



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A.MIRA-BEJAIA
Faculté des sciences exactes
Département informatique
Laboratoire d'informatique médicale(LIMED)

THÈSE

Présentée par

GHILAS Hamza

Pour l'obtention du grade de

DOCTEUR EN SCIENCES

Filière : Informatique

Option : Cloud Computing

Thème

Repérage de mots dans les anciens documents Arabes

Soutenue le : 23 Novembre 2021

Devant le Jury composé de :

Nom et Prénom

Grade

Mr AZNI Mohamed

MCA

Univ. de Bejaia

Président

Mr TARI Abdelkamel

Professeur

Univ. de Bejaia

Rapporteur

Mr AZOUAOU Faiçal

Professeur

ESTIN, Bejaia

Examineur

Mr SEBAA Abderrazak

MCA

ESTIN, Bejaia

Examineur

Année Universitaire : 2021/2022

Remerciements

Je remercie Mr **TARI Abdelkamel** mon directeur de thèse pour ses conseils, ses recommandations, sa disponibilité et son écoute.

Je remercie Mr **CHERIET Mohamed** de nous avoir ouvert les portes de son laboratoire pour effectuer des stages et collaborer avec les chercheurs de l'école des technologies supérieures à Montréal.

Je remercie mon épouse **GAGAOUA Meriem** d'être patiente et compréhensive.

Je remercie les membres de jury d'avoir accepté d'examiner mon travail.

Je remercie tous ceux qui ont contribué de près ou de loin dans la réalisation de ce travail.

Dédicaces

Je dédie ce modeste travail à :

- Mes chers parents.
- Ma chère femme Meriem.
- Mon fils Mohamed (MoMoh).

Liste des travaux

Revue internationale avec comité de lecture :

H. Ghilas, M. Gagaoua, A. Tari and M. Cheriet, Spatial Distribution of Ink at Keypoints (SDIK): a novel feature for word spotting in Arabic documents, *International Journal of Image and Graphics*, (2021).

M. Gagaoua, H. Ghilas, A. Tari and M. Cheriet, Histogram of Marked Background (HMB) Feature Extraction Method for Arabic Handwriting Recognition, *International Journal of Image and Graphics*, 2250015 (2021), doi:10.1142/S0219467822500152

Communications internationales avec acte et comité de lecture :

H. Ghilas, M. Gagaoua, A. Tari and M. Cheriet, "Arabic word spotting based on key-points features," *8th International Conference of Pattern Recognition Systems (ICPRS 2017)*, 2017, pp. 1-5, doi:10.1049/cp.2017.0162.

M. Gagaoua, H. Ghilas, A. Tari and M. Cheriet, Distribution, Directional, structural and concavity features for historical Arabic handwritten recognition: a comparative study, in *Proceedings of the International Conference on Computing for Engineering and Sciences*. 2017, Association for Computing Machinery: Istanbul, Turkey. p. 70–75.

Table des matières

Liste des figures.....	IV
Liste des tableaux.....	VI
Index	VII
Introduction générale	9
Problématique	9
Les anciens manuscrits Arabes.....	9
La numérisation des anciens manuscrits.....	10
Le repérage de mots (Word spotting)	11
L'écriture arabe manuscrite.....	11
Objectif de la thèse.....	12
Contributions	12
Structure de la thèse.....	13
1. Le repérage de mots dans les images de documents.	14
1.1. Classification des méthodes de repérage de mots	14
1.2. L'architecture d'un système de repérage de mots.....	15
1.3. Binarisation	17
1.3.1. Binarisation par seuillage	17
1.3.2. Détection de contour.....	19
1.3.3. Modèles statistiques.....	20
1.3.4. Classification des pixels	22
1.4. La segmentation.....	23
1.4.1. Segmentation en lignes	23
1.4.2. Segmentation en mots.....	23
1.4.3. Segmentation en sous-mots	24
1.5. L'extraction des caractéristiques (features extraction).....	25
1.6. Calcul de similarité.....	26
1.6.1. Méthodes à base d'apprentissage.	26
1.6.2. Méthodes sans apprentissage.....	28

1.7.	Conclusion.....	29
2.	Extraction des caractéristiques : revue de la littérature.	30
2.1.	Méthodes de repérage de mots dans les documents latins	30
2.1.1.	Les profils des mots.....	30
2.1.2.	Caractéristiques de zonage.....	31
2.1.3.	Caractéristiques à base du Gradient.....	32
2.1.4.	Caractéristiques de type motif.....	33
2.1.5.	Caractéristiques structurelles.....	34
2.1.6.	Attributs des réseaux de neurones profonds (deep NN).....	35
2.2.	Extraction des caractéristiques des documents arabes	35
2.2.1.	Méthodes indépendantes de la langue	35
2.2.2.	Caractéristiques dédiées à la langue Arabe	36
2.3.	Conclusion.....	37
3.	Etude préliminaire : repérage de mots en utilisant les caractéristiques des points-clés.....	38
3.1.	Méthode proposée.....	38
3.2.	Préparation des données.....	39
3.2.1.	Binarisation et segmentation en sous-mots	40
3.2.2.	Classification des sous-mots.....	41
3.3.	Le repérage de mots (word spotting).....	44
3.3.1.	Extraction des points-clés.....	45
3.3.2.	Extraction des caractéristiques des points-clés	49
3.3.3.	Mesure de dissimilarité (image matching)	50
3.4.	Expérimentation et résultats	51
3.4.1.	La base d'image	51
3.4.2.	Métriques d'évaluation.....	51
3.4.3.	Résultat et discussions.....	53
3.5.	Conclusion.....	55
4.	Spatial Distribution of Ink at Keypoints (SDIK) : Un nouveau descripteur pour le repérage de mots dans les documents arabes.	56
4.1.	Contributions	56
4.2.	Méthode proposée.....	57

4.3.	Extraction des caractéristiques SDIK.....	58
4.4.	Calcul de similarité (Image Matching).....	60
4.4.1.	Expérimentation et résultats	62
4.4.2.	Les bases d'images.....	62
4.4.3.	Segmentation du document en lignes	63
4.4.4.	Résultats qualitatifs	64
4.4.5.	Résultats quantitatifs.....	66
4.4.6.	Comparaison avec la littérature	69
4.5.	Conclusion.....	71
	Conclusion générale	72
	Bibliographie	75

Liste des figures

Figure 1. Quelques exemples d'images des manuscrits arabes.	10
Figure 1.1. Architecture générale des systèmes de repérage de mots.....	16
Figure 1.2. Exemple de binarisation d'image de document ; (a) l'image originale : page 11 de la base IBN SINA, (b) l'image afficher en niveau de gris, et (c) l'image binaire.	17
Figure 1.3. Exemples d'images binarisées avec la méthode d'Otsu, (a) image de document avec fond uniforme, (b) image de document avec arrière-plan irrégulier.	18
Figure 1.4. Exemple de binarisation d'image de document par détection de contour et la montée des eaux. (a) l'image originale, (b) les contours détectés par la méthode de Canny, (c) l'image binaire.	20
Figure 1.5. Binarisation d'images avec un CRF, (a) image originale, (b) image pré-binarisée (pixels incertain en bleu), (c) image binarisée avec la méthode de Sauvola, (c) binarisation optimisée par un CRF.	21
Figure 1.6. Binarisation par un réseau de neurone de type encodeur/décodeur.	22
Figure 1.7. Segmentation en mots avec la méthode de. (a) extraction des lignes, (b) segmentation en mots avec les distances entre les enveloppes des composantes connexes	24
Figure 1.8. Un mot arabe et ses trois PAWs. Les têtes des flèches pointent vers des CCs majeures, les autres CCs sont des composantes mineures.....	25
Figure 2.1. Profils de mot. (a) image originale, (b) profil de projection, (c) profil inférieur, (d) profil supérieur, et (e) transitions entre l'ancre et le fond.	31
Figure 2.2. Exemple de division d'une image en zones.....	32
Figure 2.3. Extraction des histogrammes de l'orientation du gradient.	33
Figure 2.4. Exemple d'extraction des histogrammes LBP.....	34
Figure 2.5. Exemple d'extraction des graphèmes.	34
Figure 3.1. Diagramme général du système proposé (étude préliminaire).	39
Figure 3.2. Préparation de données.....	40
Figure 3.3. Quelques images de la base IBN SINA.	41
Figure 3.4. Calcul de du rapport hauteur/largeur.....	41
Figure 3.5. Calcul du nombre de minimas dans la projection verticale d'une image d'un sous-mot.	42
Figure 3.6. Calcul du nombre de loupes.....	42
Figure 3.7. Calcul du nombre de jambes et de hampes.....	43
Figure 3.8. Le repérage de mots.	45
Figure 3.9. (a) image binarisée d'un mot arabe. (b) les points terminaux (rouge) et les points de branchements (vert) dans le squelette de l'image.	46
Figure 3.10. Minimais locaux dans la projection verticale d'une image.	47
Figure 3.11. Un exemple d'extraction de points clés : (a) points terminaux et points de branchement bruts. (b) Points clés obtenus après nettoyage par l'algorithme 2.	49
Figure 3.12. Extraction des caractéristiques de point-clés.	50
Figure 3.13. Un exemple des matchs des points-clés d'une CC dans une autre CC.	51
Figure 3.14. La précision de la classification des CCs.....	53

Figure 4.1. Trois instances d'un mot arabe écrit de différentes manières. Les rectangles montrent des parties non discriminantes de l'écriture.	58
Figure 4.2. Diagramme de la méthode proposé : (a) un document cible et un mot de requête. (b) Le document cible segmenté en lignes. (c) Caractéristiques SDIK extraites de chaque pointé dans les lignes du document et le mot de requête. (d) Lignes de texte ordonnées en fonction de leur similarité avec le mot de requête	58
Figure 4.3. Les zones directionnelles et circulaires utilisées pour partitionner le voisinage d'un point-clé.....	59
Figure 4.4. Calcul des descripteurs SDIK : (a) Le modèle de zones directionnelles et circulaires utilisé. (b) Descripteur SDIK affiché en niveaux de gris (sombre = valeur élevée). (c) Une partie d'une image avec les points-clés extraits	60
Figure 4.5. Exemples de vecteurs SDIK extraits à partir des points-clés d'une ligne de texte	60
Figure 4.6. Un exemple de calcul de dissimilarité entre un mot de requête et une ligne. (a) L'image de la requête et les matchs de ses points-clés dans la ligne cible. (b) La matrice de dissimilarité et les points-clés sélectionnés comme matchs des points-clés de la requête	62
Figure 4.7. L'image binarisée de la page 11 de la base d'image IBN SINA et sa projection horizontale, les pics entourés dans la projection horizontale sont sélectionnés comme emplacements des lignes de texte.	64
Figure 4.8. Les 10 premières lignes récupérées pour quelques requêtes. Les images en haut sont les requêtes.	65
Figure 4.9. Positions des occurrences pertinentes dans les résultats. Chaque ligne représente les résultats récupérés pour une image de requête. Les points noirs représentent des occurrences pertinentes.....	65
Figure 4.10. Courbes précision/rappel pour les requêtes de sous-mots et de mots.	67

Liste des tableaux

Tableau 3.1. Matrice de probabilité d'un modèle de mélange de gaussiennes à 8 classes.	44
Tableau 3.2. Résultats de la méthode proposée.	54
Tableau 3.3. Résultats des profils de mots avec DTW.	54
Tableau 4.1. Résultats obtenus en termes de MAP et P@k moyenne.	67
Tableau 4.2. Résultats obtenus en terme de MAP (%) pour différentes configurations de zones directionnelles et circulaires.....	68
Tableau 4.3. Résultats obtenus en terme de MAP (%) pour différentes valeurs du rayon r et du nombre de zones circulaires c.	68
Tableau 4.4 La précision moyenne et le temps de réponse moyen (en secondes) du descripteur SDIK et les autres méthodes.	70

Index

A

ANN	
Artificial Neural Network	26
AP	
Average Precision	52
<i>ASW</i>	
average stroke width	46

C

CDP	
Continuous Dynamic Programming	28
<i>CNM</i>	
Centre National des Manuscrits	10
CNN	
Convolutionnal Neural Network	22
CRF	
Conditional Random Fields	20

D

DTW	
Dynamic Time Warping	28

E

EM	
Expectation Maximization	20

F

FCN	
Fully Connected Networks	22
FSM	
Flexible Sequence Matching	28

G

GMM	
-----	--

Gaussian mixture model	20
GSC	
Gradient Structural Concavity	35

H

HMM	
Hidden Markov Model	26
HoG	
Histogram of Oriented Gradient	32

M

MAP	
Mean Average Precision	52

O

OCR	
Optical Character Recognition	11

P

PAW	
Part of Word	15

Q

QBE	
Query By Example	14
QBS	
Query By String	14

R

ReLU	
Rectified Linear Units	35
RLSA	
Run Length Smoothing Algorithm	24

S

SDIK	
Spatial distribution of Ink at Keypoints	13

Introduction générale

Problématique

Les anciens manuscrits transcrits en langue Arabe constituent un véritable patrimoine scientifique et culturel. La solution adoptée par la majorité des pays pour préserver leur patrimoine documentaire est la **numérisation**, ce qui a généré des quantités énormes des images de documents. L'indexation et la recherche d'information dans ces images est devenue un besoin incontournable. Plusieurs chercheurs sont intéressés par le développement d'algorithme de repérage de mots dans des images de document (*word spotting*). Dans le cas des documents transcrits en langue Arabe, il y a peu de travaux réalisés pour la recherche d'information et l'indexation. En plus des dégradations physiques dues à l'âge de ces manuscrits, la nature complexe de l'écriture arabe, font de l'exploration de ces images une tâche très complexe. La proposition d'un algorithme de repérage de mots dans ces images est un besoin incontournable.

