du squelette. Les points terminaux ayant une distance  $D_s$  inférieure à ASW sont considérés comme inutiles et sont donc supprimés. Les autres points sont conservés.

L'ensemble des points terminaux après nettoyage est défini par l'équation suivante:

$$EP_{clean} = \{EP_i \in EP / D_s(EP_i, KP_j) > ASW, \forall KP_j \in KP\}$$

**b.2) Nettoyage des points de branchement**

Pour nettoyer les points de branchements bruts  $BP$ , les points terminaux nettoyés  $EP$  sont considérés comme des points de branchement bruts  $BP$ . Un poids  $w$  est associé à chaque  $BP_i$  dans  $BP$  qui a été considéré comme les distances cumulées à ses points-clés voisins. Si la distance entre deux points de branchement est inférieure à ASW ; le point de branchement ayant un poids minimal est supprimé. L'ensemble de points de branchement nettoyés  $BP_{clean}$  est défini par l'équation:

$$BP_{clean} = \{BP_i \in BP / D_s(BP_i, KP_j) \geq ASW, \forall KP_j \in (BP \cup EP_{clean})\}$$

**III.2.3.10. Enregistrement des vecteurs de caractéristiques**

A chaque mot de notre base d'image est associé un vecteur de 65 caractéristiques numériques. Cependant, une fois que les caractéristiques de tous les mots ont été extraites, nous les enregistrons dans un fichier au format .csv (Comma-Separated Values). C'est un format de fichier couramment utilisé pour représenter des données tabulaires. Le fichier .csv permet de conserver la structure des données avec chaque caractéristique représentée dans une colonne distincte et chaque image de mot dans une ligne. Un exemple de tel fichier est illustré dans la figure suivante:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
1	14	29	147.0	29.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.8599668422583801	0.5288845398631937	0.6470588235294118	1.0	0.6	0.0	0.444444444	0.4819277101	0.870772253	0.807173960	1.0	1.0
2	184	19	86.0	42.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.200493698344090	0.559480775111735	0.2352941176470588	0.5	0.5	0.0	0.222222222	0.433734939	0.164184667	0.078795954	0.018819762	0.0241
3	304	24	161.0	42.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.5902661625109119	0.13998862747292388	0.5882352941176471	0.25	0.6	0.0	0.666666666	0.289156626	0.482452303	0.373734466	0.062473518	0.0217
4	500	22	94.0	35.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.3397760276549079	0.39725734427964565	0.3529411764705882	0.0	0.6	0.0	0.555555555	0.192771084	0.255775091	0.164344466	0.066037148	0.0536
5	677	11	114.0	58.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.1825790808596898	0.0	0.2941176470588235	0.0	0.1	0.0	0.666666666	0.361445783	0.066101548	0.055435644	0.013289200	0.0059
6	130	87	137.0	54.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.3073246359241127	0.26510814580196	0.4117647058823529	1.0	0.5	0.0	0.777777777	0.746987951	0.216478337	0.135550668	0.053762676	0.0125
7	300	89	58.0	32.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.1491800862969715	0.595353318884079	0.0588235294117647	0.5	0.4	1.0	0.333333333	0.096385542	0.091953926	0.051763740	0.363512675	0.0
8	379	92	60.0	35.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.1277429097238270	0.5206745711829763	0.2352941176470588	0.25	0.3	0.0	0.111111111	0.120481927	0.101805340	0.045397043	0.008280363	0.0556
9	464	79	108.0	55.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.1821685379411808	0.2059082625243195	0.2352941176470588	0.5	0.3	0.0	0.666666666	0.337349397	0.091234020	0.062586127	0.040479820	0.0179
10	613	86	112.0	46.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.2850053155147506	0.2995674145672695	0.5882352941176471	0.25	0.7	0.0	0.777777777	0.530120481	0.183540001	0.116126788	0.177483070	0.0808
11	748	91	109.0	42.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.3200278852257414	0.2031390469415155	0.3529411764705882	0.25	0.4	0.0	0.444444444	0.289156626	0.216680755	0.144967312	0.177167570	0.1722
12	42	141	95.0	56.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.1238452293577060	0.1735581903943251	0.3529411764705882	0.25	0.4	0.0	0.555555555	0.240963855	0.047971612	0.036053853	0.094555083	0.0186
13	162	156	149.0	42.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.5279034730634385	0.2529624693257148	0.4705882352941176	0.5	0.6	0.0	0.444444444	0.457831325	0.448053840	0.332602591	0.015617046	0.01127
14	347	139	88.0	63.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.0584510398376145	0.206043592054338	0.3529411764705882	0.25	0.1	0.0	0.666666666	0.337349397	0.0	0.010136372	0.0	0.0275
15	458	154	177.0	43.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.6520245512915851	0.1695392596920648	0.5294117647058824	0.0	1.0	0.0	0.433734939	0.576593758	0.464637533	0.067277561	0.0413	
16	679	159	96.0	47.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.1993936606969839	0.2963820946608527	0.5882352941176471	0.25	0.6	0.0	0.555555555	0.240963855	0.117354400	0.070415402	0.077348851	0.0337
17	805	165	61.0	31.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.1830646647986589	0.5789566866131796	0.0588235294117647	0.5	0.2	0.0	0.222222222	0.120481927	0.143243146	0.071645027	0.378297210	0.0156
18	54	224	84.0	42.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.1901055991733002	0.3942316046343132	0.2941176470588235	0.5	0.3	0.0	0.333333333	0.361445783	0.131231915	0.069003775	0.030676529	0.0141
19	175	225	110.0	40.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.3538076507878150	0.38270244527687486	0.5294117647058824	0.75	0.6	0.0	0.777777777	0.385542168	0.448053840	0.137025150	0.049247869	0.0477
20	315	221	35.0	31.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.0	1.0	0.1176470588235294	0.0	0.2	0.0	0.222222222	0.024096385	0.046127431	0.0	0.084293990	0.2313
21	382	223	89.0	42.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.2160900476530124	0.2986616617016384	0.5882352941176471	0.25	0.5	0.0	0.666666666	0.228915662	0.124178388	0.076255571	0.073690232	0.0402
22	491	228	60.0	27.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.2386099277392951	0.761508163324051	0.1764705882352941	0.5	0.3	0.0	0.222222222	0.240963855	0.202745790	0.117693867	0.245027405	0.2705
23	597	221	217.0	38.0	C:\Users\Yaakoub\Documents\Sans titre.png	1.0	0.30522770251035497	1.0	0.75	1.0	0.0	0.777777777	1.0	1.0	1.0	0.709476835	0.6984
24	833	222	53.0	39.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.0501893028697930	0.4941108820838995	0.0	0.0	0.3	0.0	0.0	0.0	0.032798865	0.012690070	0.114476099	0.1126
25	136	290	88.0	32.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.3538076507878150	0.520731027773865	0.1176470588235294	0.5	0.5	1.0	0.333333333	0.108433740	0.272920460	0.182944054	0.426369868	0.7803
26	246	289	48.0	33.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.0710495715568078	0.7096939140460321	0.1764705882352941	0.25	0.2	0.0	0.111111111	0.072289156	0.055743825	0.016293938	0.088792683	0.2057
27	315	294	61.0	34.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.1451677926384802	0.46389520232429843	0.1176470588235294	0.5	0.0	0.0	0.222222222	0.216867489	0.081420850	0.044411186	0.168188308	0.0625
28	408	281	98.0	54.0	C:\Users\Yaakoub\Documents\Sans titre.png	0.1496853362817246	0.21292326878158148	0.6470588235294118	0.25	0.2	0.0	0.222222222	0.385542168	0.074837702	0.048013264	0.002725100	0.0830

Figure III.11. Exemple du fichier contenant les caractéristiques de tous les mots

### III.2.4. Regroupement des mots similaires en utilisant l'algorithme K-means

Une fois que les caractéristiques de mots sont extraites et stockées dans un fichier .csv, l'étape suivante est d'effectuer un regroupement ou un clustering pour de regrouper les mots similaires ensemble afin de faciliter la recherche et la reconnaissance ultérieure.

L'utilité de cette phase (clustering) est lorsque nous avons un grand nombre de mots à traiter, il peut être inefficace de comparer chaque mot avec tous les autres pour trouver des correspondances. Le clustering permet de regrouper les mots similaires, c'est-à-dire ceux qui partagent des caractéristiques communes, dans des groupes homogènes. Cela réduit considérablement l'espace de recherche et accélère le processus de recherche lorsqu'un mot spécifique est recherché.

En effet, plusieurs méthodes de clustering existent et chacune d'entre elles possèdent ses propres caractéristiques, avantages et limites. Dans notre approche nous avons opté pour l'utilisation de l'algorithme K-means [79].

#### III.2.4.1. L'algorithme Kmeans

K-means est un algorithme d'apprentissage non supervisé utilisé pour regrouper un ensemble de données en un nombre prédéfini de clusters ou de groupes homogènes. L'objectif de l'algorithme K-means est de minimiser la variance intra-cluster, c'est-à-dire la somme des distances entre les données d'un cluster et leur centroïde (point central du cluster).

En utilisant l'algorithme K-means, nous pouvons regrouper les mots ayant des caractéristiques similaires dans des clusters, ce qui facilite l'analyse et l'interprétation des données.

Voici comment fonctionne l'algorithme K-means :

- 1- Spécifiez le nombre de clusters ( $K$ ) que vous souhaitez former.
- 2- Sélectionnez  $K$  points aléatoires comme centres initiaux de chaque cluster, appelés centroïdes.

- 3- Assignez chaque point de données au cluster dont le centroïde est le plus proche, en utilisant une mesure de distance telle que la distance euclidienne.
- 4- Recalculez les centroïdes de chaque cluster en prenant la moyenne des positions des points du cluster.
- 5- Répétez les étapes 3 et 4 jusqu'à ce que les centroïdes convergent vers des positions stables ou que le nombre d'itérations prédéfini soit atteint.
- 6- Les points de données sont maintenant regroupés en  $K$  clusters, où chaque point appartient à un seul cluster.

Notons que l'algorithme K-means est sensible au choix initial des centroïdes, car il peut converger vers un minimum local. Pour atténuer ce problème, des techniques comme la répétition multiple de l'algorithme avec différents choix initiaux de centroïdes ou l'utilisation d'algorithmes d'initialisation plus avancés peuvent être utilisées.

Le choix d'utiliser l'algorithme K-means repose sur plusieurs justifications. Tout d'abord, K-means est un algorithme largement utilisé et bien établi dans le domaine du clustering, ce qui témoigne de son efficacité et de sa pertinence. Sa simplicité de mise en œuvre nous permettra de l'intégrer facilement dans notre pipeline de traitement de mots. De plus, étant donné que nous traitons un grand nombre de mots dans notre corpus, la scalabilité de K-means est un avantage crucial, nous permettant de traiter efficacement de grandes quantités de données.

Un autre avantage de K-means est son interprétabilité. Les clusters résultants de K-means sont facilement interprétables, ce qui nous permettra d'analyser les mots regroupés dans chaque cluster et de tirer des conclusions significatives. De plus, notre choix d'utiliser des descripteurs numériques pour représenter les mots s'aligne parfaitement avec les caractéristiques adaptées de K-means pour travailler avec des données continues et numériques.

#### **III.2.4.2. Résultats du clustering**

A la sortie de cette phase, nous avons créé un ensemble de catégories ou clusters regroupant des mots ayant des caractéristiques similaires et étroitement liées. Chaque catégorie est composée de mots qui partagent des similitudes importantes entre eux.

L'idée principale est de regrouper les mots en fonction de leurs caractéristiques communes afin de former des catégories cohérentes. Ces caractéristiques peuvent inclure des aspects sémantiques, syntaxiques, thématiques ou d'autres propriétés spécifiques aux mots.

Le regroupement des mots en catégories présente plusieurs avantages. Il facilite l'organisation et la compréhension des mots en créant des ensembles distincts où les mots ont des liens étroits les uns avec les autres. Cela permet également de simplifier les tâches d'analyse et de traitement ultérieurs en se concentrant sur des groupes spécifiques de mots plutôt que sur l'ensemble du vocabulaire.