Les anciens manuscrits Arabes

Les anciens manuscrits arabes est un patrimoine à sauvegarder soigneusement. Ils représentent une source d'information très importante pour les historiens et d'autres scientifiques. Leur contenu est pluridisciplinaire tel que la littérature (poésie), la religion, la philosophie, les mathématiques, la médecine, l'astronomie et d'autres domaines. La majorité de ces documents se trouvent actuellement dans un état inquiétant, des documents incomplets, des documents ayants subi des dégradations irréversibles, des documents sous forme de feuilles volantes et etc. Les supports utilisés pour la transcription de ces documents subissent généralement des dégradations avec la manipulation manuelle et l'âge de ces documents. En plus des artefacts humains, l'écriture arabe a évolué dans le temps et il existe plusieurs types d'écriture allant de sa forme simple jusqu'aux formes artistiques. La figure 1 montre des exemples de fragments de quelques anciens documents arabes. Cette variabilité des anciens documents arabes dans leur forme et leur contenu fait de la sauvegarde de ces documents un domaine de recherche très important et motivant. Cette thèse s'inscrit dans ce domaine et contribue dans la sauvegarde de ce patrimoine.



Figure 1. Quelques exemples d'images des manuscrits arabes.

La numérisation des anciens manuscrits

La solution idéale pour la préservation des anciens manuscrits est la numérisation, elle permet de créer une version numérique du manuscrit et éviter la manipulation manuelle du document original. Le format numérique d'un document facilite l'accès, le partage, et la reproduction des documents. L'Algérie a également opté pour la numérisation pour préserver le patrimoine documentaire de la nation. Un centre dédié à la numérisation des anciens manuscrits est implanté dans le sud du pays, dans la wilaya de Adrar. Le choix du site du *centre national des manuscrits (CNM)* reflète la concentration de ce type de document dans le sud du pays. Les anciens manuscrits sont généralement une propriété et un héritage familial. Ces manuscrits sont préservés par des individus sous forme de bibliothèque familiales appelées *khisanates*. L'objectif du centre est de faire l'inventaire national du fond des manuscrits, de localiser les *khisanates*, de cataloguer, de conserver et de procéder à la paléographie des plus importantes pièces de ces bibliothèques traditionnelles. La numérisation des anciens manuscrits a généré des quantités énormes d'images de documents, mais l'exploitation de ces images reste une tâche très complexe. Une bonne exploitation de ces images implique le développement des algorithmes pour l'indexation de ces images, la recherche d'information dans le contenu, la reconnaissance des caractères, la classification des images par la reconnaissance de scripteur et d'autre problématique. Le repérage de mots dans des images de documents permet de mieux exploiter ces images et permet aux moteurs de recherche de mieux indexer ces bibliothèques numériques.

Le repérage de mots (Word spotting)

Les OCRs (Optical Character Recognition) sont les premiers systèmes proposés pour le traitement des images de documents qu'ils soient imprimés ou manuscrits. Ils consistent à convertir une image de document au format texte. C'est la solution idéale pour mieux exploiter le contenu des documents, à savoir : le partage du contenu, la recherche d'information, la réutilisation de contenu, et etc. Dans le cas des anciens manuscrits, les résultats des OCRs ne sont pas satisfaisants, ceci est dû à la complexité de la forme de ces documents d'une part et les dégradations subies à cause de leurs âges d'une autre part. La solution alternative pour faciliter l'utilisation des images de document est le *word spotting*. Cette méthode se définit comme le *repérage de mots*, son objectif est de repérer dans une image un mot donné par l'utilisateur comme requête. Le repérage de mots est très bénéfique pour les moteurs de recherche ainsi que d'autres applications.

L'écriture arabe manuscrite.

La langue arabe est développée par la civilisation musulmane. Aujourd'hui c'est la langue officielle de plusieurs pays dans le monde. La langue arabe s'écrit de droite à gauche avec un alphabet de 28 consonnes et n'utilise pas les voyelles. De nombreuses lettres sont similaires par leur squelette et ne se distinguent que par des points utilisés comme diacritiques au-dessus ou au-dessous du squelette. L'écriture arabe est cursive, la majorité des lettres s'attachent entre elles à l'exception de quelques-unes. Les lettres qui ne s'attachent pas aux autres lettres créent des espaces à l'intérieur des mots arabes. Les frontières entre les mots de l'écriture arabe ne peuvent être connues par le lecteur que s'il comprend le sens. La segmentation visuelle d'un texte arabe en mots n'est pas évidente. L'écriture arabe est complexe par nature, dans le cas des anciens manuscrits, cette complexité est amplifiée par la mauvaise qualité de ces documents. De plus, l'écriture arabe a évolué dans le temps, la forme des lettres et la manière d'écrire varient d'un scripteur à l'autre. Le manque des méthodes de repérage de mots qui prennent en charge des documents arabes est la motivation principale du travail réalisé dans cette thèse.

Objectif de la thèse

Les travaux réalisés dans le cadre de cette thèse ont pour objectif de contribuer dans la sauvegarde et la préservation de l'ancien patrimoine de notre pays. Les systèmes de repérage de mots effectuent plusieurs étapes à savoir l'acquisition des images, la binarisation et la suppression des bruits, l'extraction des caractéristiques et finalement la recherche dans l'espace des primitives. L'étape ayant un lien direct avec la langue du document est l'extraction des caractéristiques. Notre objectif est de proposer une nouvelle méthode d'extraction des caractéristiques adaptée aux caractéristiques particulières et complexes de l'écriture arabe.

Contributions

La problématique de repérage de mots dans des images de document (word spotting) est largement traitée dans la littérature mais pour des documents transcrits en latin. Il y a très peu de solutions proposées pour le repérage de mots dans des documents arabes manuscrits et les résultats obtenus sont insuffisants. La structure, la forme ainsi les règles d'écriture particulières du manuscrit arabe implique la proposition d'un algorithme de repérage de mots qui prend en considération les caractéristiques particulières de l'écriture arabe. Nous avons élaboré une nouvelle méthode pour le repérage de mots dans un manuscrit arabe adaptée à la nature complexe de ces documents. La méthode proposée représente une image de document par des valeurs numériques extraites à partir des régions d'intérêts. La méthode proposée a résolu le problème de segmentation d'un document arabe en mots par un mécanisme de comparaison d'une image d'un mot de requête avec des images de lignes. Le repérage de mots se fait dans un document segmenté en lignes au lieu de le segmenter en mots. Nos contributions dans cette thèse se résument dans les travaux suivants :

1. Proposition d'un algorithme pour l'extraction des points-clés à partir des images de document.
2. Proposition d'une nouvelle méthode pour l'extraction des caractéristiques (features extraction).
3. Proposition d'un mécanisme de calcul de similarité entre une image de mot de requête et une ligne de texte entière du document cible.

Nos contributions dans le cadre de cette thèse sont concrétisées par une communication orale qui a démontré la faisabilité de l'idée de l'extraction des caractéristiques à partir des points-clés. Les résultats obtenus par l'application de la méthode proposée a fait l'objet d'une publication dans un journal du domaine.

Structure de la thèse.

Le reste de cette thèse est réparti sur 04 chapitres et une conclusion générale

- Chapitre 1 : dans ce chapitre nous avons présenté les éléments essentiels dans un système de repérage de mots à savoir : les méthodes de binarisation d'images de documents, les techniques de segmentation, l'extraction des caractéristiques et finalement les méthodes de comparaison d'image (image matching).
- Chapitre 2 : ce chapitre est un état de l'art sur les caractéristiques (primitives) utilisées dans la littérature pour la description des images de document. Nous avons présenté les différents types de primitives utilisées en générale et particulièrement les primitives utilisées avec l'écriture Arabe.
- Chapitre 3 : Dans ce chapitre nous avons présenté une première contribution qui est une étude préliminaire où nous avons démontré l'efficacité de l'utilisation des point-clés dans le repérage de mots. Nous avons présenté un algorithme que nous avons conçu pour l'extraction des points-clés, les caractéristiques utilisées pour décrire les points clés et enfin les résultats obtenus.
- Chapitre 4 : Dans ce chapitre nous avons présenté notre contribution principale dans cette thèse qui est la proposition d'un nouveau descripteur de point-clés appelé SDIK (Spatial distribution of Ink at Keypoints). Nous avons présenté de manière détaillée le nouveau descripteur SDIK ainsi un algorithme pour le calcul de similarité entre une image requête et une ligne du document cible. Les bases d'images utilisées pour l'évaluation et les résultats obtenus sont également présentées dans ce chapitre.
- Conclusion générale : c'est la synthèse de tous les travaux réalisés dans le cadre de cette thèse, elle inclue les perspectives et les futurs travaux à envisager.

Le repérage de mots dans les images de documents.

Dans ce chapitre, nous présentons le principe de fonctionnement des méthodes de repérage de mots. Un système de repérage de mots est un processus à plusieurs étapes à savoir : la binarisation des images, la segmentation des images en des entités plus petites (ligne, mots et sous-mots), l'extraction des caractéristiques et le repérage des requêtes. Nous présentons une revue de la littérature de chaque étape, nous présentons aussi les différentes classifications des méthodes de repérage de mots dans la littérature.

1.1. Classification des méthodes de repérage de mots

Les systèmes de repérage de mots peuvent être classés selon différents points de vue : selon le format de la requête, selon l'algorithme de recherche utilisé (avec ou sans apprentissage) et selon les prétraitements que subissent les pages du document (avec ou sans segmentation). Les techniques de repérage de mots peuvent être classées en deux groupes en fonction du format de la requête : Query-By-String (QBS) et Query-By-Example (QBE)[1-4]. Dans les systèmes à requête texte (QBS) [5, 6], l'utilisateur saisit la requête sous forme de texte. Ces approches sont issues des systèmes de reconnaissance de l'écriture (OCR), elles essaient de reconnaître les caractères de la requête dans les images du document. D'autre part, les méthodes à base d'exemple QBE [7, 8], l'utilisateur doit sélectionner manuellement une image du mot requête dans le document cible. La requête est une image qui est appariée avec les images des mots du document cible. Certaines méthodes à base d'exemple acceptent le format texte pour la requête à partir duquel une image est synthétisée pour être utilisée dans la recherche [9, 10].

Les techniques de repérage de mots peuvent être classées par rapport à l'utilisation de l'apprentissage artificiel. Les méthodes à base d'apprentissage

(learning-based)[5, 11-13] acceptent généralement des requêtes texte arbitraires. Ils sont capables de gérer les variations liées au style de l'écriture, mais ils nécessitent de larges bases de données étiquetées manuellement pour l'entraînement du système. Les méthodes sans apprentissage (learning-free) [14-17] utilisent la comparaison d'images (image matching) pour identifier les images des mots qui sont similaires à la requête. Ces méthodes sont très pratiques dans le cas où les données d'apprentissage sont difficiles à obtenir tel est le cas des anciens manuscrits.

Les méthodes de repérage de mots sont également classées en fonction des prétraitements effectués sur le document. Deux types de méthodes peuvent être distingués dans la littérature : les approches basées sur la segmentation et les approches sans segmentation. Les méthodes basées sur la segmentation utilisent différents niveaux de segmentation, en lignes, en mots ou en sous-mots. Dans le cas des documents arabes, les frontières entre les mots ne sont pas connues. Le document est généralement segmenté en composantes connexes appelées par certains auteurs parties de mots PAW (Part of Word)[12]. D'autre part, les méthodes sans segmentation, représentent le document avec des vecteurs de caractéristiques extraits à partir de petites zones de l'image (patches) [7, 14, 18] et le repérage de la requête se fait dans l'espace des caractéristiques.

1.2. L'architecture d'un système de repérage de mots

Un système de repérage de mots nécessite plusieurs étapes de préparation et de prétraitement (Figure 1.1). La première étape consiste en l'acquisition des images en utilisant des équipements spécialisés dans la numérisation. Dans une deuxième étape, les images acquises subissent une opération de nettoyage et de binarisation dans le but de distinguer les pixels qui représentent le texte des pixels de l'arrière-plan. Une fois les images de document sont nettoyées, les images des pages sont segmentées en ligne, en mots ou en sous-mots, selon l'algorithme de recherche utilisé. Les images des pages segmentées ainsi que la requête de l'utilisateur sont représentées par des vecteurs numériques qui représentent les caractéristiques de ces images. Finalement, un algorithme de recherche est utilisé pour repérer la requête dans les images du document en utilisant leurs caractéristiques.