Il est important de noter que les catégories doivent être formées de manière à ce que les mots à l'intérieur d'une même catégorie soient très similaires, tandis que les mots appartenant à des catégories différentes présentent des différences significatives. Cela garantit une distinction claire entre les groupes de mots et une cohérence interne au sein de chaque catégorie.

A la fin de cette étape, les clusters, et leurs mots respectifs accompagnés de leurs caractéristiques extraites et de leurs propriétés (chemin de l'image qui le contient, sa position dans l'image, etc.) sont enregistrés dans une base de données.

### **III.2.5. La recherche**

Dans cette phase, nous exploitons les résultats obtenus par les premières phases notamment les caractéristiques et les clusters de mots pour répondre à une requête de l'utilisateur. Le système répond à la requête en retournant un ensemble d'images jugées pertinentes pour l'utilisateur. Un document est jugé pertinent pour l'utilisateur s'il contient la requête cherchée ou bien s'il contient des mots proches de la requête.

L'utilisateur commence par la formulation de sa requête qui subira les mêmes traitements que les mots de la base, et le système procède une recherche par appariement des caractéristiques.

#### ***III.2.5.1. Saisie du mot et extraction des caractéristiques***

Dans notre approche, la requête est formulé sous forme d'une image d'un ou plusieurs mot. L'image introduite va subir les mêmes étapes de traitement appliquées sur les mots de la base et qui ont été décrites précédemment.

Ces étapes comprennent la conversion de l'image en niveaux de gris, la binarisation, la segmentation, pour finalement pouvoir extraire ses caractéristiques. L'extraction des caractéristiques à partir de la requête est essentielle pour représenter la requête de manière appropriée et discriminative par des descripteurs numériques. Ces caractéristiques servent de base pour la recherche et la comparaison avec d'autres mots dans la base de données.

Les caractéristiques considérées ici sont les mêmes celles extraites à partir des images de mots de notre base d'images, à savoir: le rapport d'aspect, la densité des pixels noirs, le nombre de minimas dans le profil de projections verticales, le nombre de boucles, le nombre d'hampes et de jambages, le nombre de composantes connexes, les moments de Hu, le nombre de points clés dans le mot, et les distances des points-clés du SIFT au centre de l'image.,

#### ***III.2.5.2. Appariement des caractéristiques***

Une fois les caractéristiques de la requête sont extraites, nous procédons à la recherche des documents comportant cette requête. La recherche se traduit par une comparaison des caractéristiques de la requête avec les caractéristiques des mots de la base.

En effet, la comparaison avec les caractéristiques de tous les mots de la base de données peut être fastidieuse et très coûteuse en termes de temps, surtout lorsque la taille de la base est élevée. C'est là que l'utilité du clustering que nous avons effectué lors d'une phase antérieure se manifeste. Ce clustering en utilisant l'algorithme K-means a regroupé les mots similaires dans des clusters ou classes. Étant donné que les mots d'un même cluster partagent les mêmes caractéristiques (ou des caractéristiques similaires), il n'est pas nécessaire de comparer les caractéristiques de la requête avec celles de tous les mots de la base de données. Il suffit de les comparer avec un représentant de chaque classe, ou avec les centroïdes des classes. L'objectif est d'identifier la classe la plus proche de la requête introduite par l'utilisateur.

L'algorithme K-means a été appliqué précédemment pour regrouper les mots en différentes classe. Chaque classe est représentée par son centroïde, qui est essentiellement le point central de la classe, déterminé par les caractéristiques moyennes des mots de cette classe.

Pour déterminer quelle classe ou cluster est le plus proche de la requête introduite par l'utilisateur, nous calculons une mesure de similarité entre les caractéristiques du mot et les centroïdes de chaque classe. Cette mesure de similarité peut être basée sur différentes techniques, telles que la distance euclidienne ou la similarité cosinus. Dans notre approche nous avons utilisé la distance de Manhattan.

La distance de Manhattan est une mesure de distance utilisée pour calculer la distance entre deux points dans un espace bidimensionnel. Elle est définie comme la somme des différences absolues des coordonnées  $x$  et  $y$  des deux points. Ainsi, pour deux points  $A(x_1, y_1)$  et  $B(x_2, y_2)$  la distance de Manhattan est calculée comme suivant :

$$d = |x_2 - x_1| + |y_2 - y_1|$$

En comparant les caractéristiques du mot avec les centroïdes, nous identifions la classe dont le centroïde est le plus proche du mot. La proximité est déterminée par la mesure de similarité calculée. Plus la similarité est élevée, plus le mot est considéré comme proche de la classe correspondante.

Une fois la classe la plus proche identifié, il est possible d'attribuer ce mot à cette classe. Cela permet de classer le mot dans la classe qui partage des caractéristiques similaires.

Le processus de recherche peut être récapitulé par le pseudo-code suivant:

Algorithme

**Entrée:** la requête ( $R$ ) et les mots de l'ensemble de données ( $Ens$ )

**Sortie:** liste de documents pertinents ( $LP$ )

**Variables utilisées:** *Seuil*

Début

Fixer une valeur du *Seuil* ;

$C_R \leftarrow$  vecteur de caractéristique de la requête;

Pour chaque mot  $i$  de l'ensemble de données Faire

Début

$C_i \leftarrow$  vecteur de caractéristiques de l'image de mot  $i$ ;

$D \leftarrow$  distance de Manhattan entre  $C_R$  et  $C_i$ ;

Si  $D < \text{seuil}$  alors ajouter le document contenant le mot  $i$  à  $LP$

Fin;

Retourner *LP*;

Fin.

### **III.2.5.3. Affichage des résultats**

Une fois que nous avons identifié la classe à laquelle appartient la requête, nous procédons à l'étape finale qui est l'affichage des résultats à l'utilisateur.

Ainsi, tous les mots appartenant à la classe la plus proche de la requête sont considérés pertinents et doivent être affichés à l'utilisateur.

Pour ce faire, nous accédons à l'entrée de cette classe spécifique dans la base de données et nous récupérons les informations de ses mots telles que les coordonnées *x* et *y*, la longueur, la largeur et le chemin de l'image associée à chaque mot appartenant à cette classe.

En utilisant ces informations, nous encadrons les mots de cette classe qui sont similaires à la requête et nous les affichons à l'utilisateur. Cela signifie que nous dessinons un cadre autour de ces mots dans l'image pour les mettre en évidence et faciliter leur identification.

## **III. 3. Conclusion**

En conclusion, ce chapitre présente notre approche proposée pour l'appariement de mots dans les images de document. L'approche proposée se compose de plusieurs étapes de traitement afin d'atteindre les objectifs visés. Elle inclut ainsi des étapes de prétraitement (transformation en niveaux de gris, et binarisation), de segmentation en lignes et en mots, et d'extraction de caractéristiques, de clustreing, et de recherche.

Différentes caractéristiques, telles que le rapport d'aspect, la densité des pixels noirs, le nombre de minimas dans la projection verticale, le nombre de boucles, les hampes et les jambages, ainsi que le nombre de composantes connexes, sont extraites pour chaque mot.

Le regroupement des mots est réalisé à l'aide de l'algorithme K-means; Les mots sont regroupés en fonction de la similarité de leurs caractéristiques.

Dans la phase de recherche, la saisie du mot et l'extraction de ses caractéristiques sont effectuées. La classification finale repose sur la similarité des caractéristiques avec les centroïdes des clusters obtenus de la phase précédente.

L'approche proposée permet d'identifier et de mettre en évidence les mots similaires grâce à l'analyse d'images et à la classification par catégories. Cela ouvre des possibilités pour une meilleure compréhension et interprétation des données textuelles à partir d'images.

En somme, ce chapitre présente une méthodologie complète et prometteuse pour l'analyse de mots dans les images de documents.

# Chapitre IV. Implémentation et résultats

---

## IV. 1. Introduction

Dans ce chapitre, nous allons plonger plus en profondeur dans la réalisation pratique de notre système. Après avoir passé du temps à planifier et à concevoir notre application, il est maintenant temps de mettre nos idées en action.

Pour commencer, nous allons présenter les ressources et les outils de développement que nous avons utilisés tout au long du processus d'implémentation. Cela peut inclure des langages de programmation spécifiques, des frameworks, des bibliothèques ou des environnements de développement intégrés (IDE). Nous discuterons des raisons pour lesquelles nous avons choisi ces outils et de la manière dont ils ont contribué à simplifier le développement de notre application.

Après, nous mettrons l'accent sur les interfaces principales de notre application. Nous présenterons les différentes parties de l'interface utilisateur que nous avons conçu, en expliquant les choix de conception, les fonctionnalités implémentées.

Nous présentons finalement quelques résultats des tests et nous discutons ces résultats tout en essayant d'expliquer les raisons des bonnes et mauvaises performances.

## IV. 2. Outils de développement

### IV.2.1. Langage et environnement de développement

Notre application a été développée en langage de programmation Java, avec l'environnement de développement EclipseIDE Développer - 2022-12 et la Version: 2.3.1200.v20221123-1800.

#### IV.2.1.1. Java

Java est un langage de programmation populaire et polyvalent, largement utilisé pour le développement d'applications logicielles, de sites web et d'applications mobiles. Il a été créé par Sun Microsystems (maintenant une partie d'Oracle) et a été publié pour la première fois en 1995.

Java est un langage orienté objet, ce qui signifie qu'il organise les fonctionnalités dans des objets autonomes qui interagissent les uns avec les autres pour créer des programmes. Il est également portable, ce qui signifie qu'il peut être exécuté sur différents systèmes d'exploitation sans nécessiter de modifications majeures. Cela est rendu possible grâce à la machine virtuelle Java (JVM) qui permet d'exécuter le code Java de manière indépendante de la plateforme.

En outre, Java possède une vaste bibliothèque standard, appelée "Java Standard Edition" (Java SE), qui fournit de nombreuses classes et méthodes prédéfinies pour faciliter le développement de logiciels. Il existe également de nombreuses bibliothèques et frameworks

tiers qui étendent les fonctionnalités de base de Java et simplifient le processus de développement.

#### **IV.2.1.2. Eclipse**

Eclipse, qui est un environnement de développement intégré (IDE) populaire utilisé principalement pour la programmation en Java, bien qu'il puisse également prendre en charge d'autres langages de programmation tels que C/C++, Python, PHP, etc.

Eclipse fournit un ensemble d'outils et de fonctionnalités puissants pour faciliter le développement de logiciels. Il offre un éditeur de code avancé avec des fonctionnalités telles que la coloration syntaxique, la complétion automatique, la navigation dans le code et le débogage. L'IDE propose également un système de gestion de projet qui permet d'organiser et de gérer les fichiers sources, les bibliothèques et les ressources nécessaires à un projet.

Une caractéristique clé d'Eclipse est sa modularité. Il est basé sur un framework extensible appelé Eclipse Platform, qui permet aux développeurs d'ajouter des plugins et des extensions pour personnaliser l'IDE en fonction de leurs besoins spécifiques. Cela signifie que vous pouvez étendre les fonctionnalités d'Eclipse en ajoutant des plugins pour prendre en charge des langages de programmation supplémentaires, des outils de gestion de version, des frameworks de test, etc.

#### **IV.2.2. Bibliothèques utilisées**

Pour le développement de notre application, nous avons utilisé les bibliothèques suivantes :

##### **IV.2.2.1. Jtessboxeditor**

Jtessboxeditor est un outil pratique pour l'entraînement et l'amélioration des modèles OCR utilisant Tesseract OCR. Il offre une interface utilisateur conviviale pour éditer les fichiers de boîtes de caractères, ajuster les boîtes autour des caractères, et générer les fichiers de données d'entraînement nécessaires. Jtessboxeditor est largement utilisé dans le domaine de la reconnaissance optique de caractères pour améliorer la précision et l'efficacité de Tesseract OCR.

##### **IV.2.2.2. Weka**

La bibliothèque Weka en Java offre aux développeurs un ensemble d'outils puissants pour l'apprentissage automatique, leur permettant de manipuler les données, d'appliquer des algorithmes d'apprentissage automatique et d'évaluer les performances des modèles. Elle est largement utilisée dans la communauté de l'apprentissage automatique pour la recherche, l'éducation et le développement de solutions pratiques.