Dans les sections suivantes, nous présentons une revue de la littérature pour chacune des étapes d'un système de repérage de mots.

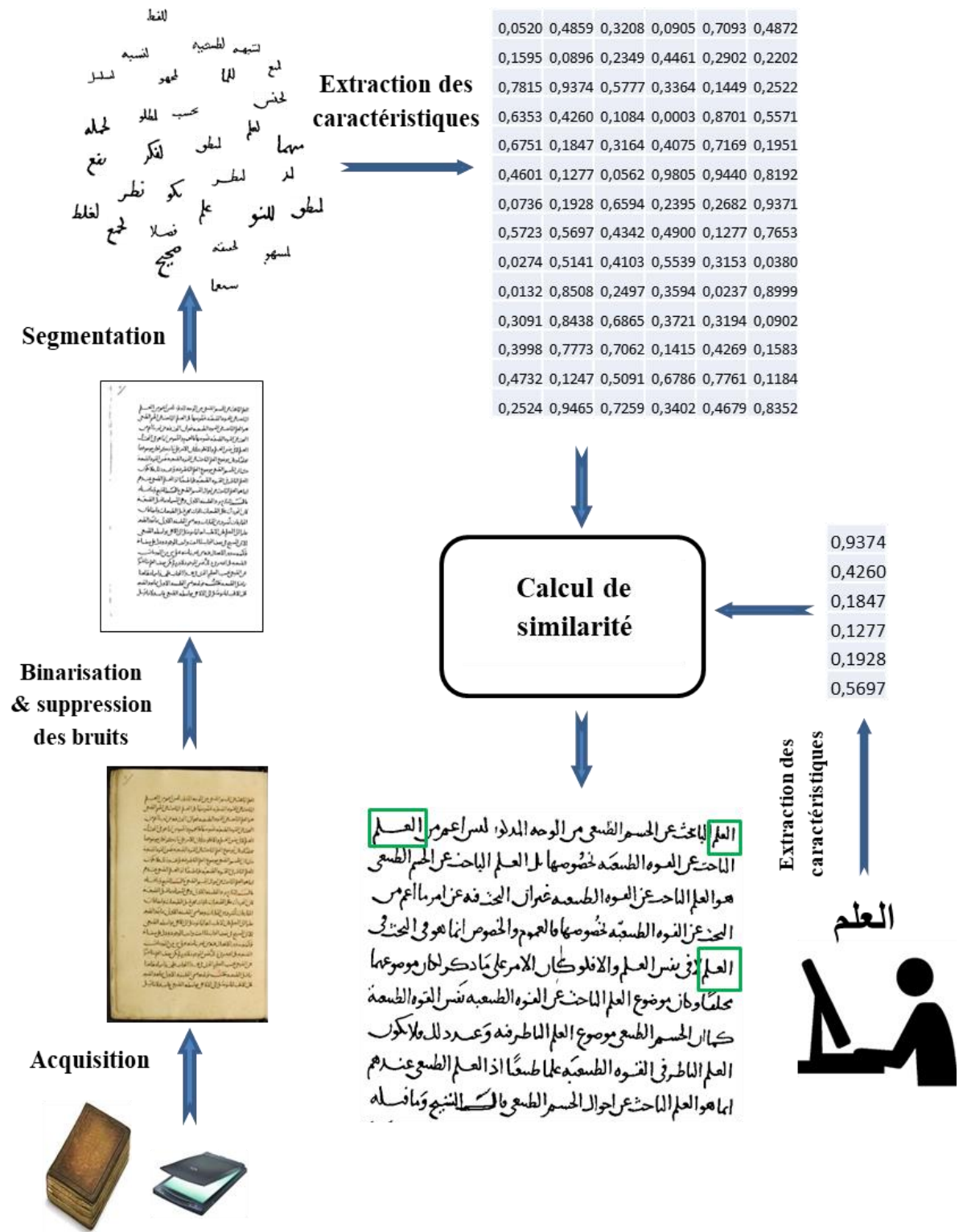


Figure 1.1. Architecture générale des systèmes de repérage de mots.

1.3. Binarisation

L'objectif de la binarisation est de produire à partir des images en niveaux de gris, des images binaires contenant deux types de pixels, des pixels noirs pour le texte et des pixels blancs pour l'arrière-plan. La figure 1.2 montre un exemple d'images binarisées. La binarisation est la première étape pour la majorité des méthodes de repérage de mots, elle est obligatoire pour certaines méthodes tel que les méthodes nécessitant la segmentation du document en lignes, en mots ou en composantes connexes (connected components). La binarisation est nécessaire aussi pour les méthodes qui extraient des caractéristiques à partir du contour ou du squelette de l'écriture. Plusieurs méthodes de binarisation sont appliquées aux images de document, nous présentons dans cette section quelques méthodes à savoir : la binarisation par seuillage, la détection de contours, les méthodes utilisant des modèles statistiques et finalement la classification des pixels.

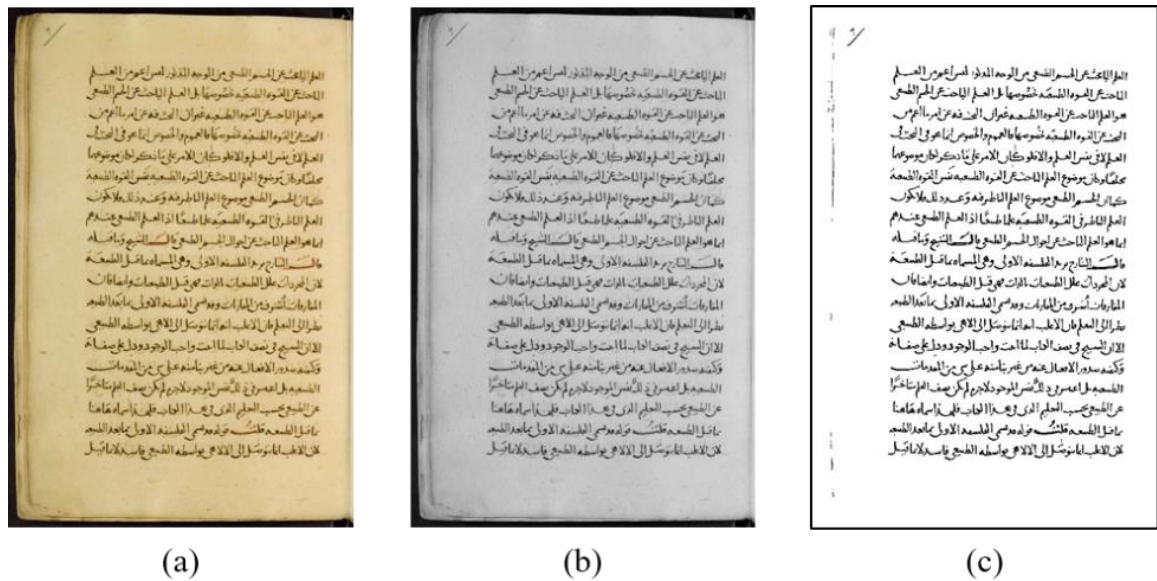


Figure 1.2. Exemple de binarisation d'image de document ; (a) l'image originale : page 11 de la base IBN SINA [19], (b) l'image afficher en niveau de gris, et (c) l'image binaire.

1.3.1. Binarisation par seuillage

Nous distinguons deux types de méthodes de binarisation par seuillage : les méthodes à seuil global et les méthodes à seuil local et parfois une hybridation des deux seuils est utilisée. La binarisation par seuillage global consiste à déterminer une valeur de seuil et tous les pixels ayants leurs intensités inférieures au seuil sont considérées comme du texte et les autres pixels appartiennent à

l'arrière-plan de l'image. La méthode d'Otsu [20] est l'ancêtre de la plupart des méthodes de binarisation par seuillage global. Cette méthode de binarisation est adoptée par plusieurs auteurs de méthodes de repérage de mots [21-23]. Cette méthode sélectionne une valeur de seuil globale parmi tous les seuils possibles comme étant celle qui minimise la variance intra-classe des pixels noirs et blancs. Can et al [24] ont utilisé une méthode de binarisation très simple et ils ont obtenus des résultats similaires à ceux de la méthode d'Otsu. Ils ont utilisé l'intensité moyenne des pixels de l'image en niveaux de gris comme un seuil global. Un inconvénient majeur des méthodes de seuillage global est que l'arrière-plan peut être non uniforme, des pixels de l'arrière-plan peuvent être plus sombres que certains pixels de premier plan (Figure 1.3). Par conséquent, aucune méthode de seuillage global ne peut parfaitement binariser une telle image.

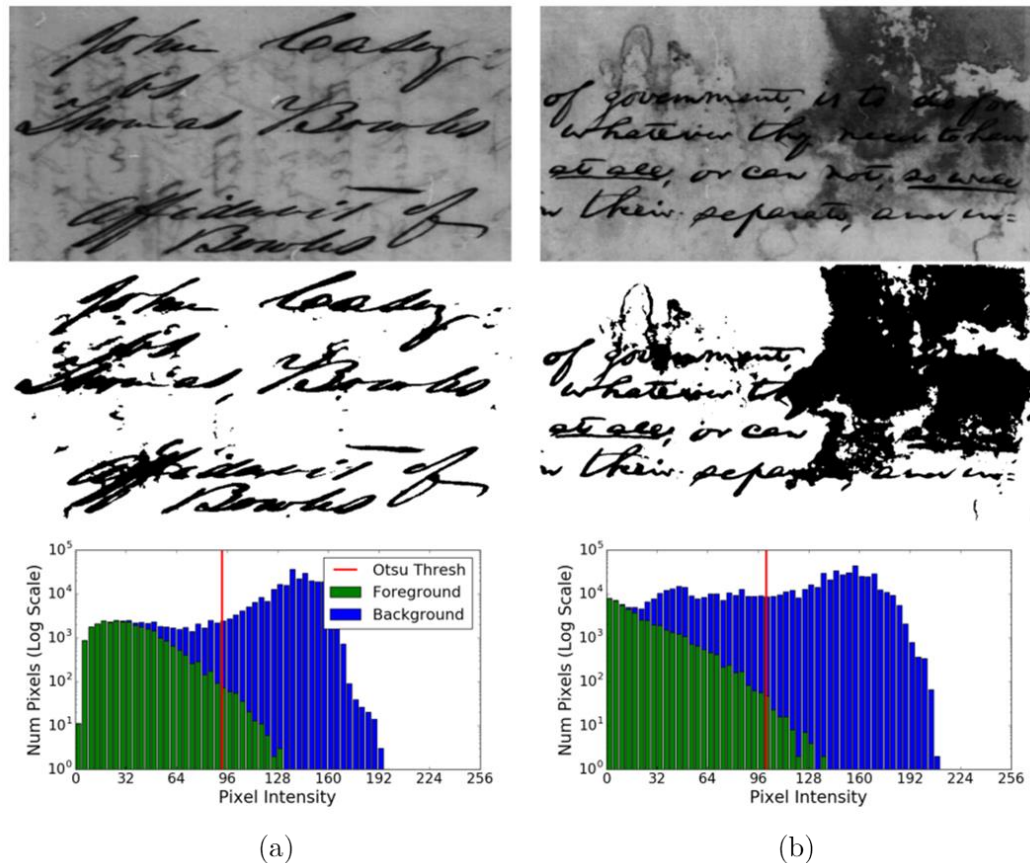


Figure 1.3. Exemples d'images binarisées avec la méthode d'Otsu, (a) image de document avec fond uniforme, (b) image de document avec arrière-plan irrégulier [25].

Les anciens documents souffrent généralement de plusieurs types de dégradations comme l'éclairage non uniforme, la variation du contraste de l'image, l'effet de transparence et autres. Dans ce cas, des techniques de seuillage local plus efficaces sont nécessaires. La binarisation par seuillage local utilise un seuil

propre à chaque pixel de l'image. Le seuil local d'un pixel est calculé en fonction de l'intensité des pixels avoisinants. Par exemple, la méthode de Sauvola [26] calcule le seuil local d'un pixel en utilisant la valeur moyenne et l'écart type de l'intensité des pixels avoisinants. Les zones locales qui participent dans le calcul des seuils sont généralement sélectionnées par une fenêtre glissante qui explore la totalité de l'image. La binarisation par seuillage local est adoptée par plusieurs auteurs de méthodes de repérage de mots dans les images de document [27-29].

D'autres approches de binarisation de documents dégradés reposent sur des techniques hybrides qui combinent un seuillage global avec un seuillage local. La méthode proposée dans [30] utilise trois méthodes de seuillage global pour extraire les régions d'intérêt des images en niveaux de gris. Chaque région d'intérêt est ensuite traitée par une version modifiée de l'algorithme de Sauvola avec des tailles variables de la fenêtre glissante.

1.3.2. Détection de contour

Les méthodes de binarisation par détection de contour utilisent le contour de l'écriture comme frontières entre le texte et l'arrière-plan. La méthode populaire de Canny [31] pour la détection de contour est utilisée pour la binarisation des images de documents [32]. La méthode de Canny utilise l'image de l'amplitude du gradient et effectue un seuillage par hystérésis. Pour cela on fixe deux seuils, un seuil haut S_h et un seuil bas S_b . On commence par sélectionner les points qui dépassent le seuil S_h et on les considère comme des pixels de contour. Les pixels ayant l'amplitude du gradient entre le seuil bas et le seuil haut, qui sont connectés à des pixels de contour sont ajoutés au contour. Les pixels ayant leur amplitude du gradient inférieure à S_b sont rejetés. Une autre méthode utilisant la détection de contours est proposé dans [33] pour la binarisation des images de document, la méthode utilise l'algorithme de montée des eaux. Une image est considérée comme une surface tridimensionnelle, avec les niveaux de gris représentant l'altitude. Ainsi, un pixel noir représente une vallée et un pixel blanc une montagne. L'image est inondée par de l'eau et les parties couvertes par l'eau et encerclées par des contours sont désignées comme étant du texte. La figure 1.4 montre un exemple d'images binarisées avec la méthode proposée dans [33].

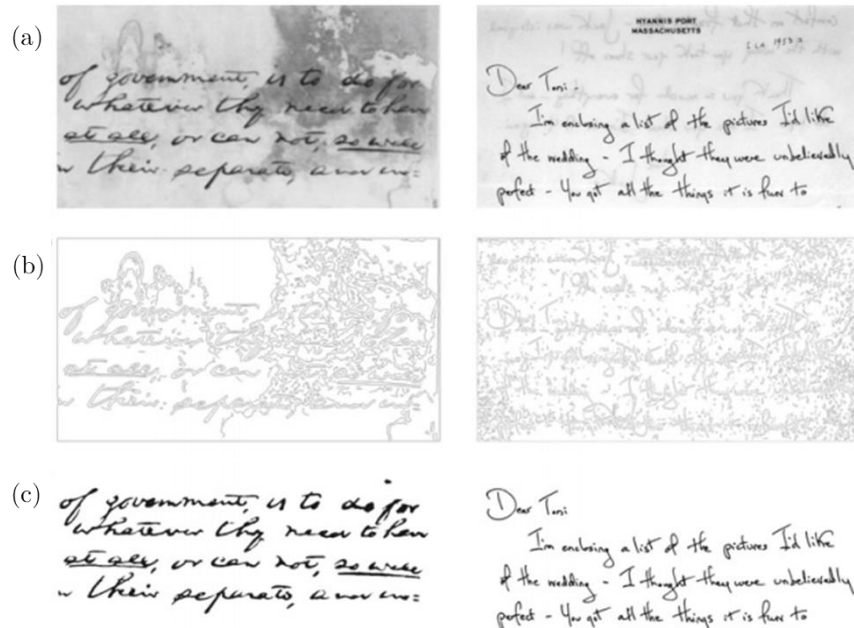


Figure 1.4. Exemple de binarisation d'image de document par détection de contour et la montée des eaux [33]. (a) l'image originale, (b) les contours détectés par la méthode de Canny, (c) l'image binaire.

1.3.3. Modèles statistiques

Les modèles statistiques sont utilisés dans la binarisation d'image comme des outils pour partitionner les pixels de l'image en deux clusters, les pixels du texte et les pixels de l'arrière-plan. Les mélanges de gaussiennes GMM (Gaussian mixture model) et les champs aléatoires conditionnels CRF (Conditional Random Fields) sont appliqués à la binarisation d'image de documents.

Un mélange de gaussienne GMM est un modèle probabiliste défini par une somme de lois normales telle que défini dans l'équation (1.1)

$$F(x) = \sum_{j=1}^K w_j N(\mu_j, \sigma_j) \quad (1.1)$$

Avec N est une loi normale, K est le nombre de composantes du modèle et $\sum_{j=1}^K w_j = 1$. Les paramètres du modèle sont ajustés par l'algorithme de maximisation de la vraisemblance EM (Expectation Maximization). La composante $w_j N(\mu_j, \sigma_j)$ est la probabilité pour que x appartient à la classe j . La manière d'utiliser les GMMs pour la binarisation d'image diffère d'un auteur à un autre. Par exemple, les deux méthodes proposées dans [34, 35] ont utilisé un modèle

de mélange de Gaussiennes à deux composantes pour modéliser les pixels de texte et ceux de l'arrière-plan.

Les champs aléatoires conditionnels CRF fournissent un outil pour inclure les dépendances spatiales entre les pixels dans le processus de binarisation. Un CRF est guidé par une fonction d'énergie qui évalue le degré de correspondance entre une image binarisée et l'image d'entrée (Equation 1.2).

$$E(B, I) = \sum_{i=1}^n E_i(B_i, I) + \sum_{i=1}^n \sum_{j=1}^i E_{ij}(B_i, B_j, I) \quad (1.2)$$

Avec I est l'image d'entrée ayant n pixels, B est une image binarisée de I . E_i et E_{ij} sont les fonctions de cout. En générale un CRF peut segmenter le corpus de donnée en plusieurs classes, pour la binarisation seulement deux classes sont utilisées, E_1 indique l'appartenance d'un pixel au texte, et E_2 exprime la vraisemblance entre les pixels voisins. La fonction E_2 donne de faibles valeurs lorsque des pixels adjacents et similaires sont affectés à la même classe. Binariser une image avec un CRF consiste à minimiser la fonction d'énergie (Equation 1.3).

$$\hat{B} = \underset{B}{\operatorname{argmin}} E(B, I) \quad (1.3)$$

La méthode proposée dans [36] effectue une pré-binarisation de l'image en classifiant les pixels de l'image en trois types de pixels : les pixels d'encre, les pixels de fond et les pixels incertains. Un CRF est utilisé par la suite pour classifier les pixels incertains. La figure 1.5 affiche un exemple de binarisation par la méthode proposée dans [36].



Figure 1.5. Binarisation d'images avec un CRF, (a) image originale, (b) image pré-binarisée (pixels incertain en bleu), (c) image binarisée avec la méthode de Sauvola, (d) binarisation optimisée par un CRF [36].

1.3.4. Classification des pixels

Au cours des dernières années, les modèles d'apprentissage profond (deep learning) sont très utilisés pour le traitement d'image en générale. Pour la binarisation d'image, les réseaux de neurones sont utilisés pour la classification des pixels dans deux classes : pixels d'encre et pixels de fond. Les premières utilisations des réseaux de neurones convolutifs CNN (Convolutional Neural Network) impliquaient de petites images pour la reconnaissance de caractères, puis ils sont devenus populaires pour la classification d'images de grandes tailles.

La méthode proposée dans [37] utilise un réseau entièrement connectés FCN (Fully Connected Networks) pour la classification des pixels de l'image d'entrée. La binarisation est un problème de classification avec un nombre de classes $K=2$, et les méthodes basées sur les FCN se sont avérées très efficaces pour cette tâche. Dans une autre méthode [38] un réseau à convolution (CNN) est utilisé et a démontré que les réseaux CNN sont plus performants dans la binarisation d'images de document que les réseaux de type FCN.

La méthode proposée dans [39] a adopté une architecture de type encodeur/décodeur. L'encodeur échantillonne progressivement l'image d'entrée pour extraire une couche de caractéristiques représentant l'image d'entrée. Le décodeur effectue ensuite un sur-échantillonnage de la couche de caractéristique pour produire une image binarisée de même taille que l'image d'entrée. L'architecture du réseau utilisé est présentée dans la figure 1.6.

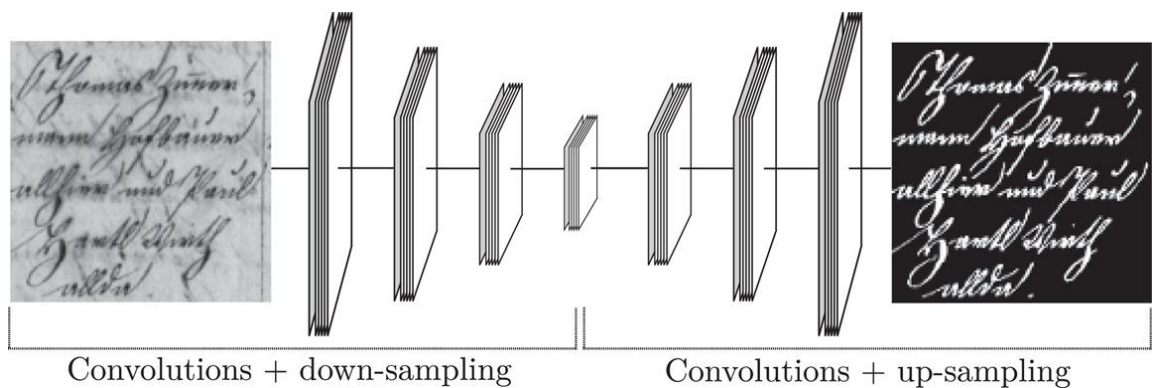


Figure 1.6. Binarisation par un réseau de neurone de type encodeur/décodeur [39].

1.4. La segmentation

Selon la méthode de repérage de mots adoptée, le document cible doit être segmenté en lignes, en mots ou en composantes connexes. Dans le cas des documents imprimés, la segmentation s'avère plus simple car la distance entre les lignes ainsi que les espaces entre les mots sont uniformes. La segmentation est plus contraignante dans le cas des documents manuscrits, dans de tel manuscrits la taille de l'écriture est irrégulière, les lignes ainsi que les mots peuvent se chevaucher.

1.4.1. Segmentation en lignes

La segmentation en lignes consiste à déterminer des zones dans l'image du document qui représentent des lignes de texte. Nous distinguons plusieurs approches :

La segmentation avec le profil de projection horizontale [40]. Dans l'histogramme de la projection horizontale, les maximas représentent les lignes de texte et les minimas correspondent aux espaces entre les lignes. Ces méthodes sont plus adaptées aux documents imprimés et peuvent être utilisé pour segmenter les documents manuscrits qui sont transcrits en ligne de texte.

La technique de lissage [41] consiste à noircir les petits espaces entre les pixels noirs consécutifs sur la direction horizontale, ce qui aura pour résultat de les connecter. Les boîtes qui englobent les composantes connexes successives dans l'image lissée forment les lignes.

La segmentation par regroupement [42] construit des lignes de texte à partir d'éléments de bas niveau tel que les pixels et les composantes connexes. Les méthodes basées sur la transformée de Hough [43] utilisent une projection d'un ensemble de points dans l'espace de Hough pour détecter des lignes de texte.

1.4.2. Segmentation en mots

Certaines méthodes utilisant les projections horizontales pour la séparation des lignes de texte suivent le même principe pour la segmentation en mots. Les profils verticaux sont utilisés pour identifier les frontières entre les mots. Rodriguez-Serrano et Perronin [44] utilisent des profils de projection horizontale pour obtenir des lignes de texte. Pour chaque ligne, ils calculent les enveloppes des composants

connexes et définissent une distance entre les composants voisins comme la distance minimale entre leurs enveloppes. Les distances supérieures à un seuil donné sont considérées comme les frontières entre les mots. Un exemple de leur processus de segmentation est présenté dans la figure 1.7.

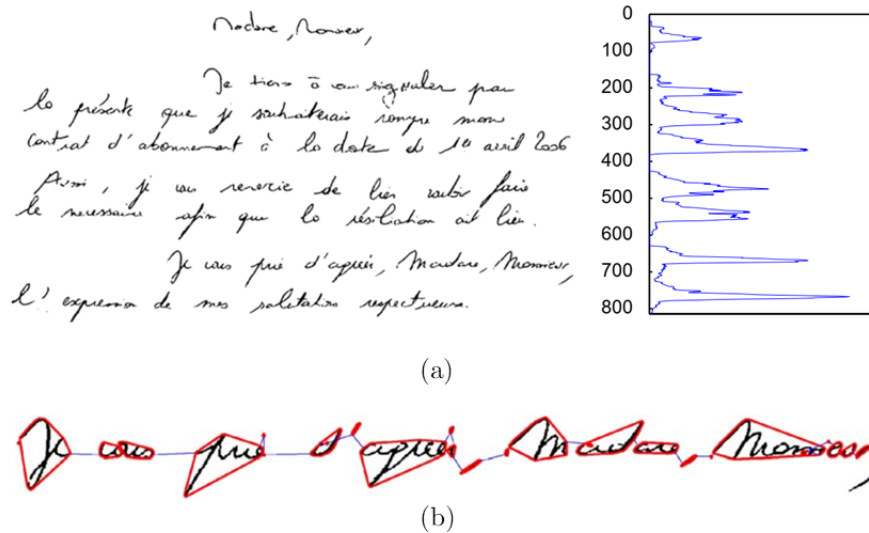


Figure 1.7. Segmentation en mots avec la méthode de [44]. (a) extraction des lignes, (b) segmentation en mots avec les distances entre les enveloppes des composants connexes.