##### **IV.2.2.3. OpenCV**

La bibliothèque OpenCV (Open Source Computer Vision Library) est une bibliothèque de vision par ordinateur puissante et polyvalente, offrant des fonctionnalités complètes pour le traitement d'images et de vidéos, la détection d'objets, la vision par ordinateur et l'apprentissage automatique. Elle est largement utilisée dans divers domaines tels que la robotique, la surveillance, l'automatisation industrielle, la réalité augmentée, la reconnaissance faciale, etc.

#### IV.2.2.4. *JFreeChart*

JFreeChart est une bibliothèque Java puissante et flexible pour la création de graphiques et de visualisations de données. Elle offre une variété de types de graphiques, des options de personnalisation, un support de données polyvalent et une intégration facile avec d'autres bibliothèques. JFreeChart est largement utilisée dans le développement d'applications de visualisation de données, les rapports, les tableaux de bord et d'autres domaines où la représentation graphique des données est nécessaire.

### IV. 3. Présentation de l'application

Notre application est divisée en 3 sections ou interfaces principales:

- La première section, "*Portail de l'Application*". C'est l'interface principale de notre application. Elle comprend deux boutons qui permettent de choisir entre l'espace d'indexation et l'espace de recherche.
- La deuxième section (*Indexation*) est chargée du traitement initial, comprenant le prétraitement, la segmentation, l'extraction de caractéristiques et le regroupement des mots similaires.
- La troisième section (*Recherche*) est dédiée à l'interaction avec l'utilisateur. Celui-ci peut entrer un mot sous forme d'image ou de texte, puis le processus de recherche est effectué pour trouver des correspondances.

Nous décrivons dans ce qui suit chacune de ces interfaces tout en déroulant un scénario d'utilisation complet, qui explique le fonctionnement de notre système.

#### IV.3.1. Interface de "Portail de l'Application"

Cette interface (figure IV.1) présente deux boutons qui offrent la possibilité de sélectionner l'une des deux options disponibles : accéder à l'interface de recherche ou accéder à l'interface d'indexation.



Figure IV.1. Interface principale de notre application

### IV.3.2. Interface de l'indexation

Cette interface se compose de deux parties (voir figure IV.2). À gauche, se trouvent des onglets qui représentent les résultats de chaque étape de traitement. À droite se trouvent les boutons de traitement. Un seul bouton peut être activé à la fois.

Lorsqu'un bouton est enfoncé, il exécute l'action correspondante et active un autre bouton. Les boutons présents dans cette interface sont les suivants : "*Ouvrir Image*", "*Niveau de gris*", "*Binarisation*", "*Segmentation*", "*Extraction de Caractéristiques*", "*Résultat de K-means*" et "*Classification de Mots*".

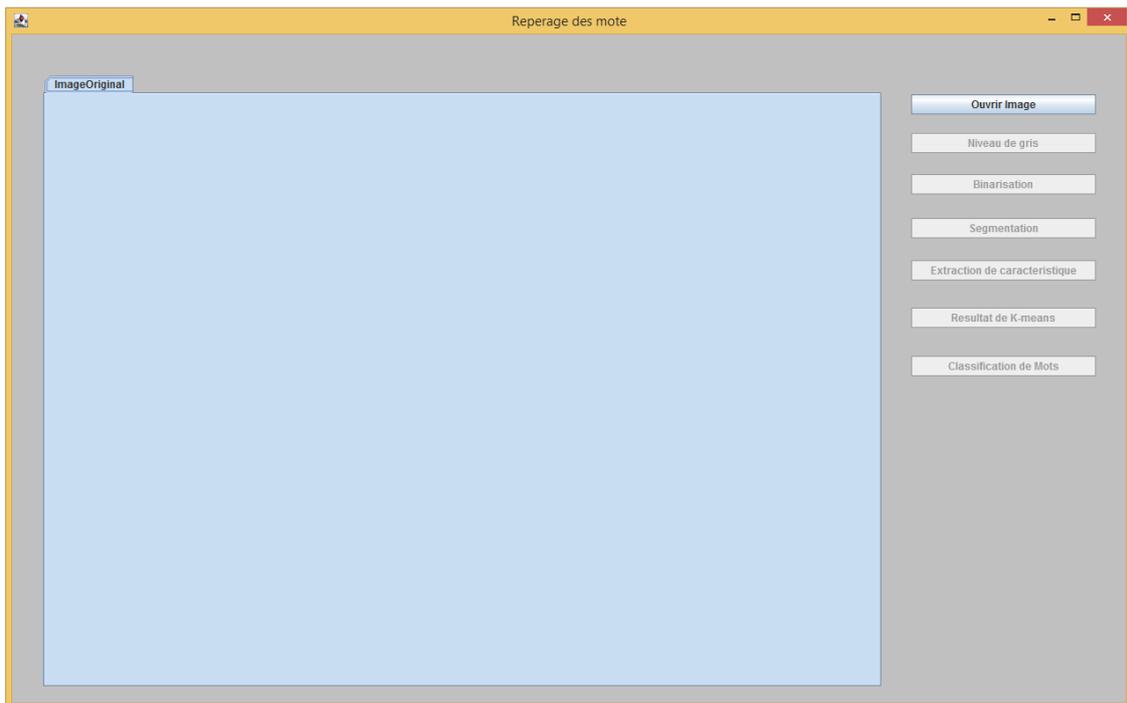


Figure IV.2. Interface de l'Indexation

Dans la suite de notre démonstration nous allons utiliser l'image suivante comme un exemple de test.

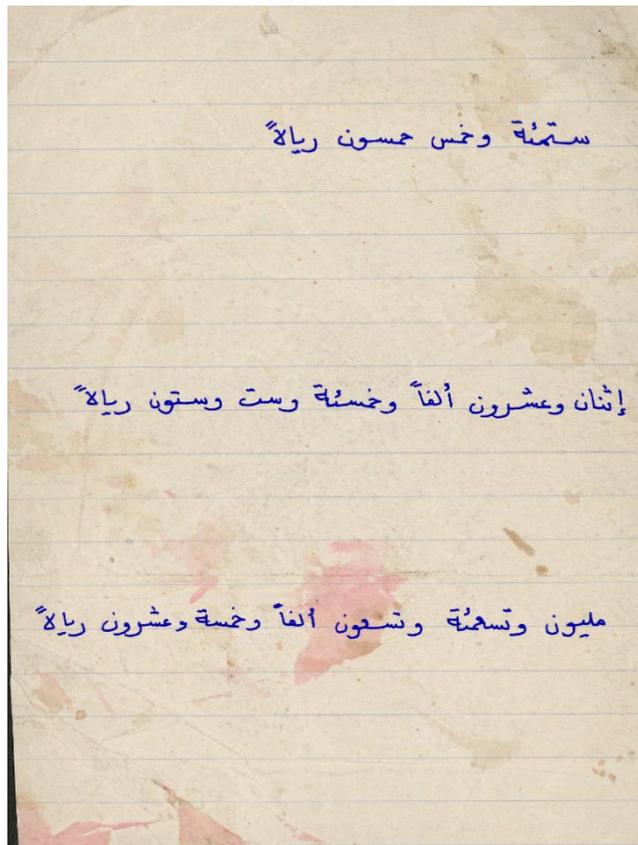


Figure IV.3. Exemple d'une image de test.

#### IV.3.2.1. Chargement d'images

L'application commence par importer une image de texte. Pour ce faire, nous cliquons sur le bouton "Ouvrir image". Une boîte de dialogue s'affiche, nous permettant de sélectionner l'image de texte à traiter. Veuillez voir la figure suivante:

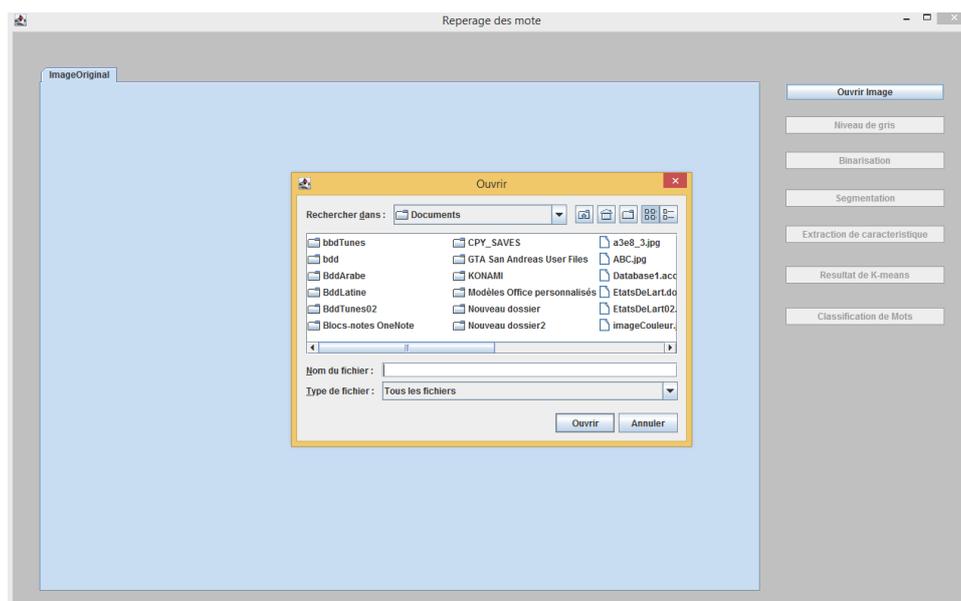


Figure IV.4. Chargement d'une image de texte.

En cliquant sur « Ouvrir image », l'image choisie s'affiche donc sur le panneau du premier onglet comme le montre la figure suivante :

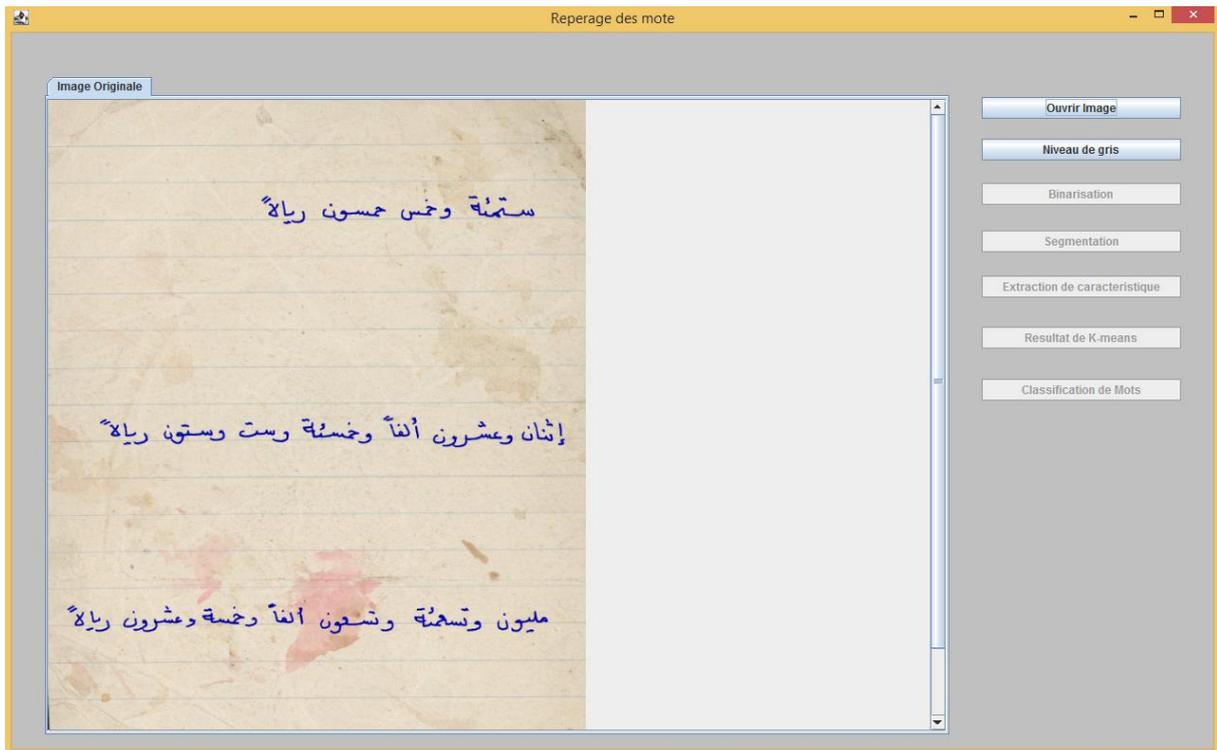


Figure IV.5. Affichage de l'image chargée sur le panneau d'affichage.