L'algorithme RLSA (Run Length Smoothing Algorithm) [45] est une technique qui consiste à faire un double lissage unidirectionnel de l'image à segmenter selon les directions horizontale et verticale en utilisant deux seuils. Le premier seuil pour le lissage horizontal est généralement la moyenne des espaces inter mots. Le deuxième seuil du lissage vertical est égal à l'espace interligne. Le résultat de l'algorithme RLSA est une image où les caractères sont regroupés dans des composants connexes. Ensuite, une analyse des composants connexes est effectuée afin de segmenter le document en mots. L'algorithme RLSA est adopté par plusieurs méthodes de repérage de mots [46-48]. L'algorithme RLSA est bien adapté aux documents imprimés mais moins performant avec les documents historiques manuscrits où les espaces entre les mots et les lignes sont variables.

1.4.3. Segmentation en sous-mots

La segmentation en sous-mots est très fréquente dans les méthodes de repérage de mots dans les documents arabe. Les frontières entre les mots sont impossibles à reconnaître visuellement elles dépendent de la sémantique du texte. La plupart des méthodes de repérage de mots dans les documents arabes [29, 49-51] effectuent une segmentation en composants connexes CCs appelées aussi parties de mots

PAW (Part of Word). Khayat et al.[12] ont proposé un algorithme pour l'extraction des PAWs d'une ligne de texte. Ils considèrent qu'une partie de mot arabe est composée d'une composante connexe majeure et de quelques composantes mineures (points diacritiques). L'algorithme commence par distinguer les CCs majeures par leurs tailles et leurs emplacements, et il associe ensuite les CCs mineures à leurs CC majeure voisine la plus proche. La figure 1.8 montre un exemple de parties de mot composées de CCs majeures et mineures.



Figure 1.8. Un mot arabe et ses trois PAWs. Les têtes des flèches pointent vers des CCs majeures, les autres CCs sont des composantes mineures[12].

1.5. L'extraction des caractéristiques (features extraction)

Une image de document est une matrice à valeurs numériques difficile à interpréter. L'image doit être convertie dans un autre espace plus représentatif. Une image est représentée par un ensemble de caractéristiques dans le but de capter les propriétés discriminatives de l'écriture. Cette étape est très importante pour un système de repérage de mots. Nous distinguons deux approches principales dans la littérature, les approches *globales* et les *approches* locales.

Les approches globales extraient un seul vecteur de caractéristiques pour toute l'image. Comme exemples de ces caractéristiques, la largeur et la hauteur de l'image, le rapport hauteur/largeur, le nombre de pixel noir et d'autres. Les approches globales sont limitées en performances dans le repérage de mots dans les images de document.

Les approches locales extraient un vecteur de caractéristiques dans différentes régions locales de l'image. Ces caractéristiques peuvent être extraites avec une fenêtre glissante, dans des régions d'intérêts ou dans des patches définis avec une grille superposée sur l'image. La majorité des méthodes d'extraction des attributs utilisées dans le traitement d'image sont appliquées au repérage de mots.

Le choix des caractéristiques à utiliser pour représenter les images du document est très crucial pour un système de repérage de mots. La forme de

l'écriture diffère d'une langue à une autre et les caractéristiques utilisées doivent capturer les informations discriminantes de l'écriture. De ce fait, le choix des caractéristiques a un lien direct avec la langue du document. Une revue de la littérature sur l'extraction des caractéristiques est présentée dans le prochain chapitre.

1.6. Calcul de similarité

Le repérage de mots se fait dans l'espace des primitives, les caractéristiques de la requête sont appariées (matchées) avec les caractéristiques des mots ou des lignes du document cible. Nous avons distingué deux approches principales dans la littérature, des méthodes qui calculent une dissimilarité entre le mot requête et les mots du document cible en utilisant des fonctions de distance et des méthodes qui utilisent des modèles à base d'apprentissage pour la reconnaissance du mot requête dans le document.

1.6.1. Méthodes à base d'apprentissage.

Les méthodes de repérage de mots à base d'apprentissage sont issues des systèmes de reconnaissance de l'écriture (OCR). Les caractéristiques extraites à partir des images de mots ou de lignes sont modélisées par des modèles statistiques tel que : les modèles de Markov cachés HMM (Hidden Markov Model) et les réseaux de neurones ANN (Artificial Neural Network).

Rodríguez et al [44] ont utilisé un HMM pour modéliser les séquences de vecteurs de caractéristiques extraites à l'aide d'une fenêtre glissante de la requête et des images des mots de l'ensemble de données. Les auteurs utilisent plusieurs instances d'une requête pour l'entraînement des HMMs. Lors du processus d'appariement de la requête avec les images des mots du corpus de données, la similarité entre la requête et une image de mot est obtenue par la probabilité postérieure produite par l'HMM de la requête. Leur méthode ne peut pas repérer les mots hors du vocabulaire considéré lors de la phase d'apprentissage. Dans un autre travail [52], ils ont remédié à cet inconvénient par un HMM semis-continu (SC-HMM). En effet, les paramètres du modèle sont estimés de manière non supervisée sur un pool d'échantillons qui permettent au modèle de s'adapter à l'image de la requête.

Une méthode basée des HMMs de caractères pour la repérage de mots dans un texte manuscrit est proposée par Fischer et al [53]. Chaque image de ligne de texte est représentée par une séquence de vecteurs de caractéristiques qui est obtenue par une fenêtre glissante. Des HMMs de caractères sont entraînés à l'aide d'images de lignes de texte étiquetées. Ensuite, les modèles de lignes de texte sont générés avec la concaténation des séquences de modèles de caractères (character-HMMs). Dans la phase en ligne, une requête textuelle est représentée par un HMM de mot qui est construit à partir des HMMs de caractère. Le score final de d'appariement de la requête et de la ligne de texte est obtenu par une fonction de similarité entre le modèle de la requête et des modèles des lignes de texte.

Les réseaux de neurones sont devenus populaires dans le traitement d'images en générale et le traitement des images de document en particulier. Zhong et al. [54] ont utilisé un réseau de neurones qui reçoit deux images de mots en entrée et génère directement un score de similarité pour les deux images. Le point faible des réseaux de neurones est le besoin de large base d'images étiquetées pour l'apprentissage des paramètres du réseau. Afin de faire face à cette contrainte, les auteurs ont utilisé la technique d'augmentation de l'ensemble d'apprentissage. Cette technique consiste à appliquer des transformations sur les images de la base d'apprentissage pour créer de nouvelles images.

Une méthode à base des réseaux de neurones de type BLSTM (Bidirectional Long-Short Term Memory) est proposée dans [55] pour le calcul de similarité entre un mot de requête et des lignes de texte. Le réseau est constitué d'une couche d'entrée, une couche cachée et une couche de sortie. La couche d'entrée contient un neurone pour chaque composante du vecteur de caractéristiques qui sont extraites avec une fenêtre glissante. La couche cachée est constituée de cellules de mémoire à long terme (LSTM) et la couche de sortie contient un neurone pour chaque caractère de l'alphabet. La sortie du réseau est donc une matrice de probabilités indiquant la probabilité que chaque lettre apparait à une position particulière. Pour repérer un mot de requête à l'intérieur d'une ligne de texte, l'algorithme trouve le meilleur chemin à travers la matrice de sortie qui passe par les lettres du mot recherché. La dissimilarité de la requête avec la ligne de texte est le produit de toutes les probabilités du chemin optimal qui est normalisé par la longueur du mot de requête.

Les méthodes de repérage de mots dans les images de document à base d'apprentissage sont très performantes lorsque la base d'apprentissage est suffisamment large, elles admettent généralement des requêtes textuelles arbitraires (QBS). Dans le cas du repérage de mots dans les images des documents anciens, ces méthodes partagent tous le même inconvénient qui est le manque de base d'images étiquetées pour l'entraînement du système.

1.6.2. Méthodes sans apprentissage.

Les méthodes sans apprentissage sont souvent de type QBE (Query-By-Example) où le mot de requête est une image. Des vecteurs de caractéristiques sont extraits à partir de la requête et des mots du document. Une dissimilarité est calculée entre les vecteurs de caractéristiques de la requête et ceux des mots du document cible qui sont souvent de tailles différentes. L'algorithme d'alignement de séquences DTW (Dynamic Time Warping) est conçu pour le calcul de dissimilarité entre des séries temporelles de tailles différentes. L'algorithme DTW est utilisé pour la première fois par Rath et Manmatha [56] pour le repérage de mots dans des images de documents. Les pages du document sont segmentées en images de mots qui sont représentées par des vecteurs de caractéristiques extraites à chaque colonne de l'image. Les vecteurs de caractéristiques sont donc de tailles différentes et l'algorithme DTW est adapté au calcul de dissimilarité entre le vecteur de caractéristiques du mot de requête et ceux des mots du document cible. L'algorithme DTW est adopté par plusieurs auteurs de méthodes de repérage de mots [57-60].

Plusieurs variantes de l'algorithme DTW sont proposées pour le repérage de mots. Terasawa et Tanaka [61] ont segmenté les images du document en lignes de texte et ils ont utilisé une fenêtre glissante pour extraire les vecteurs de caractéristiques. Des séquences de longueurs variables représentant la requête et les lignes de texte sont obtenues. Ces vecteurs de caractéristiques sont ensuite appariés à l'aide d'une technique basée sur DTW qui utilise la programmation dynamique continue CDP (Continuous Dynamic Programming). L'algorithme calcule les similarités entre la séquence de la requête et toutes les sous-séquences possibles d'une ligne de texte. Mondal et al. [17] ont proposé une technique d'appariement de séquences flexibles FSM (Flexible Sequence Matching) qui est basée sur DTW. Leur algorithme a la capacité de récupérer le mot de requête complet ou une partie. De plus, l'algorithme FSM est moins sensible aux

variations locales de l'orthographe des mots et aux effets de dégradation au sein des images du document.

1.7. Conclusion

Dans ce chapitre nous avons présenté les différentes étapes d'un système de repérage de mots. Nous avons présenté quelques méthodes de binarisation des images de document ainsi que quelques méthodes de segmentation. L'étape d'extraction de caractéristiques n'est pas traitée en détails dans ce chapitre car elle fait l'objet d'un chapitre à part entière. Un état de l'art sur les méthodes d'extraction des caractéristiques est présenté dans le chapitre suivant.

Extraction des caractéristiques : revue de la littérature.

L'extraction des caractéristiques est une étape cruciale dans un system de repérage de mots dans les images de document. Dans ce chapitre nous présentons les différentes caractéristiques utilisées dans la littérature pour représenter une image de document. Nous présentons les caractéristiques utilisées dans la littérature en générale et nous nous sommes focalisés sur les méthodes de repérage de mots dans les images de documents Arabes.

2.1. Méthodes de repérage de mots dans les documents latins

Le repérage de mots dans les images de documents des langues latines est largement implémenté dans la littérature. Nous présentons ci-après une liste non exhaustive des caractéristiques utilisées.

2.1.1. Les profils des mots

Ces caractéristiques sont utilisées pour la première fois par Rath et Manmatha [56, 62] et réutilisées ensuite par plusieurs auteurs. Ces primitives sont extraites dans chaque colonne de l'image d'un mot (figure 2.1), ils comprennent :

- Le profil supérieur (upper profile) : c'est la position du pixel d'encre le plus haut.
- Le profil inférieur (lower profile) : c'est la position du pixel d'encre le plus bas.
- Le nombre de transitions entre l'encre et l'arrière-plan (ink to background transitions).
- Le profil de projection : c'est le nombre de pixels noirs dans la colonne (number of foreground pixels).

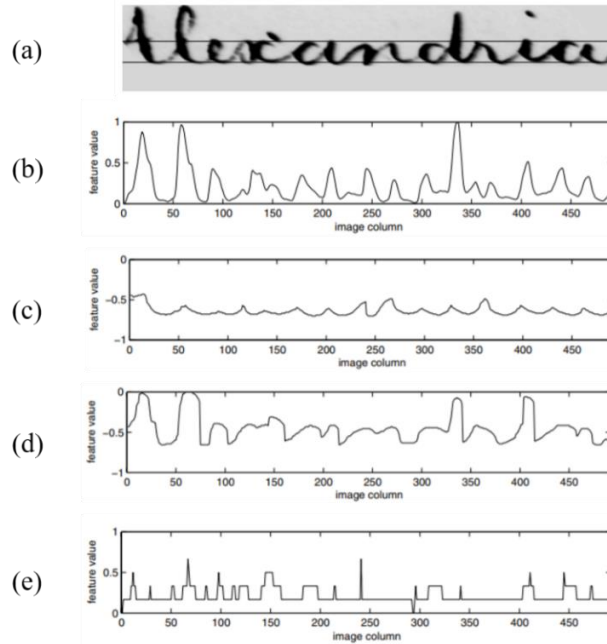


Figure 2.1. Profils de mot. (a) image originale, (b) profil de projection, (c) profil inférieur, (d) profil supérieur, et (e) transitions entre l'ancre et le fond [63].