#### IV.3.2.2. Transformation en niveaux de gris

En cliquant sur le bouton "Niveaux de gris", l'image subit une transformation en niveau de gris. L'image résultante est ensuite affichée dans la fenêtre principale, dans un nouvel onglet, comme illustré ci-dessous.

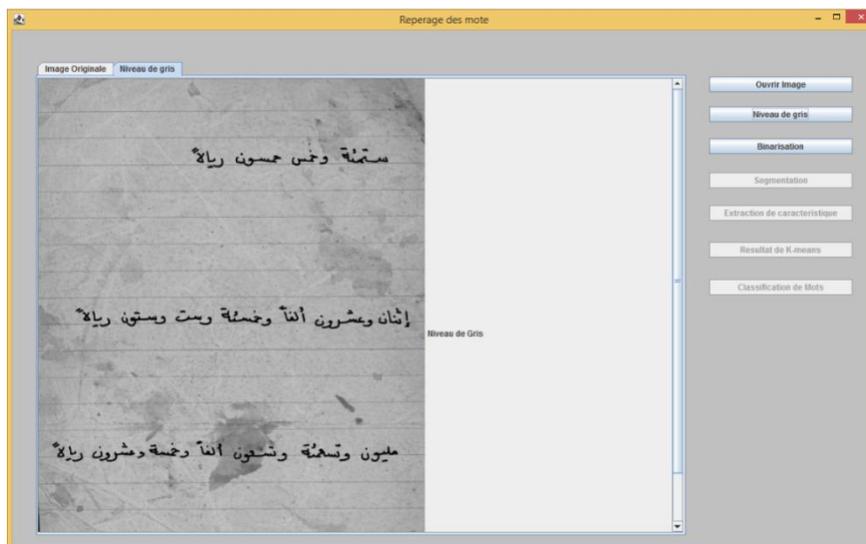


Figure IV.6. Affichage de l'image au niveau de gris dans un nouvel onglet.

### IV.3.2.3. Binarisation

La binarisation est assurée par le bouton "Binarisation". L'image résultante s'affiche sur la fenêtre principale dans un nouvel onglet comme suit :

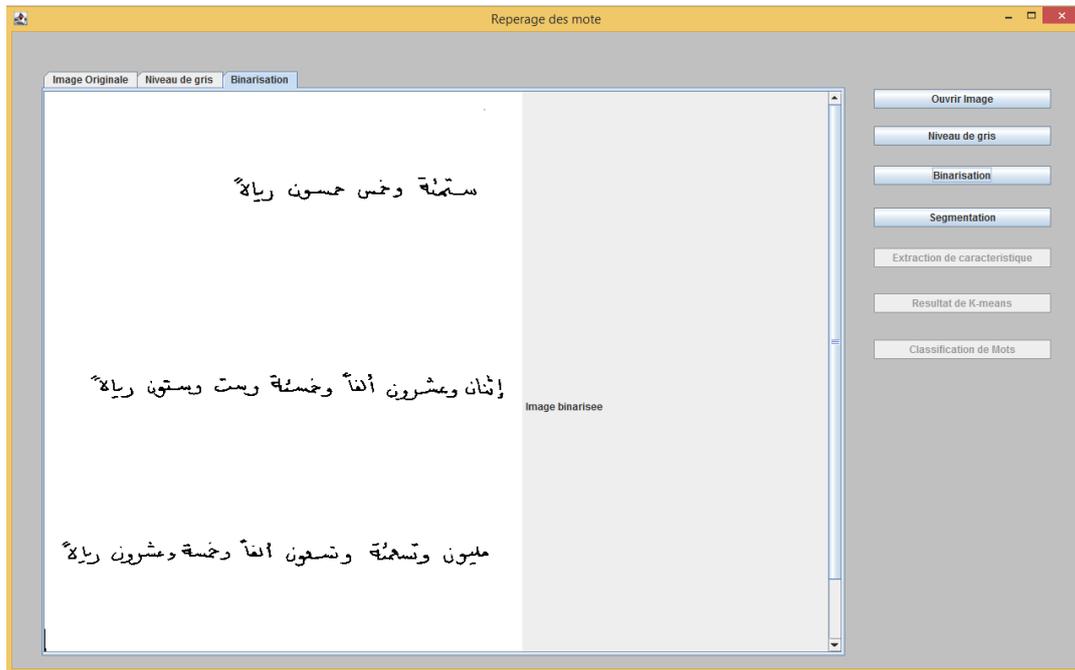


Figure IV.7. Affichage de l'image binaire dans un nouvel onglet.

### IV.3.2.4. Segmentation

À cette étape, l'image subit une segmentation en lignes et en mots simultanément. En cliquant sur le bouton "Segmentation", l'image contenant le résultat de segmentation en lignes et en mots, est affichée dans la fenêtre principale, dans un nouvel onglet dédié à cette étape de traitement, comme illustré dans la figure suivante:

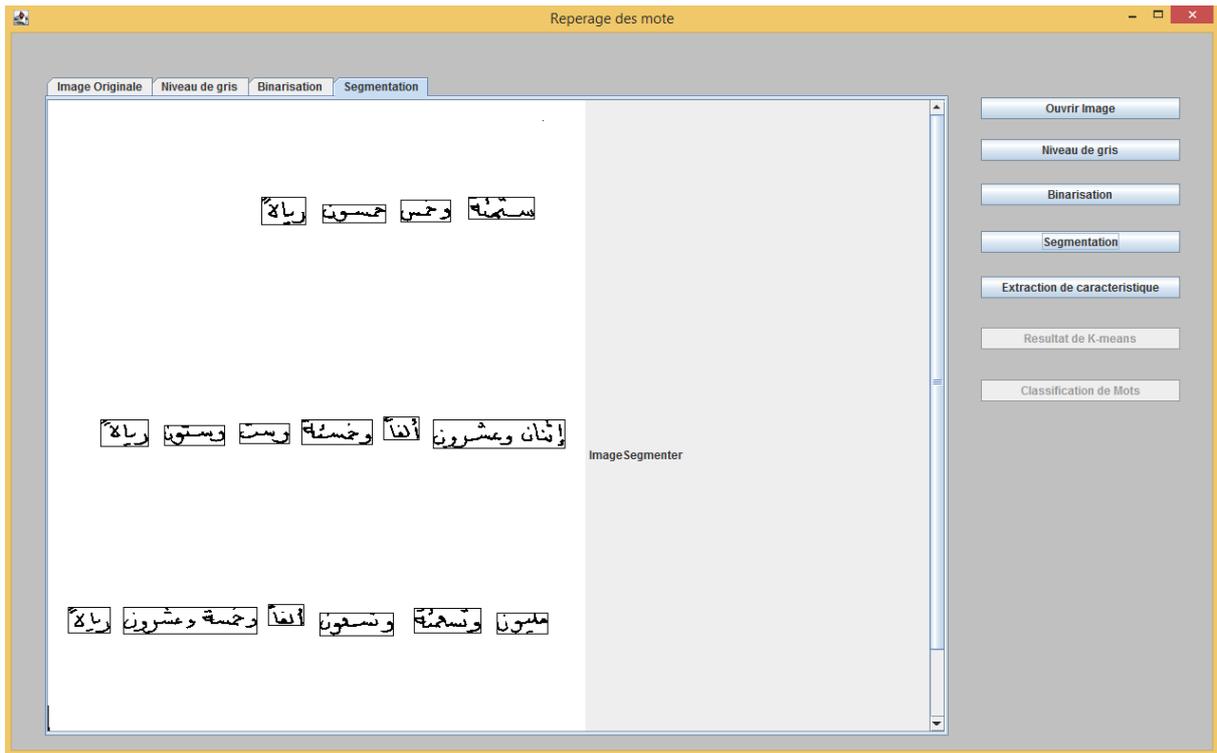


Figure IV.8. Affichage de l'image segmenter dans un nouvel onglet.

#### IV.3.2.5. Extraire des caractéristiques

Lorsque le bouton "Extraire les caractéristiques" est activé, l'outil analyse l'image et extrait ses caractéristiques décrites dans le chapitre précédent. Ces caractéristiques extraites sont ensuite utilisées pour représenter l'image de manière abstraite. Les résultats de cette extraction sont affichés dans un nouvel onglet de la fenêtre Indexation (figure suivante).

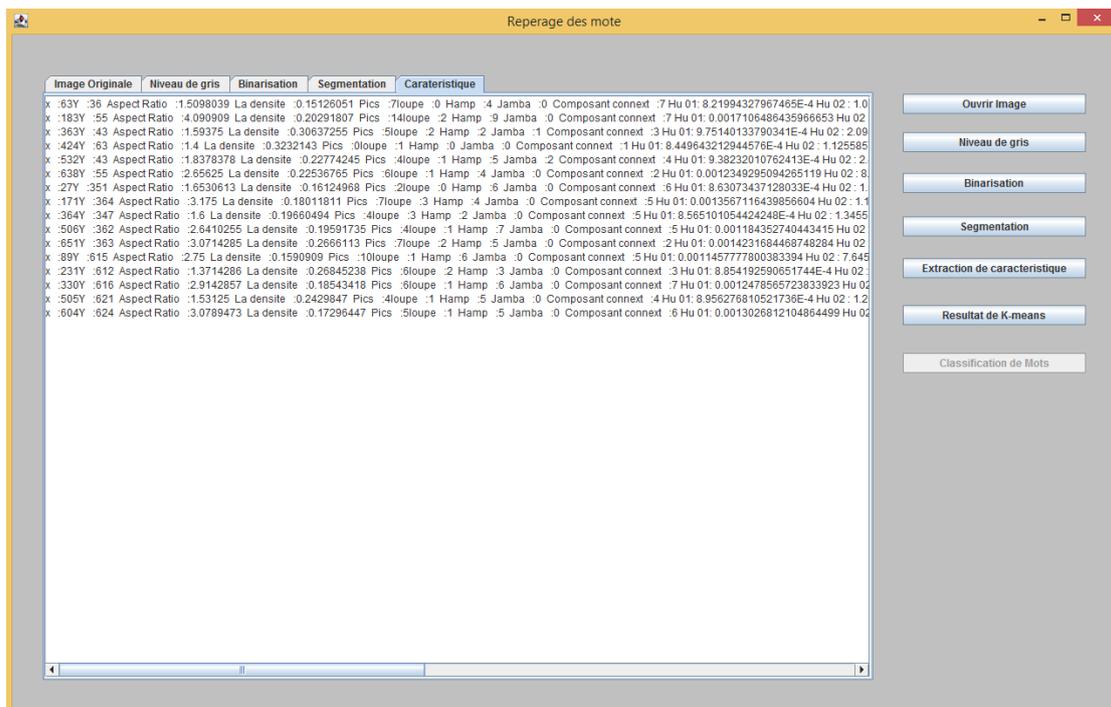


Figure IV.9. Résultat d'extraction des caractéristiques affiché dans un nouvel onglet

### IV.3.2.6. Clustering en utilisant K-means

Lorsqu'on clique sur le bouton "Résultat K-means", l'outil applique l'algorithme K-means dans le but de regrouper les mots de l'image en différents clusters en fonction de leurs similarités de caractéristiques. Les résultats de regroupement, ainsi que les centroïdes correspondants, sont affichés dans un nouvel onglet de la fenêtre indexation (figure suivante).