Les profils de mot sont combinés avec d'autres types de caractéristiques. Les méthodes proposées dans [55, 64-66] ont utilisé des caractéristiques locales à base des profils extraites avec une fenêtre glissante qu'ils ont combiné avec des caractéristiques globales. Les moments des pixels noirs sont utilisé comme attributs globaux (les moments d'ordre 0, 1 et 2) et les attributs locaux sont : le profil supérieur et le profil inférieur, l'inclinaison du contour le plus haut et celle du contour le plus bas, le nombre de transition entre l'encre et l'arrière-plan, et le niveau de gris moyen des pixels d'encre dans la fenêtre.

Kesidis et al [48] ont utilisé les profils de mot en combinaison avec la densité des pixels. L'image de mot est divisée en un ensemble de zones et la densité des pixels d'encre ainsi que les caractéristiques des profils sont calculées dans chaque zone.

2.1.2. Caractéristiques de zonage

Les images de document sont divisées en petites zones en utilisant généralement une grille. Dans chaque zone de l'image des caractéristiques statistiques sont calculées telle que la densité des pixels noirs[67]. Les caractéristiques de zones ont donné des résultats satisfaisants dans les documents imprimés mais ils sont moins performants dans les documents manuscrits. La figure 2.2 illustre un exemple de découpage d'une image en zones pour y extraire des caractéristiques. Les

caractéristiques de zones sont utilisées par plusieurs auteurs des méthodes de repérage de mots [48, 68, 69].

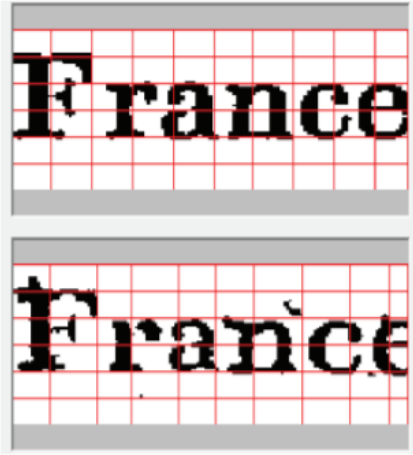


Figure 2.2. Exemple de division d'une image en zones [67].

2.1.3. Caractéristiques à base du Gradient

Les caractéristiques basées sur le gradient sont largement utilisées dans le repérage de mots dans des images de documents. Cette famille de caractéristiques est plus performante dans les documents multi-scripteurs, car le gradient permet de capter l'orientation des traits dans l'écriture, qui est une propriété très discriminante dans la comparaison des images de mots. Les exemples typiques de ce type de caractéristiques sont le HoG (Histogram of Oriented Gradient)[70] et le SIFT (Scale Invariant Feature Transform) [71]. Ces deux caractéristiques extraient des histogrammes de l'orientation du gradient dans des régions locales. La différence entre les deux est que le HoG est calculé en partitionnant l'image en zone local de manière uniforme, un exemple de calcul des histogrammes HOG est présenté dans la figure 2.3. Tandis que les caractéristiques SIFT sont calculées dans des régions d'intérêt comme les points-clés par exemple. Plusieurs variantes des caractéristiques HoG et SIFT ont été utilisées avec succès pour le repérage de mots [14, 61, 72, 73].

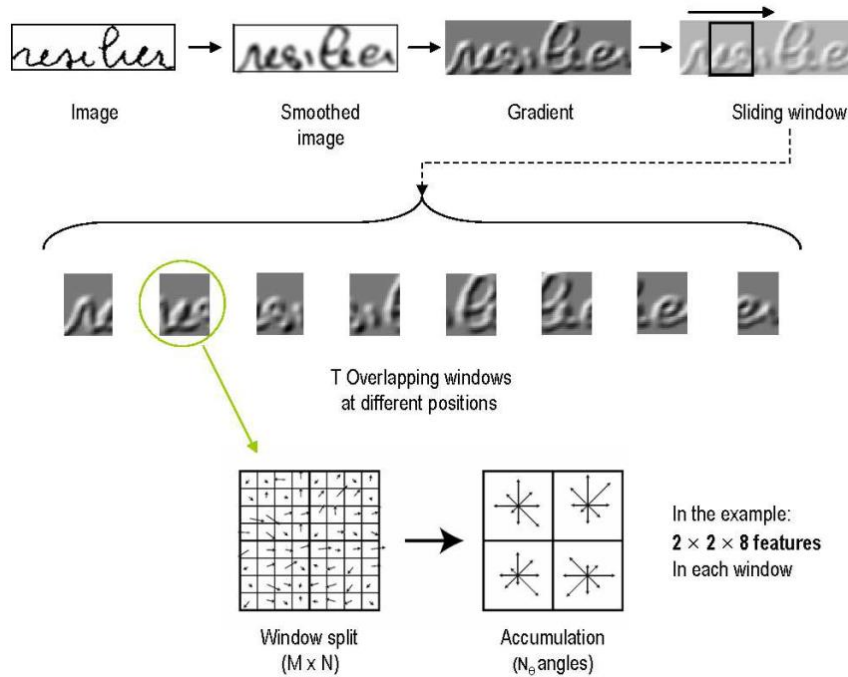


Figure 2.3. Extraction des histogrammes de l'orientation du gradient[59].

2.1.4. Caractéristiques de type motif

Les caractéristiques de motif permettent de détecter la présence de motif dans une région locale de l'image. Un exemple de ce type de caractéristique est le LBP (Local Binary Patterns)[74]. Le code LBP est calculé pour chaque pixel de l'image en impliquant ses pixels voisins. Les motifs LBP peuvent capturer plus d'informations locales que les caractéristiques extraites avec une fenêtre glissante. Un histogramme est extrait à partir de l'image en quantifiant le nombre de pixel de chaque code LBP. Dey et al [75] ont extrait des histogrammes des codes LBP à partir des images de mots en niveaux de gris. Pour ajouter de l'information spatiale à leurs histogrammes, l'image est subdivisée en plusieurs quadrants en utilisant le centre de masse de l'image. Les histogrammes LBP extraits de chaque quadrant sont concaténés pour former un histogramme représentant la totalité de l'image. La figure 2.4 présente leur processus d'extraction des histogrammes LBP.

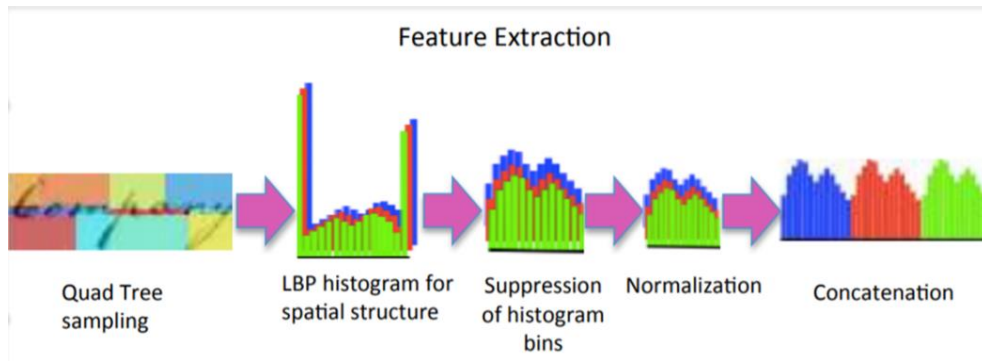


Figure 2.4. Exemple d'extraction des histogrammes LBP[75].

2.1.5. Caractéristiques structurales

Les caractéristiques structurales ont également trouvé leur place dans le repérage de mots, ils décrivent la structure de l'écriture, on distingue les graphèmes, les segments de ligne ou les structures arborescentes. Les caractéristiques structurales peuvent être extraites du contour de l'écriture [16, 24] ou de son squelette [16, 76].

Liang et al [10] extrait des graphèmes à partir des images de mots en utilisant l'image de squelette. La figure 2.5 représente un exemple d'extraction des graphèmes. Chaque graphème est représenté par un ensemble de caractéristiques structurales telle que : le rapport hauteur/largeur, le nombre de pixels, la direction du graphème et autres. Les caractéristiques des graphèmes sont utilisées pour modéliser les images de mots au niveau caractère. Riba et al [15] représentent une image de mot avec un graphe construit avec des graphèmes. Les connexions entre les nœuds du graphe sont déterminées avec les relations spatiales des graphèmes. Le repérage du mot requête s'effectue avec l'appariement de graphes (graph matching).

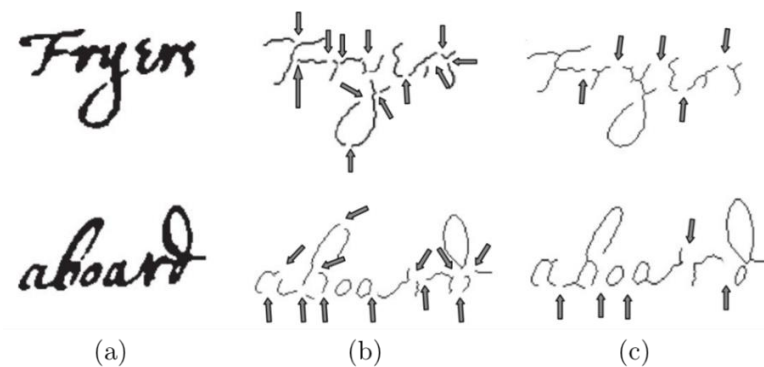


Figure 2.5. Exemple d'extraction des graphèmes [10].

2.1.6. Attributs des réseaux de neurones profonds (deep NN)

Avec l'augmentation de la puissance de calcul des ordinateurs, les réseaux de neurones sont devenus des algorithmes très puissants pour l'apprentissage artificiel. Les réseaux de neurones à convolution (CNN) permettent d'extraire des caractéristiques à partir d'images brutes. Les CNNs effectuent une réduction de la taille de l'image pour générer une image de plus petite taille et plus représentative grâce à une succession de couches de convolution, couches de pooling et des couches de correction ReLU (Rectified Linear Units). Les couches de convolution permettent la détection des caractéristiques avec des filtres de convolution. Les couches de pooling sont souvent placées entre deux couches de convolution, elles permettent de réduire la taille des images, tout en préservant leurs caractéristiques importantes. Les couches de correction ReLU remplacent toutes les valeurs négatives reçues en entrées par des zéros, elles permettent d'activer que les caractéristiques les plus importantes. La dernière couche d'un CNN représente le vecteur de caractéristiques extrait à partir de l'image d'entrée. Les paramètres du réseau sont ajustés par un algorithme d'apprentissage par correction d'erreur en utilisant une base d'images étiquetées. Les méthodes proposées dans [11, 77, 78] ont toutes utilisé un CNN pour l'extraction des caractéristiques.

2.2. Extraction des caractéristiques des documents arabes

Le repérage de mots dans les images de document est largement implémenté pour les scripts Latins mais il y a peu de travaux pour le repérage de mots dans les documents arabes. Dans cette section nous présentons une liste non exhaustive des travaux que nous avons recensés sur le repérage de mots dans les documents arabes. Nous distinguons des méthodes indépendantes du langage et sont appliquées à la langue arabe et des méthodes proposées et dédiées aux documents arabes.

2.2.1. Méthodes indépendantes de la langue

Wshah et al [5] ont proposé un système statistique de repérage de mots dans des images de document. Leur système est indépendant de la langue du document cible. Ils ont testé plusieurs combinaisons des caractéristiques GSC [79] (Gradient Structural Concavity) et l'intensité des pixels [80] et ils ont obtenu de meilleurs

résultats avec la combinaison du gradient avec l'intensité des pixels d'encre. Pour évaluer leur méthode dans les documents arabes, ils ont utilisé une base d'images de documents modernes et les résultats obtenus pour l'arabe sont inférieurs à ceux obtenus avec le script latin.

Rodríguez Serrano et Perronnin [52] ont implémenté la primitives de l'histogramme locale du gradient LGH (Local Gradient Histogram) qu'ils ont proposé dans un travail antérieur [73]. L'amplitude et l'orientation du gradient sont calculées pour chaque pixel de l'image. Ils ont utilisé une fenêtre glissant à travers l'image d'un mot et l'amplitude ainsi que l'orientation du gradient sont quantifiés dans un histogramme pour chaque frame. Ils ont évalué leur méthode avec l'écriture arabe en utilisant la base d'image IFN/ENIT [81] et ils ont obtenus de faibles résultats en comparant avec les autres langues.

Can and Duygulu [24] ont représenté une image d'un mot par un ensemble de segments de lignes extraits à partir du contour de l'écriture. La position, l'orientation et la longueur de chaque segment de ligne sont utilisées comme primitives pour décrire une image de mot. Pour l'évaluation de leur méthode, ils ont utilisé une base de document Ottoman contenant un mélange de documents manuscrit et imprimés.

Mhiri et al [11] ont proposé un système pour la reconnaissance et le repérage de mots à base de réseaux de neurones. Ils ont utilisé un réseau de neurones à convolution (CNN) pour l'extraction des primitives. Les attributs sont extraits directement à partir des images brutes par apprentissage mais leur méthode nécessite une large base d'images étiquetées pour ajuster les paramètres du réseau. Ils ont évalué leur méthode avec la base IFN/ENIT pour la langue arabe et les résultats obtenus sont inférieurs aux résultats obtenus avec les autres langues.

2.2.2. Caractéristiques dédiées à la langue Arabe

Moghaddam et Cheriet [29] ont suivi la première méthode de repérage de mots proposée par Rath et Manmatha [82]. Leur approche évite la segmentation en mots en segmentant le document en composantes connexes (CC). Un vecteur de six caractéristiques a été extrait d'une CC : le rapport hauteur/largeur, la fréquence horizontale, centre de masse vertical, le nombre de points de branchement, le rapport entre la hauteur de la CC et la hauteur de la ligne de la CC et enfin la présence de loupe. Une mesure de similarité à base de la distance

euclidienne et l'algorithme d'alignement de séquences DTW sont utilisés pour la comparaison des CCs. Afin de réduire le temps de recherche, les cartes auto-organisatrices (Self-Organizing maps) ont été utilisées pour regrouper les CCs dans des bibliothèques, puis une requête CC est repérée dans sa propre bibliothèque.

Dans un autre travail, Cheriet et Moghaddam [83] ont suivi la même approche mais avec des caractéristiques basées sur le squelette de l'image. Ils ont extrait des caractéristiques topologiques et géométriques du squelette des CCs. La forme du script a été analysée pour extraire des descripteurs à l'aide des points terminaux et des points de branchement.

Saabni et Bronstein [84] ont proposé une méthode de repérage de mots pour traiter des documents de bonne qualité. Ils ont utilisé le contour du trait pour représenter une image de mot.

Sari et Kefali [51] ont extrait des caractéristiques structurelles d'un document segmenté en CCs. Le document a été codé par un fichier ASCII selon des propriétés structurelles telles que les boucles, les ascendants et les descendants. Ensuite, le repérage de mots est réalisé par recherche de texte.

Khayyat et al [12] ont proposé un système de repérage de mots basé sur l'apprentissage qui reconnaît les morceaux de mots arabes (PAW) avec les caractéristiques HOG en utilisant deux algorithmes de reconnaissance différents : les machines à vecteur support (SVM) et l'analyse discriminante régularisée (RDA). Leur système recherche les PAWs du mot requête et utilise les modèles de langage pour construire des mots correspondants au mot de la requête. Leur méthode nécessite une large base de données étiquetée pour l'apprentissage des paramètres du système.

2.3. Conclusion

Dans ce chapitre, nous avons présenté les méthodes d'extraction des caractéristiques utilisées pour le repérage de mots. Les caractéristiques proposées pour le repérage de mots dans les documents latins sont moins performantes avec les documents arabes. L'écriture arabe a sa forme particulière d'où la nécessité de proposer une nouvelle méthode pour l'extraction des caractéristiques pour le repérage de mots dans les documents arabes.

Etude préliminaire : repérage de mots en utilisant les caractéristiques des points-clés.

Nous avons fait une étude préliminaire pour vérifier l'efficacité de l'utilisation des points-clés pour l'extraction des caractéristiques pour l'appariement d'images de mots arabes (image matching). L'étude a démontré que l'extraction des caractéristiques à partir des points-clés est prometteuse pour le repérage de mots dans les documents arabes. Dans ce chapitre nous présentons la démarche entreprise en détails ainsi les résultats obtenus.

3.1. Méthode proposée

Nous avons proposé un système de repérage de mots de type QBE (*Query by Example*) permettant de repérer dans une collection d'images de document un mot de requête. La méthode proposée compare le mot requête avec les mots du document cible et retourne une liste ordonnée selon la similarité de la requête avec les mots du document. La méthode proposée est sans apprentissage et ne nécessite pas de base d'images étiquetées.

Nos contributions dans cette étude préliminaire se résument comme suit :

- Nous avons proposé un algorithme pour l'extraction de point-clés à partir d'une image de document. L'algorithme extrait les points terminaux et les points de branchement et effectue ensuite une sélection des points-clés importants.
- Nous avons proposé et extrait des caractéristiques de points-clés à partir de chaque point extrait de l'image.
- Nous avons proposé une fonction pour le calcul de similarité entre l'image requête et les mots du document cible.

- Nous avons proposé un algorithme pour la sélection des mots candidats, l'algorithme effectue une répartition des mots du document en méta-classes dans le but de réduire le nombre de mots à comparer avec la requête dans la phase recherche. Notre algorithme effectue une segmentation légère (soft clustering) en utilisant un modèle statistique de type mélange de gaussiennes GMM (Gaussian mixture model).

Notre système se compose de deux phases : la préparation de données et le repérage de mots. Le fonctionnement de notre système est illustré dans la figure 3.1.

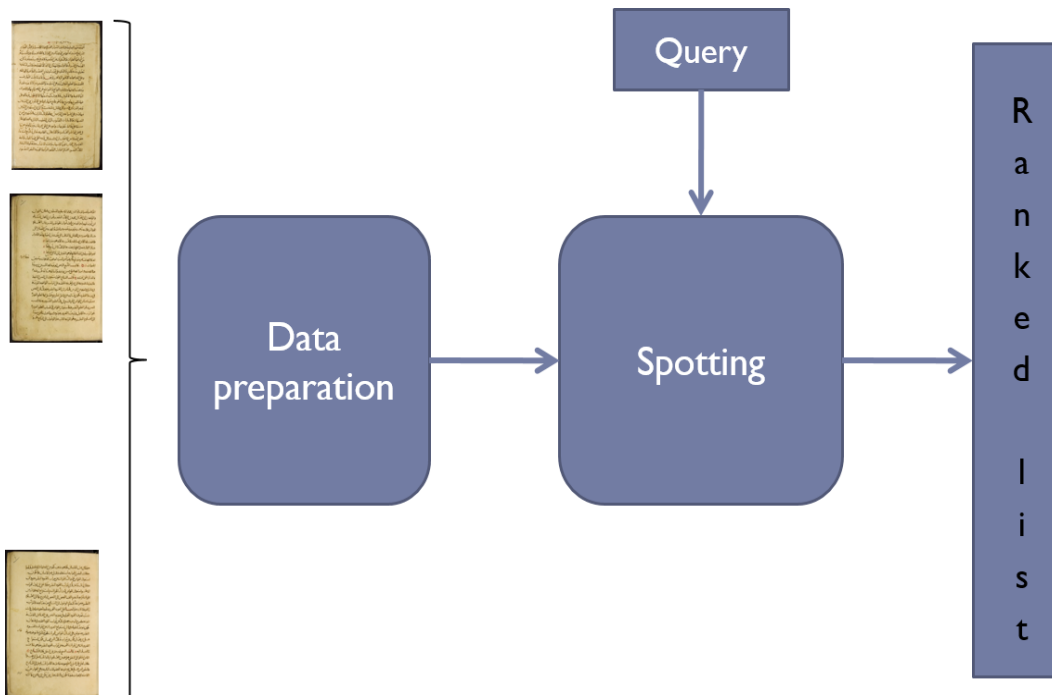


Figure 3.1. Diagramme général du système proposé (étude préliminaire).

3.2. Préparation des données

L'utilisation des images originales pour le repérage de mots est très difficile, toutes les méthodes effectuent une étape de prétraitement sur les images du document afin de supprimer les bruits et préparer les images pour les autres étapes. Avant de passer à la comparaison du mot requête avec les mots du document cible, les images sont binarisées et segmentées en sous-mots. Les méthodes proposées pour les langues latines effectuent une étape d'élagage, les images des mots de petites tailles sont écartés dans la phase recherche de mots. Les petits mots sont insignifiants et sont tout simplement ignorés. Dans la langue arabe tous les mots

sont à prendre en considération même les plus petits de taille. Donc, les techniques d'élagage ne sont pas applicables. De ce fait, nous avons introduit une étape de catégorisation des mots du document cible afin de comparer le mot requête avec seulement les mots qui lui ressemblent et par conséquent réduire le temps de recherche. La figure 3.2 illustre l'étape de préparation de données.

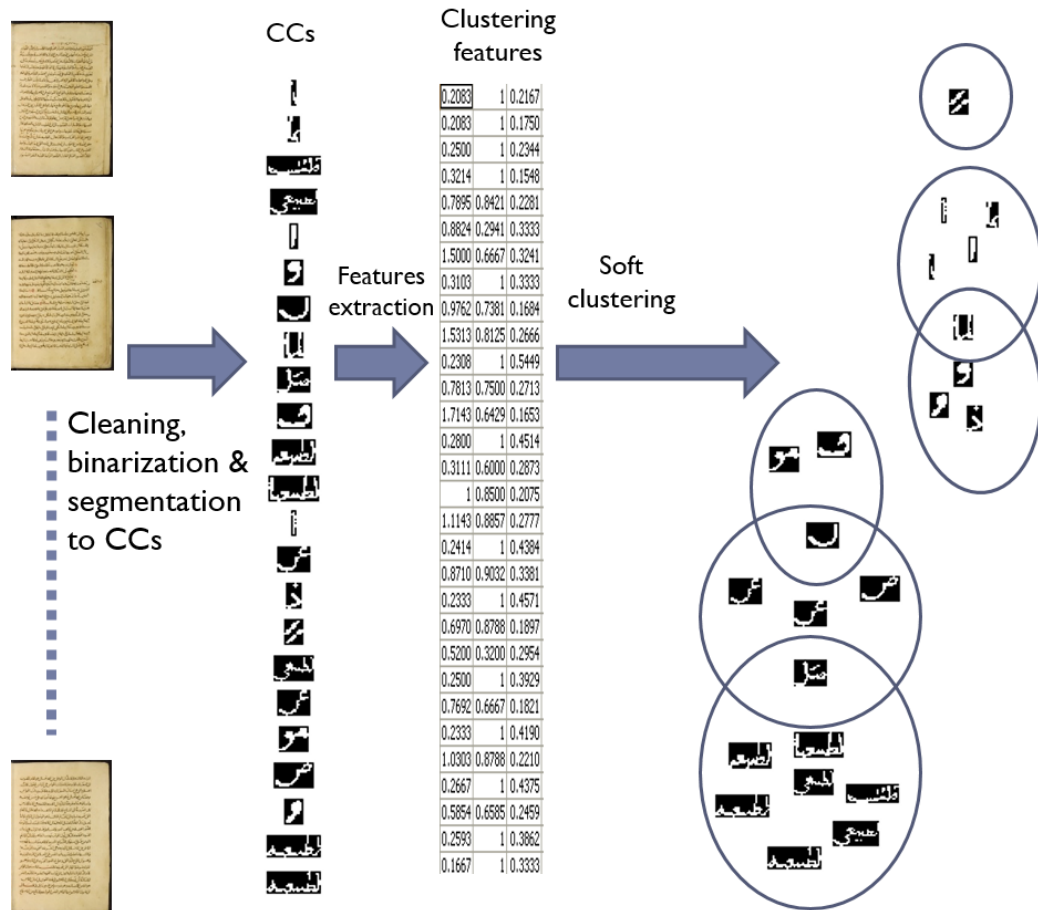


Figure 3.2. Préparation de données.

3.2.1. Binarisation et segmentation en sous-mots

La binarisation, la suppression du bruit ainsi que la segmentation en mots ne sont pas l'objet de la présente thèse. De ce fait, nous avons choisi une base d'images binarisées et segmentées en composantes connexes CCs (sous-mots). La base d'images choisie est IBN SINA [85], la base est créée à partir d'un ancien manuscrit arabe. Les images de la base sont déjà binarisées et segmentées en CCs et notre système effectue le repérage des CCs. La figure 3.3 présente quelques images tirées de la base IBN SINA.



Figure 3.3. Quelques images de la base IBN SINA.

3.2.2. Classification des sous-mots

Le but de la classification des CCs est de les répartir dans des méta-classes afin de réduire le nombre d'image à comparer avec le mot de requête. C'est la solution adoptée par certains auteurs pour réduire l'espace de recherche. Moghadam et al [29] ont utilisé un réseaux de neurone de type Self-Organizing Map pour créer des bibliothèques de composantes connexes et le mot de requête est recherché dans la bibliothèque au quelle il appartient. Mais les erreurs de la création des bibliothèques peuvent amplifier le taux d'erreur du système. Pour remédier à cet inconvénient nous avons opté pour une segmentation légère (soft clustering) car une séparation stricte des CCs n'est pas nécessaire. Nous avons fait en sorte qu'une CC peut appartenir à plusieurs classes. Nous avons utilisé six caractéristiques structurelles extraites de chaque CC avec un modèle de mélange de gaussiennes GMM pour la classification des CCs.

a) Extraction des caractéristiques

Un vecteur de six caractéristiques structurelles est extrait de chaque CC de la base d'images. Les caractéristiques sont comme suit :

- **Aspect ratio** : c'est le rapport entre la hauteur et la largeur de l'image H/W (figure 3.4).