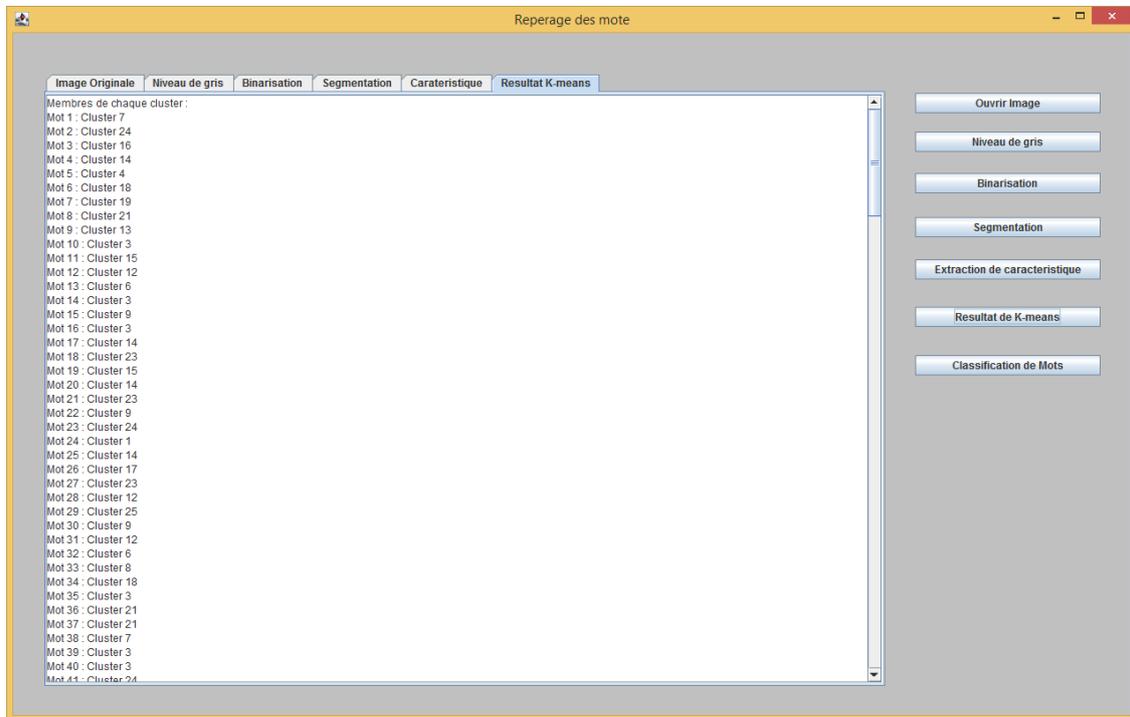


Figure IV.10. Résultats de K-means affichés dans un nouvel onglet

### IV.3.2.7. Affichage des groupes de mots

Lorsqu'on clique sur le bouton "Classification des mots", l'outil affiche dans un nouvel onglet les groupes des mots obtenus du clustering Kmeans comme le montre la figure suivante:



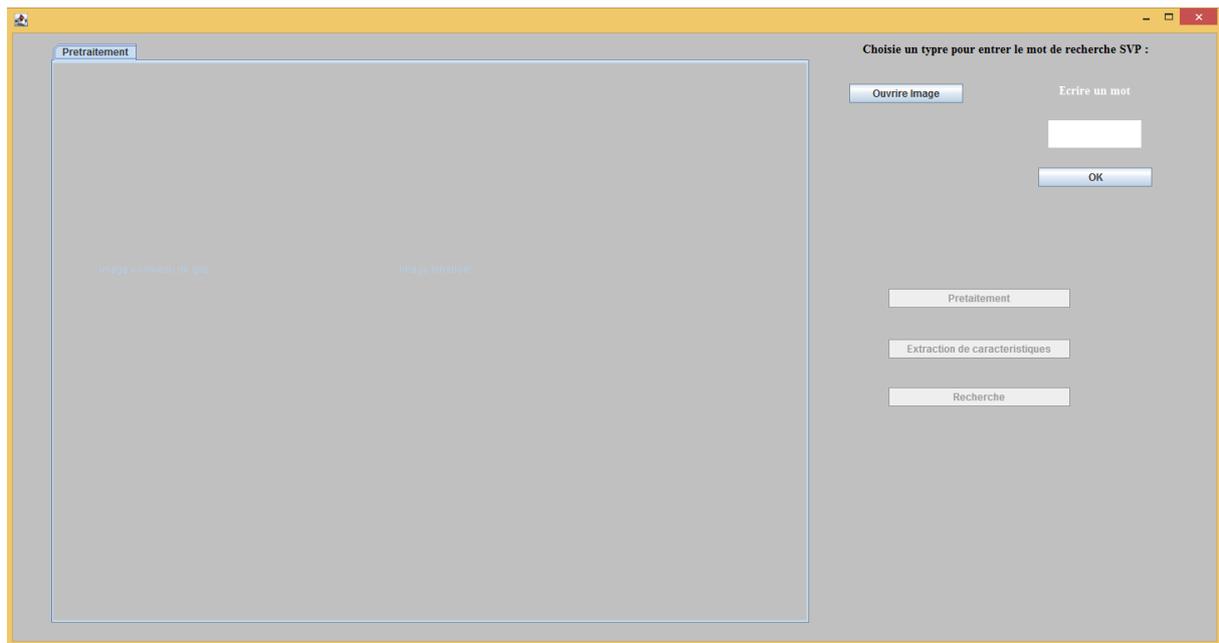


Figure IV.12. interface de recherche

#### IV.3.3.1. Charger la requête

Le bouton "Ouvrir l'image" offre à l'utilisateur la possibilité de sélectionner une requête sous forme d'image qu'il souhaite rechercher. L'image choisie s'affiche ensuite sous le bouton, permettant à l'utilisateur de visualiser la requête sélectionnée comme illustré ci-dessous.

En revanche, si l'utilisateur préfère saisir le mot en tant que texte, il lui suffit de le taper dans une zone de texte située à droite du bouton.

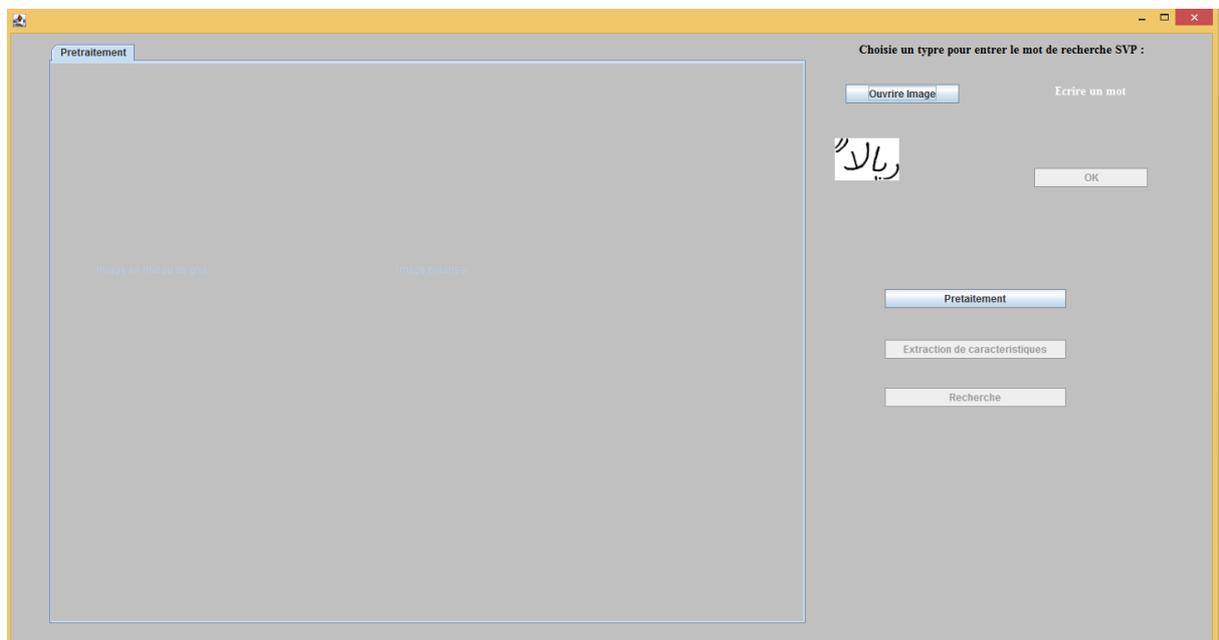


Figure IV.13. Chargement d'une requête sous forme d'image de mot

### IV.3.3.2. Extraire les caractéristiques

Lorsque le bouton "Extraire les caractéristiques" est cliqué, l'outil analyse l'image et extrait ses caractéristiques. Les résultats de cette extraction sont affichés dans un nouvel onglet de la fenêtre "Recherche". Voir la figure suivante:

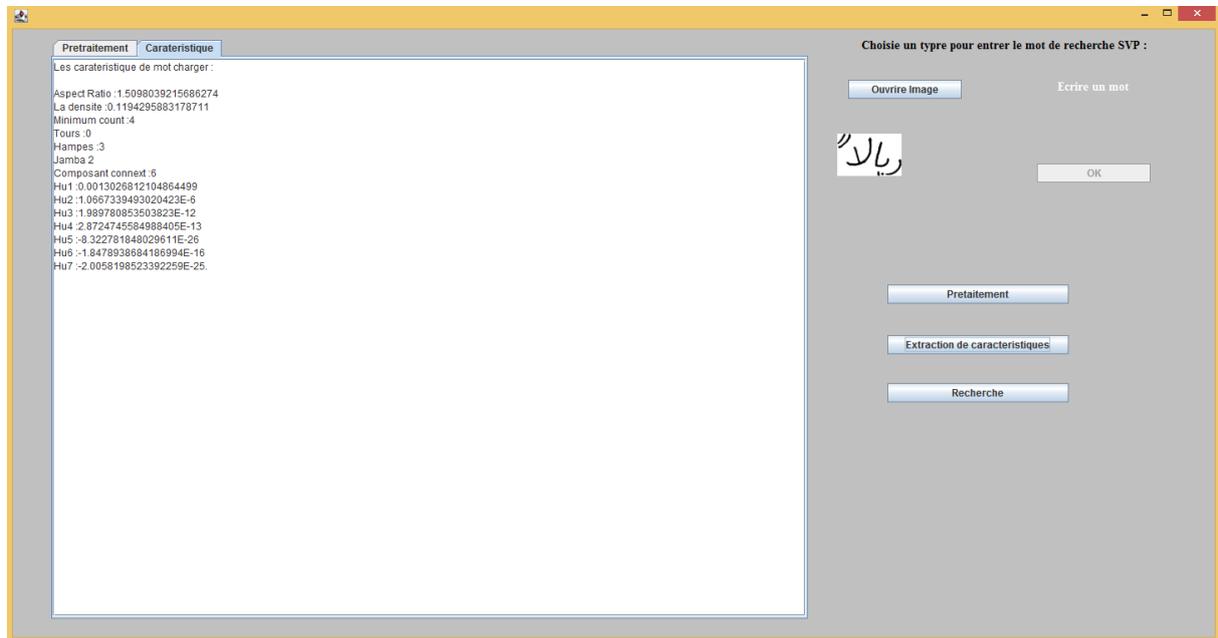


Figure IV.14. Extraction des caractéristiques de la requête

### IV.3.3.3. Recherche

Il ne reste maintenant que de lancer la recherche en activant le bouton "Rechercher". L'outil affiche les documents contenant des mots similaires à la requête introduite par l'utilisateur. Ces mots similaires sont mis en évidence dans les documents à l'aide d'un rectangle rouge, permettant ainsi à l'utilisateur de repérer facilement les occurrences correspondantes.

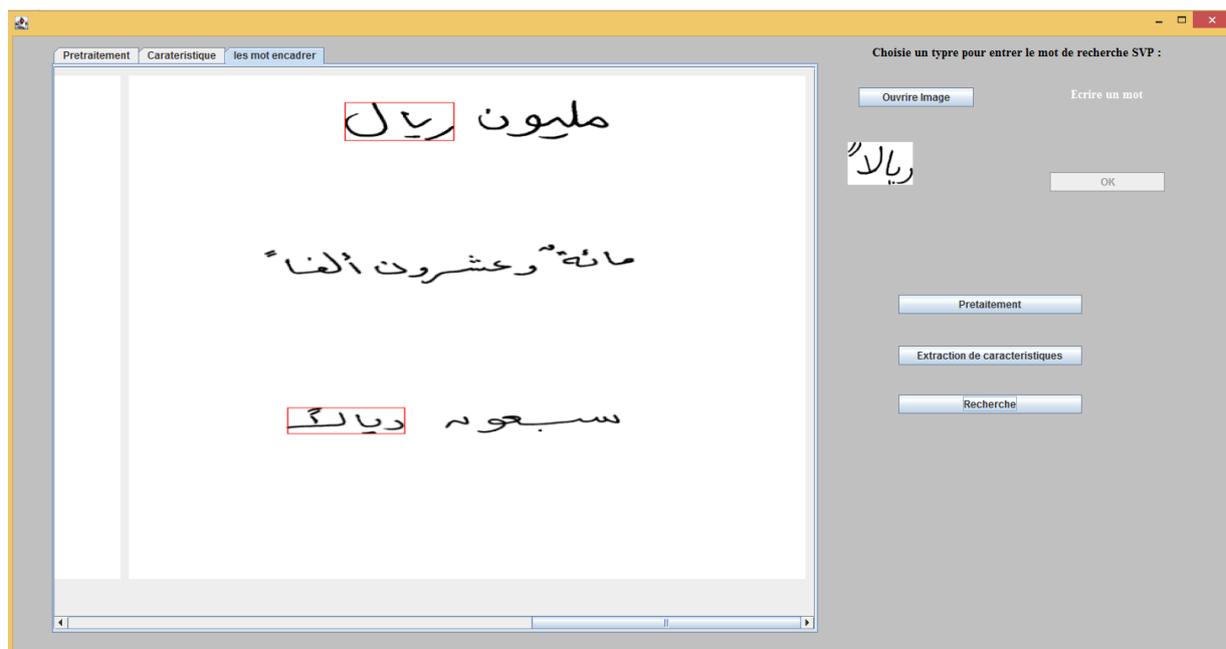


Figure IV.15. Affichage d'une image résultante de la recherche

## IV. 4. Expérimentations et résultats

Dans cette section nous présentons les bases d'images utilisées, les expérimentations menées pour l'évaluation des performances de notre approche proposée, ainsi que les résultats obtenus.

### IV.4.1. Bases d'images utilisés

Deux ensembles d'images ont été utilisées pour l'évaluation des performances de notre système. Nous décrivons brièvement chacune d'entre eux.

#### IV.4.1.1. Base de données AHDB

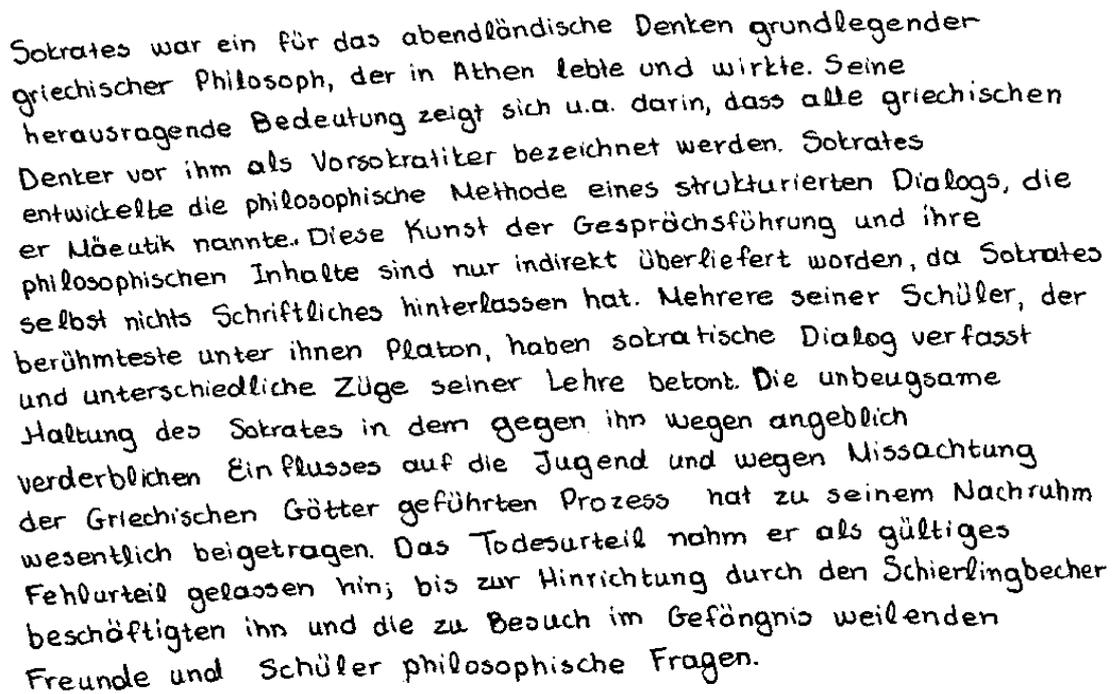
La base de données AHDB [80] est une collection de documents manuscrits contenant des images des mots et textes arabes.

Une centaine de scripteurs étaient invités à écrire des mots provenant du vocabulaire des montants numériques, et quelques lignes de textes libres. Ces données sont composées d'un lexique de 47 mots. Dans l'ensemble, la base AHDB se compose de 150 images sous forme binaire. Pour nos tests, nous avons sélectionné un sous-ensemble de 20 images à partir de cette base. Parmi ces images, la figure 4.3 est l'une des images incluses dans la base AHDB.

#### IV.4.1.2. Base de données de la compétition de word spotting ICFHR 2014

ICFHR (International Conference on Frontiers in Handwriting Recognition) est une conférence internationale de renommée spécialisée dans les domaines de l'analyse de documents, la reconnaissance de l'écriture manuscrite, etc. En 2014, ICFHR a organisé une compétition internationale sur le repérage de mots dans les documents manuscrits. Un grand nombre de chercheurs ont participé à cette compétition dans le but d'évaluer leurs méthodes et approche sur la même base d'images.

Cette base de données regroupe des exemples d'écriture manuscrite provenant d'une variété de sources et de styles d'écriture, rédigés en quatre langues (anglais, français, allemand et grec). La base de référence compte au total 100 images de documents (25 pour chacune des quatre langues), et 300 images de mots de requête. A partir de cette base, nous avons pris 10 images différentes pour les utiliser dans les tests. Des exemples d'images de documents de cette base de données sont illustrés à la Figure IV.16.



Sokrates war ein für das abendländische Denken grundlegender griechischer Philosoph, der in Athen lebte und wirkte. Seine herausragende Bedeutung zeigt sich u.a. darin, dass alle griechischen Denker vor ihm als Vorsokratiker bezeichnet werden. Sokrates entwickelte die philosophische Methode eines strukturierten Dialogs, die er Mäeutik nannte. Diese Kunst der Gesprächsführung und ihre philosophischen Inhalte sind nur indirekt überliefert worden, da Sokrates selbst nichts Schriftliches hinterlassen hat. Mehrere seiner Schüler, der berühmteste unter ihnen Platon, haben sokratische Dialog verfasst und unterschiedliche Züge seiner Lehre betont. Die unbeugsame Haltung des Sokrates in dem gegen ihn wegen angeblich verderblichen Einflusses auf die Jugend und wegen Missachtung der Griechischen Götter geführten Prozess hat zu seinem Nachruhm wesentlich beigetragen. Das Todesurteil nahm er als gültiges Fehlurteil gelassen hin; bis zur Hinrichtung durch den Schierlingbecher beschäftigten ihn und die zu Besuch im Gefängnis weilenden Freunde und Schüler philosophische Fragen.

Figure IV.16. Exemples d'image de la base de la compétition ICFHR 2014

#### IV.4.2. Les mesures d'évaluation utilisées

Pour quantifier les performances de notre système dans les différentes expériences, nous utilisons des mesures standard d'évaluation, à savoir: *Rappel*, *Précision*, et *F-Mesure*.

Notons *VP*, *FP*, et *FN*, le nombre de vrai positifs, faux positifs, et faux négatifs respectivement.

- Un vrai positif est le résultat où le système renvoie correctement une image.
- Un faux positif est le résultat où le système renvoie une image ne contenant pas le mot cherchée réellement.
- Un faux négatif est le résultat où le système échoue à trouver une correspondance.

*F-Mesure* a été introduit la première fois par Chinchor in [81]. A cause de sa simplicité, ce métrique est considéré comme l'une des mesures les plus utilisées pour l'évaluation quantitative des systèmes de recherche. *F-measure* est donné par :

$$FM = \frac{2 \times \text{Rappel} \times \text{Précision}}{\text{Rappel} + \text{Précision}}$$

$$\text{Avec } \text{Rappel} = \frac{VP}{VP+FN} \text{ et } \text{Précision} = \frac{VP}{VP+FP}$$

*F-Measure* atteint sa meilleure valeur à **1** (précision et rappel parfaite) et le pire à **0**.

#### IV.4.3. Expérimentations

Afin d'évaluer les performances de l'approche proposée à différents niveaux, plusieurs tests ont été réalisés en utilisant deux ensembles de caractéristiques distincts. Chacun de ces ensembles a été évalué individuellement pour mesurer son efficacité dans la tâche donnée.

Par la suite, nous avons étudié l'impact de la combinaison de ces ensembles de caractéristiques. En les combinant, nous cherchions à déterminer si une synergie pouvait être créée pour améliorer les performances globales de notre approche.

Dans chacune de ces expérimentations, le système est interrogé par un certain nombre de requêtes images et textuelles (**40** requêtes pour la première base et **20** requêtes pour la deuxième base), pour lesquelles nous avons établi manuellement la liste de documents pertinents. Les tableaux IV.1 et IV.2 récapitulent les requêtes utilisées dans les différentes expérimentations pour interroger chacune des deux bases ainsi que le nombre de documents attendus correspondants :

Requête	Nb de documents attendus	Requête	Nb de documents attendus
01	01	18	01
02	03	19	01
03	03	20	02
04	04	21	01
05	04	22	03
06	08	23	01
07	11	24	01
08	06	25	01
09	02	26	01
10	02	27	01
11	13	28	01
12	01	29	01
13	14	30	01
14	05	31	01
15	06	32	01
16	01	33	01
17	01		0

Tableau IV.1. Les requêtes utilisées pour interroger la première base ainsi que leurs nombres d'occurrences respectifs.

Requête	Nb de documents attendus	Requête	Nb de documents attendus
01	08	11	13
02	07	12	07
03	11	13	09
04	05	14	10
05	07	15	09
06	10	16	04
07	12	17	06
08	05	18	14
09	05	19	07
10	09	20	06

Tableau IV.2. Les requêtes utilisées pour interroger la deuxième base ainsi que leurs nombres d'occurrences respectifs.

#### IV.4.3.1. Expérimentation 1

Dans notre première expérimentation, nous avons évalué les performances du premier ensemble de caractéristiques. Cet ensemble est composé de 15 caractéristiques, à savoir :

- Le rapport d'aspect.
- La densité des pixels noirs.
- Le nombre de minimas dans la projection verticale.
- Le nombre de boucles.
- Le nombre d'hampes et de jambages.
- Le nombre de points clés.
- Le nombre de composantes connexes.
- Les 7 moments de Hu.

Cette expérimentation initiale nous a permis d'évaluer l'efficacité de cet ensemble de caractéristiques et de recueillir des résultats préliminaires sur les performances de l'approche proposée.

Après avoir appliqué les différentes étapes de traitement sur chaque image des deux bases de données, nous avons lancé la recherche en interrogeant les deux ensembles d'images par les différentes requêtes décrite précédemment.

Le tableau IV.3 suivant récapitule le rappel, la précision, et F-measure moyens obtenus sur les deux ensembles d'images.

	<b>Rappel</b>	<b>Precision</b>	<b>F-measure</b>
<b>1<sup>er</sup> ensemble</b>	0,67	0,61	0,63
<b>2<sup>ème</sup> ensemble</b>	0,47	0,42	0,44

Tableau IV.3. Performances obtenues sur les deux corpus d'images en employant le premier ensemble de caractéristiques

La précision du taux s'améliore considérablement avec des requêtes telles que "الريال خمس". Ce terme est clair dans tous les documents et facilite l'extraction des caractéristiques pertinentes. Cependant, il y a quelques requêtes qui ne sont pas optimales, ce qui affecte la qualité du taux. En particulier, lorsqu'il s'agit de saisir une requête complexe et peu compréhensible, comme lors de la saisie d'une requête sous forme de texte, les résultats obtenus peuvent être très incomplets par rapport à une saisie effectuée sous forme d'image.

#### IV.4.3.2. Expérimentation 2

Dans notre deuxième expérience, nous avons utilisé un deuxième ensemble de caractéristiques pour évaluer les performances sur les résultats de recherche. Cet ensemble se compose des points-clés des mots extraits à l'aide de l'algorithme SIFT (Scale-Invariant Feature Transform). Chaque mot est décrit par un ensemble de 50 caractéristiques, où 50 est le nombre maximum de points-clés pouvant être extraits à partir d'une image de mot.

Le tableau V.3 suivant résume les performances moyennes obtenus par l'utilisation des caractéristiques des points-clés SIFT, et en interrogeant les deux bases par les différentes requêtes.

	<b>Rappel</b>	<b>Precision</b>	<b>F-measure</b>
<b>1<sup>er</sup> ensemble</b>	0,59	0,51	0,55
<b>2<sup>ème</sup> ensemble</b>	0,44	0,39	0,41

Tableau IV.4. Performances obtenues sur les deux corpus d'images en employant le deuxième ensemble de caractéristiques

#### IV.4.3.3. Expérimentation 3

Dans notre troisième expérience, nous avons réalisé une combinaison des deux ensembles de caractéristiques précédemment utilisés. Nous avons fusionné les 15 caractéristiques du premier ensemble avec les 50 des points-clés SIFT, ce qui nous a donné un total de 65 caractéristiques. Cette combinaison vise à exploiter les avantages et les complémentarités des deux ensembles de caractéristiques en vue d'obtenir une représentation plus riche et discriminante des mots, permettant ainsi d'améliorer les performances de notre système de recherche.

Les performances moyennes obtenues par notre système en combinant toutes les caractéristiques sur les deux corpus d'images sont résumées dans le tableau suivant:

	<b>Rappel</b>	<b>Precision</b>	<b>F-measure</b>
<b>1<sup>er</sup> ensemble</b>	0,60	0,54	0,56
<b>2<sup>ème</sup> ensemble</b>	0,45	0,40	0,42

Tableau IV.5. Performances obtenues sur les deux corpus d'images en combinant toutes les caractéristiques

Malgré l'utilisation de cinquante fonctionnalités, les résultats demeurent insatisfaisants, car les premières fonctionnalités jouent un rôle crucial dans la distinction entre les mots et les mots significatifs, ce qui impacte la classification des termes. De plus, l'ajout de non-attributs a un impact significatif sur les résultats obtenus.

#### **IV. 5. Conclusion**

Dans ce chapitre, nous avons plongé plus en profondeur dans l'implémentation pratique de notre système. Nous avons présenté les ressources et les outils de développement que nous avons utilisés, tels que des langages de programmation, les IDE, et les bibliothèques. Ensuite, nous avons mis l'accent sur les interfaces principales de notre application, en décrivant les différentes parties de l'interface utilisateur que nous avons conçu. Finalement, nous avons présenté les expérimentations effectuées, les bases d'images utilisées, et les résultats obtenus.

Ce chapitre joue un rôle essentiel dans la compréhension de la façon dont notre système a été développé. Nous avons partagé notre expérience pratique, nos succès, en fournissant des détails sur les étapes spécifiques que nous avons suivies pour concrétiser notre vision.

# Conclusion générale et perspectives:

---

## Conclusion:

En conclusion, cette étude a exploré le domaine du repérage des mots dans les images de document et a présenté notre contribution dans ce domaine à travers la proposition d'une approche pour le repérage de mots dans les documents arabes.

À travers les différents chapitres, nous avons abordé les concepts clés liés au domaine général de l'analyse et la reconnaissance de documents et à son sous-domaine: le repérage de mots. Nous avons également examiné les différentes approches existantes, développé une méthodologie adaptée et mis en œuvre notre propre système de repérage des mots.

Le premier chapitre a fourni une introduction générale au repérage des mots, comme une alternative efficace à la reconnaissance de caractères pour extraire des informations pertinentes des documents, en particulier pour les documents anciens et les manuscrits. Il a mis en évidence son importance dans divers domaines et a identifié les défis et les difficultés associés à cette tâche, ce qui a jeté les bases pour notre recherche. Le premier chapitre a également passé en revue les approches existantes de repérage des mots, en mettant l'accent sur les approches holistiques et analytiques. Nous avons constaté que les approches analytiques, qui se concentrent sur la détection des caractéristiques visuelles des mots, ont montré de bons résultats, mais qu'elles sont souvent limitées par la variabilité des mises en page et des écritures.

Le deuxième chapitre a passé en revue les approches existantes de repérage des mots, en mettant l'accent sur les approches holistiques et analytiques. Nous avons constaté que les approches analytiques, qui se concentrent sur la détection des caractéristiques visuelles des mots, ont montré de bons résultats, mais qu'elles sont souvent limitées par la variabilité des mises en page et des écritures.

Dans le troisième chapitre, nous avons proposé une approche de repérage des mots qui combine la segmentation d'image, l'extraction de caractéristiques et la classification pour identifier les mots dans les documents. Nous avons décrit en détails les étapes de notre méthode, en mettant l'accent sur l'importance de la préparation des données et du choix des caractéristiques pertinentes.

Enfin, le quatrième chapitre a mis en lumière l'implémentation pratique de notre système de repérage des mots, en décrivant l'environnement de développement, l'architecture générale du système et les interfaces principales de l'application. Les expérimentations effectuées et les résultats obtenus ont également été présentés dans ce chapitre ainsi que la discussion des résultats. Nous avons montré comment notre système est capable de traiter les documents, d'indexer les mots et de fournir des résultats de recherche précis et pertinents.

Dans l'ensemble, cette étude a abouti à la création d'un système de repérage des mots prometteur, qui présente des performances encourageantes dans la détection et la localisation précise des mots dans des images de documents. Notre approche présente des avantages significatifs, tels que la capacité à traiter différents types de documents et à s'adapter à des conditions variées.

### **Perspectives:**

Les résultats obtenus encouragent la poursuite des travaux dans ce domaine, avec la possibilité d'explorer des approches plus avancées et d'étendre la portée de notre système. Ainsi, plusieurs perspectives sont envisageables:

- Comparer l'approche avec des méthodes bien connues de la littérature afin de pouvoir l'évaluer sur une base solide.
- Tester l'application sur une base de données réelle et plus large comportant divers types de documents écrits en différentes langues.
- Employer des techniques de segmentation plus performantes. Nous avons constaté que dans plusieurs cas, les erreurs dans le spotting ont été causées par les défauts de segmentation.
- Inclure d'autres caractéristiques dans le système, qui peuvent améliorer la discrimination de mots.
- Penser à l'utilisation d'autres algorithmes d'apprentissage que ce soit au niveau du regroupement des mots similaires ou au niveau du repérage.

## Bibliographie

---

- [01] Ghorbal, A. (2016), Filtres généralisés de Haar pour l'analyse de documents. Application aux word spotting et extraction de texte dans les bandes dessinées ((Doctoral dissertation, Université de Laroche)).
- [02] Biller, O., Asi, A., Kedem, K., El-Sana, J., and Dinstein, I. (2013). Webgt: An interactive webbased system for historical document ground truth generation. In 2013 12th International Conference on Document Analysis and Recognition, pages 305–308.
- [03] Ghilas, H., Gagaoua, M., Tari, A., & Cheriet, M. (2017, July). Arabic word spotting based on key-points features. In *8th International Conference of Pattern Recognition Systems (ICPRS 2017)* (pp. 1-5). IET.
- [04] R. Manmatha and J. L. Rothfeder, A scale space approach for automatically segmenting words from historical handwritten documents, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 27(8), 1212-1225 (2005), doi: 10.1109/TPAMI.2005.150
- [05] Rath, T.M., & Manmatha, R. (2007). Word spotting for historical documents. *International Journal on Document Analysis and Recognition*, 9(2-4), 139.
- [06] Rath, T. M., & Manmatha, R. (2003, August). Features for word spotting in historical manuscripts. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* (pp. 218-222). IEEE.
- [07] Lee, D. R., Hong, W., & Oh, I. S. (2012, April). Segmentation-free word spotting using SIFT. In *2012 IEEE Southwest Symposium on Image Analysis and Interpretation* (pp. 65-68). IEEE.
- [08] Zhang, S., Liu, Y., Jin, L., & Luo, C. (2018, April). Feature enhancement network: A refined scene text detector. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).
- [09] Wshah, S., Kumar, G., & Govindaraju, V. (2012, September). Script independent word spotting in offline handwritten documents based on hidden markov models. In *2012 International Conference on Frontiers in Handwriting Recognition* (pp. 14-19). IEEE.
- [10] Al-Khayat, M. (2014). *Learning-based Arabic word spotting using a hierarchical classifier* (Doctoral dissertation, Concordia University).
- [11] Moghaddam, R. F., & Cheriet, M. (2009, July). Application of multi-level classifiers and clustering for automatic word spotting in historical document images. In *2009 10th International Conference on Document Analysis and Recognition* (pp. 511-515). IEEE.

- [12] Sari, T., & Kefali, A. (2008, October). A search engine for Arabic documents. In *Colloque International Francophone sur l'Écrit et le Document* (pp. 97-102). Groupe de Recherche en Communication Ecrite.
- [13] Terasawa, K., Nagasaki, T., & Kawashima, T. (2005, August). Eigenspace method for text retrieval in historical document images. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)* (pp. 437-441). IEEE.
- [14] Terasawa, K., & Tanaka, Y. (2009, July). Slit style HOG feature for document image word spotting. In *2009 10th international conference on document analysis and recognition* (pp. 116-120). IEEE.
- [15] Ballard, D. (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111 – 122.
- [16] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg.
- [17] Leydier, Y., Lebourgeois, F., & Emptoz, H. (2007). Text search for medieval manuscript images. *Pattern Recognition*, 40(12), 3552-3567.
- [18] Leydier, Y., Le Bourgeois, F., & Emptoz, H. (2005, August). Omnilingual segmentation-free word spotting for ancient manuscripts indexation. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)* (pp. 533-537). IEEE.
- [19] Rath, Gales, K. M., Ragni, A., & M. J., Knill, S. P. (2014, May). Speech recognition and keyword spotting for low-resource languages: Babel project research at cued. In *Fourth International workshop on spoken language technologies for under-resourced languages (SLTU-2014)* (pp. 16-23). International Speech Communication Association (ISCA).
- [20] J. A. Rodríguez, G. Sánchez, J. Lladós and P. Pratim-Roy. Word spotting in archive documents using shape context. *Journal of Pattern Recognition and Image Analysis*, vol. 4478, no. 3, pages 290–297, 2007. (Cited on pages 17, 60 and 81.)
- [21] M. Rusiñol, D. Aldavert, R. Toledo and J. Lladós. Browsing heterogeneous document collections by a segmentation-free word spotting method. In *Proceedings of the 11th Int. Conf. on Document Analysis and Recognition*, pages 63–67, 2011. (Cited on pages 21, 31, 65, 79, 81, 82, 85, 86, 91, 92 and 134.)
- [22] J. Almazán, A. Gordo, A. Fornés and E. Valveny. Efficient exemplar word spotting. In *Proceedings of the British Machine Vision Conference*, pages 67.1–67.11, 2012. (Cited on pages 17, 79, 82, 85, 87, 92 and 134.)
- [23] J. Almazán, A. Gordo, A. Fornés and E. Valveny. Handwritten word spotting with corrected attributes. In *Proceedings of the Int. Conf. On Computer Vision*, pages 1017–1024, 2013. (Cited on pages 79, 82, 83, 85, 88, 92 and 134.)

- [24] A. Filatov, A. Gitis and I. Kil. Graph-based handwritten digit string recognition. In Proceedings of the 3rd Int. Conf. on Document Analysis and Recognition, pages 845–849, 1995. (Cited on page 83.)
- [25] D. López and J. M. Sempere. Handwritten digit recognition through inferring graph grammars. In Proceedings of Joint IAPR Int. Workshops on Advances in Pattern Recognition, pages 483–491, 1998. (Cited on page 83.)
- [26] S. Lu, Y. Ren and C. Suen. Hierarchical attributed graph representation and recognition of handwritten Chinese characters. *Journal of Pattern Recognition*, vol. 24, no. 7, pages 617–632, 1991. (Cited on pages 32, 68 and 83.)
- [27] A. J. Hsieh, K. G. Fan and T. I. Fan. Bipartite weighted matching for on-line handwritten Chinese character recognition. *Journal of Pattern Recognition*, vol. 28, no. 2, pages 422–446, 1995. (Cited on page 83.)
- [28] P. N. Suganthan and H. Yan. Recognition of handprinted Chinese characters by constrained graph matching. *Journal of Image and Vision Computing*, vol. 16, no. 3, pages 191–201, 1998. (Cited on page 83.)
- [29] M. Zaslavskiy, F. Bach and J.-P. Vert. A path following algorithm for the graph matching problem. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 31, pages 2227–2242, 2009. (Cited on pages 32, 68 and 83.)
- [30] A. Fischer, C. Y. Suen, V. Frinken, K. Riesen and H. Bunke. A fast matching algorithm for graph based handwriting recognition. In *Lecture Note in Computer Science*, volume 7887, pages 194–203, 2013. (Cited on pages 32, 68, 83, 84 and 85.)
- [31] Y. Chherawala, R. Wisnovsky and M. Cheriet. Tsv-lr: Topological signature vector-based lexicon reduction for fast recognition of premodern Arabic subwords. In Proceedings of the Workshop on Historical Document Imaging Processing, pages 6–13, 2011. (Cited on pages 32, 68 and 83.)
- [32] J. Lladós D. Fernández and A. Fornés. Handwritten word spotting in old manuscript images using a pseudo-structural descriptor organized in a hash structure. *Journal of Pattern Recognition and Image Analysis*, vol. 6669, pages 628–635, 2011. (Cited on pages 79, 84, 85, 86, 89, 92 and 134.)
- [33] J. Lladós, M. Rusiñol, A. Fornés, D. Fernández and A. Dutta. On the influence of word representations for handwritten word spotting in historical documents. *Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 5, pages 1 263 002.1–1 263 002.25, 2012. (Cited on pages 32, 68, 79, 84, 85, 86, 127 and 128.)
- [34] A. Dutta, J. Gibert, J. Lladós, H. Bunke and U. Pal. Combination of product graph and random walk kernel for symbol spotting in graphical documents. In Proceedings of the Int. Conf. on Pattern Recognition, pages 1663–1666, 2012. (Cited on pages 67 and 85.)

- [35] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, vol. 41, no. 6, 1990. (Cited on pages 32 and 86.)
- [36] D. Aldavert, M. Rusiñol, R. Toledo and J. Lladós. Integrating visual and textual cues for query-by-string word spotting. In *Proceedings of the 12th Int. Conf. on Document Analysis and Recognition*, pages 511–515, 2013. (Cited on pages 21, 28, 31, 79, 85, 86, 88, 91, 92 and 134.)
- [37] Y. Kessentini, T. Paquet and A. B. Hamadou. Off-line handwritten word recognition using multi-stream hidden Markov models. *Journal of Pattern Recognition Letters*, vol. 31, no. 1, pages 60–70, 2010. (Cited on pages 85 and 86.)
- [38] L. Rothacker, M. Rusiñol and G. A. Fink. Bag-of-Features HMMs for segmentation-free word spotting in handwritten documents. In *Proceedings of 12th Int. Conf. Document Analysis and Recognition*, 2013. (Cited on pages 21, 31, 79, 85, 86, 91, 92 and 134.)
- [39] A. H. Toselli and E. Vidal. Word-graph based handwriting key-word spotting: impact of word-graph size on performance. In *Proceedings of the 11th IAPR Int. Workshop on Document Analysis Systems*, pages 176–180, 2014. (Cited on pages 79, 85, 86, 88, 92 and 134.)
- [40] V. Frinken, A. Fischer and H. Bunke. A novel word spotting algorithm using bidirectional long short-term memory neural networks. In *Proceedings of the 4th IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, volume 5998, pages 185–196, 2010. (Cited on pages 85, 87, 89, 90, 91, 92 and 134.)
- [41] V. Malleron, V. Eglin, H. Emptoz, S. Dord-CrouslÃc and P. RÃ©gnier. Hierarchical decomposition of handwritten manuscripts layouts. In *Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns*, pages 221–228, 2009. (Cited on page 90.)
- [42] K. Hadjar , "Une Ã©tude de l'Ã©volutivitÃ© des modÃ©les pour la reconnaissance de documents arabes dans un contexte interactif ", Diss, ThÃ¨se de doctorat, UniversitÃ© Fribourg (Suisse), 2006.
- [43] B. Bachimont, "BibliothÃ¨ques numÃ©riques audiovisuelles: des enjeux scientifiques et techniques", *Document numÃ©rique*, pp. 2.3-4, 1998.
- [44] Ouwayed, Nazih. Segmentation en lignes de documents anciens: application aux documents arabes. Diss. 2010.
- [45] E. Trupin, "La reconnaissance d'images de documents: Un panorama", *TS, Traitement du signal*, Vol. 22, No.3, pp. 159-189, 2005.
- [46] G. Nagy, "Twenty years of document image analysis in PAMI", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No.1, pp. 38-62, 2000.
- [47] L. Robadey, " 2 (CREM): Une mÃ©thode de reconnaissance structurelle de documents complexes basÃ©e sur des patterns bidimensionnels", Diss, 2001.

- [48] D. Ketata, K. Maher, "Un survol sur l'analyse et la reconnaissance de documents: imprimé, ancien et manuscrit" ,*Colloque International Francophone sur l'Écrit et le Document (CIFED)*, 2010.
- [49] [https://fr.wikipedia.org/wiki/Gestion\\_des\\_documents\\_d'archives](https://fr.wikipedia.org/wiki/Gestion_des_documents_d'archives) Consulté le 15/02/2017.
- [50] Belaïd, A., & Belaïd, Y. (1992). Reconnaissance des formes : méthodes et applications (p. 429). InterEditions.
- [51] Kricha, A., Lasmar, A. G., & Amara, N. E. B. (2006, September). Exploration des ondelettes en prétraitement des documents anciens. In *Actes du 9ème Colloque International Francophone sur l'Écrit et le Document* (pp. 157-162). SDN06.
- [52] Boulehmi, H., Seddik, B., Kricha, A., & Amara, N. E. B. (2008, October). Prétraitement de documents anciens. In *Colloque International Francophone sur l'Écrit et le Document* (pp. 209-210). Groupe de Recherche en Communication Ecrite.
- [53] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1), 62-66.
- [54] Kapur, R. P., Yost, C., & Palmiter, R. D. (1993). Aggregation chimeras demonstrate that the primary defect responsible for aganglionic megacolon in lethal spotted mice is not neuroblast autonomous. *Development*, 117(3), 993-999.
- [55] F.R.D. Velasco, « Thresholding using the ISODATA clustering algorithm », *IEEE Transaction on system, Man and Cybernetics*, 10, p. 771-774, 1980
- [56] H.K. Aghajan and T. Kailath, « SLIDE: Subspace-based line detection », *Pattern Analysis and Machine Intelligence*, vol. 16, No.11, pp. 1057-1073, 1994.
- [57] A. Belaid, "Conception assistée de modèles de page en vue de leur utilisation en reconnaissance de documents", *Lausanne-Atelier sur les modèles de pages électroniques*, pp.17,(Lausanne),Septembre 1997, disponible sur : <http://www.loria.fr/~abelaid/Publications.html> (consulté le 19.07.2002).
- [58] Alamri, Huda, et al. "A novel comprehensive database for Arabic off-line handwriting recognition." *Proceedings of 11th International Conference on Frontiers in Handwriting Recognition, ICFHR*. Vol. 8. 2008.
- [59] D. Ketata, K. Maher, "Un survol sur l'analyse et la reconnaissance de documents: imprimé, ancien et manuscrit" ,*Colloque International Francophone sur l'Écrit et le Document (CIFED)*, 2010.
- [60] K.Y. Wong, R.G. Casey, F.M. Wahl, « Document analysis system », *IBM journal of research and development*, vol. 26, No. 6, pp. 647–656, 1982.

- [61] D. Wang, N.S .Sargur, N. Srihari, «Classification of newspaper image blocks using texture analysis», *Computer Vision, Graphics, and Image Processing*, vol. 47, No.3, pp. 327-352, 1989.
- [62] D. Wang, N.S .Sargur, N. Srihari, «Classification of newspaper image blocks using texture analysis», *Computer Vision, Graphics, and Image Processing*, vol. 47, No.3, pp. 327-352, 1989.
- [63] G. Nagy, J. Kanai, M. Krishnamoorthy, M. Thomas, M. Viswanathan, «Two complementary techniques for digitized document analysis», *DOCPROCS : Proceedings of the ACM conference on Document Processing Systems*, pp169–176, 1988.
- [64] H. Boufersaoui, I. Frihi, « Extraction de la structure logique des documents ». Mémoire de Master, Option : Ingénierie de médias, Université 08 Mai 1945 – Guelma, Juin 2015.
- [65] H.M. Sun, «Enhanced Constrained Run-Length Algorithm for Complex Layout Document Processing». *International Journal of Applied Science and Engineering*, vol. 4, No. 3, pp. 297 - 309, 2006.
- [66] S. N. Srihari, D. Wang, « Classification of newspaper image blocks using texture analysis». *Computer Vision Graphics and Image Processing*, vol. 47, No. 3, pp. 327–352, 1989.
- [67] Rodríguez-Serrano, J. A., & Perronnin, F. (2012). A model-based sequence similarity with application to handwritten word spotting. *IEEE transactions on pattern analysis and machine intelligence*, 34(11), 2108-2120.
- [68] Kefali, A. (2010). *Concurrence entre recherche approximative et classification pour la recherche d'images de documents arabes dégradés* (Doctoral dissertation, Annaba).
- [69] Granet, A. (2018). *Extraction d'information dans des documents manuscrits anciens* (Doctoral dissertation, Université de Nantes).
- [70] Devijver, P. A., & Kittler, J. (1982). *Pattern recognition: A statistical approach*. Prentice hall.
- [71] Nebti, S. (2014). *Reconnaissance de caractères manuscrits par intelligence collective* (Doctoral dissertation).
- [72] Nemouchi, S. (2010). *Reconnaissance de l'écriture arabe par systèmes flous* (Doctoral dissertation, Annaba).
- [73] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (Vol. 96, No. 34, pp. 226-231).

- [74] Kesiman, M. W. A. (2018). *Document image analysis of Balinese palm leaf manuscripts: Analyse d'images de documents des manuscrits balinais sur feuilles de palmier* (Doctoral dissertation, La Rochelle).
- [75] Koerich, A. L. (2003, September). Unconstrained handwritten character recognition using different classification strategies. In *International Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR)*.
- [76] Sanjeev Kunte, R., & Sudhaker Samuel, R. D. (2007). A simple and efficient optical character recognition system for basic symbols in printed Kannada text. *Sadhana*, 32(5), 521-533.
- [77] Mahmoud, S. A. (1994). Arabic character recognition using Fourier descriptors and character contour encoding. *Pattern recognition*, 27(6), 815-824.
- [78] Vamvakas, G., Gatos, B., Petridis, S., & Stamatopoulos, N. (2007, September). An efficient feature extraction and dimensionality reduction scheme for isolated greek handwritten character recognition. In *Ninth international conference on document analysis and recognition (ICDAR 2007)* (Vol. 2, pp. 1073-1077). IEEE.
- [79] Guellil, Z., & Zaoui, L. (2009). Proposition d'une Solution au Problème d'Initialisation Cas du K-means. In *CIIA*.
- [80] AL-MA'ADEED S., ELLIMAN D., HIGGINS C., A data base for Arabic handwritten text recognition research, 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR), pp. 485-489, 2002.
- [81] Chinchor, N., & Sundheim, B. M. (1993). MUC-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.