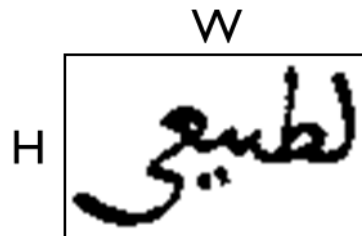


Figure 3.4. Calcul de du rapport hauteur/largeur

- **La densité des pixels noirs** : soit une image binaire de taille (H x W) définie comme ci-dessous (équation 3.1)

$$I(i,j) = \begin{cases} 1 & \text{if } I(i,j) \text{ is ink} \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

La densité des pixels d'encre dans l'image se calcule comme suit :

$$D = \left(\sum_{i=1}^H \sum_{j=1}^W I(i,j) \right) / (W \times H) \quad (3.2)$$

- **Le nombre de minimas dans la projection verticale** : c'est le nombre de pics inférieurs dans la projection verticale de l'image de la CC. Il indique approximativement le nombre de caractères du sous-mot.

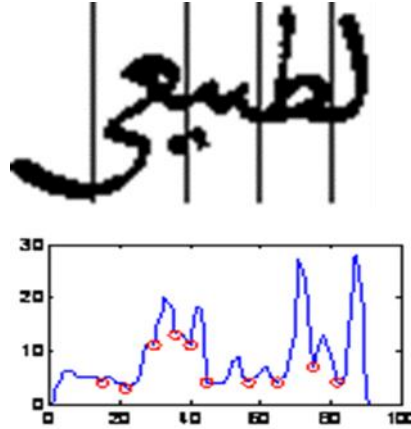


Figure 3.5. Calcul du nombre de minimas dans la projection verticale d'une image d'un sous-mot.

- **Le nombre de loupes** : une loupe est une partie de l'image encerclée par des pixels d'encre. Pour calculer le nombre de loupes, nous avons effectué un remplissage des loupes de l'image. Ensuite, l'image originale est soustraite de l'image remplie. Le nombre de composantes de l'image résultat est le nombre de loupes (figure 3.6).



Figure 3.6. Calcul du nombre de loupes.

- **Nombre d'ascendants et de descendants (hampes et jambes)** : pour calculer le nombre de jambes et le nombre de hampes, la partie médiane de l'image est supprimé (les pixels sont mis à zéro). Les composantes connexes restantes au-dessus de la partie médiane sont des hampes et celles au-dessous sont des jambes (figure 3.7).

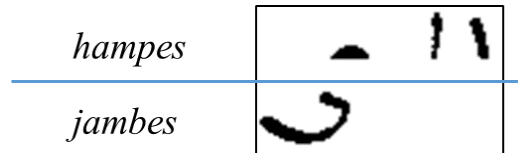


Figure 3.7. Calcul du nombre de jambes et de hampes

b) Regroupement des sous-mots (soft clustering)

Soit $C = \{CC_1 \dots CC_n\}$ l'ensemble des composantes connexes du document. L'objectif est de segmenter cet ensemble en K classes de manière souple de sorte qu'une composante connexe pourra appartenir à une ou plusieurs classes. Pour cela nous avons utilisé un modèle statistique de type mélange de gaussiennes avec K composantes. Les paramètres du modèle sont ajustés par l'algorithme de maximisation de vraisemblance [86]. La fonction de densité est donnée dans l'équation 3.3.

$$F(cc_i) = \sum_{j=1}^K w_j N(\mu_j, \sigma_j) \quad (3.3)$$

La composante $w_j N(\mu_j, \sigma_j)$ est la probabilité pour que la composante connexe CC_i appartienne à la classe j et un vecteur de probabilités de longueur K est associé à chaque composante CC_i . Par conséquent, une matrice de probabilités P de taille $(n \times k)$ est obtenue, P_{ij} est la probabilité pour que la composante CC_i appartienne à la classe j . Le tableau 3.1 montre une partie de la table de probabilités, certaines composantes ont des probabilités d'appartenance à différentes classes qui sont très proches. Dans ce cas, la composante doit être affectée à chacune de ces classes.

Tableau 3.1. Matrice de probabilité d'un modèle de mélange de gaussiennes à 8 classes.

CCs	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
01	2,18E-30	0,99999998	1,67E-08	6,57E-09	9,48E-15	1,43E-21	9,78E-19	4,81E-20
02	2,37E-14	1,90E-08	0,00656696	1,03E-07	0,99171169	9,67E-07	3,00E-07	0,00171996
03	0,99543657	4,02E-28	3,89E-10	1,50E-19	1,47E-21	1,99E-17	0,00448502	7,84E-05
05	4,58E-11	2,58E-11	0,99948631	0,00012502	1,09E-09	4,93E-08	0,00038834	2,82E-07
06	1,88E-06	5,96E-13	0,37004388	0,00074803	9,37E-05	0,00145622	0,26705906	0,36059723
07	0,99999972	4,30E-32	3,00E-14	1,19E-23	4,10E-26	3,35E-21	2,11E-07	6,46E-08
08	4,87E-06	1,42E-10	0,83144131	2,14E-05	4,28E-10	3,42E-08	0,16785145	0,00068096
09	2,76E-19	2,21E-08	0,00748404	0,48711008	5,62E-05	0,50534878	8,72E-07	3,11E-08

Pour effectuer une segmentation souple des CCs, nous avons proposé un algorithme qui utilise la matrice des probabilités pour classifier les CCs. Une composante connexe est supposée appartenir à un certain nombre de classes maximum M . Les classes d'une CC sont celles ayant leurs probabilités supérieurs à $1/(M+1)$. Si les probabilités de toutes les classes sont inférieures à $1/(M+1)$, les probabilités d'appartenance de la CC sont répartis sur un nombre de classes supérieur à M . Dans ce cas, la CC est classifiée dans les M classes ayants les meilleures probabilités. Le pseudo code de l'algorithme est donné dans Algorithme 1 ci-dessous.

Algorithm 1. CCs classification

Inputs: $p(n \times k)$: probability matrix.

M : number of cluster overlaps.

Outputs: classes of CCs.

- 1: **For** all CC_i **do**
 - 2: $Classes(CC_i) = \{C_j, P_{ij} \geq 1/(m+1)\};$
 - 3: **If** $classes(CC_i) = \emptyset$ **then**
 - 4: $Classes(CC_i) = \{the\ m\ best\ classes\};$
 - 5: **End if.**
 - 6: **End for.**
-

3.3. Le repérage de mots (word spotting)

Chaque mot de la base d'image est représenté par un ensemble de vecteurs de caractéristiques extraites à partir des points-clés. Dans la phase recherche de mots, une image est sélectionnée de la base pour être utilisée comme une requête.

Une fonction de distance est utilisée pour calculer la similarité de la requête avec chacune des CCs appartenant à la même classe que la requête. Le système renvoie en résultat une liste ordonnée (ranked list) des CCs de la base selon leurs similarités avec la requête (figure 3.8).

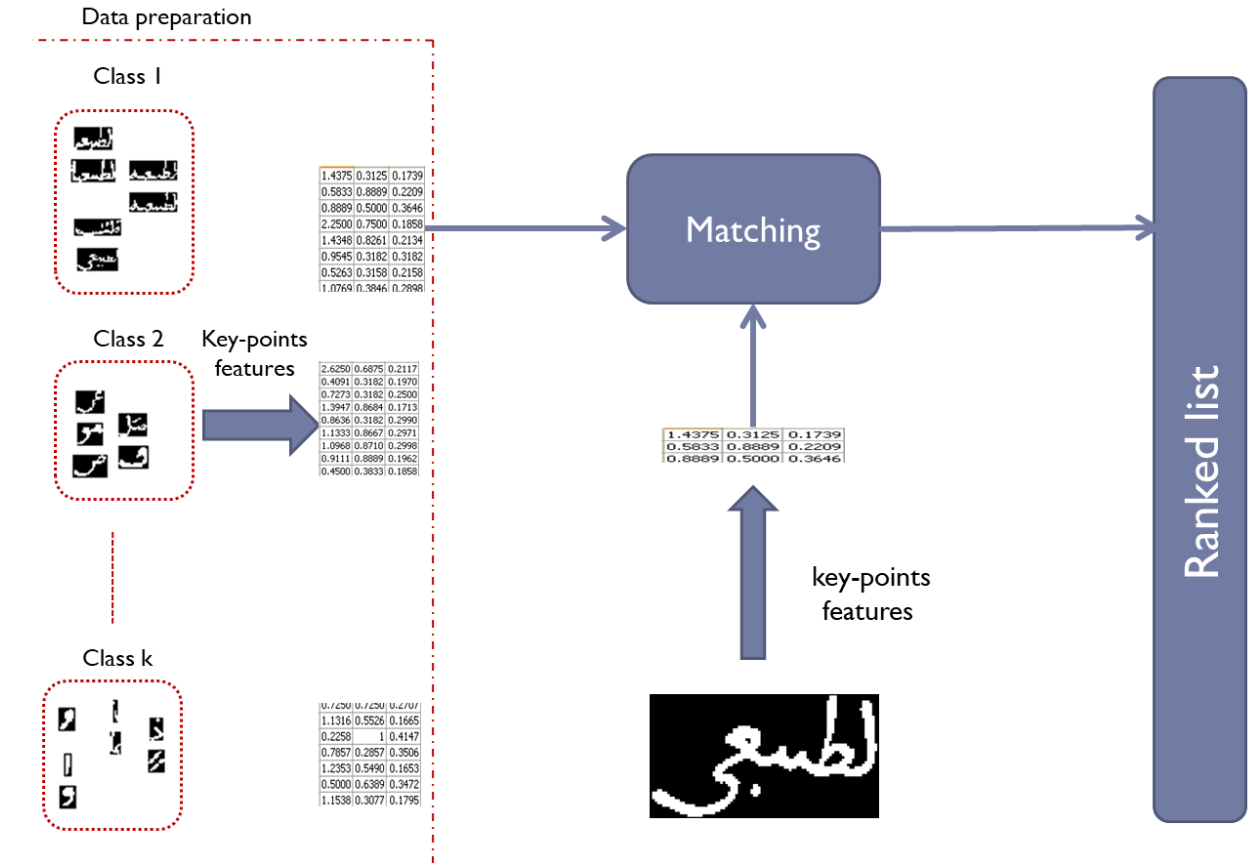


Figure 3.8. Le repérage de mots.

3.3.1. Extraction des points-clés

Avant de présenter le processus d'extraction des points-clés, nous définissons les notions de points terminaux et points de branchement :

- Point terminal : c'est un pixel du squelette de l'image ayant un seul voisin.
- Point de branchement : les points de branchement sont les points d'intersection des traits du squelette de l'image. Un point de branchement est un pixel ayant deux voisins est plus dans le squelette de l'image.

Dans ce mémoire, le mot point-clé désigne indifféremment un point terminal ou un point de branchement. Des exemples de points-clés sont montrés dans la figure 3.9.

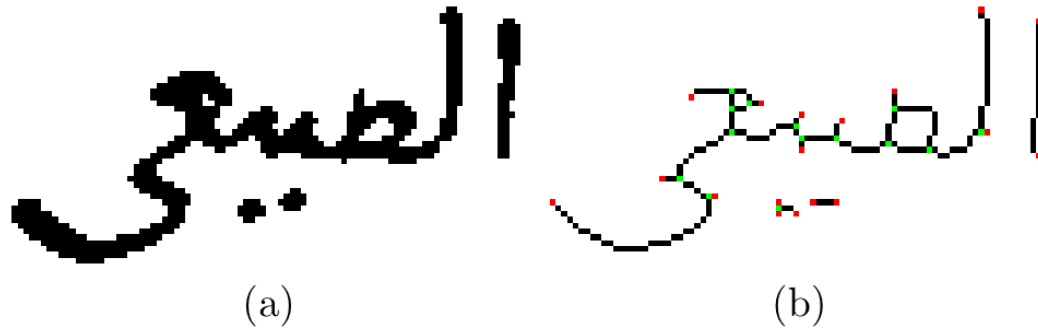


Figure 3.9. (a) image binarisée d'un mot arabe. (b) les points terminaux (rouge) et les points de branchements (vert) dans le squelette de l'image.

Tous les points terminaux et les points de branchement sont extraits de l'image et une opération de nettoyage est appliquée pour supprimer les bruits. Les points-clés sont nettoyés en utilisant la distance de squelette D_s et un paramètre estimé à partir des images cible à savoir l'épaisseur moyenne de l'écriture ASW (*average stroke width*). La distance de squelette $D_s(KP_i, KP_j)$ entre deux points-clés KP_i et KP_j est le nombre de pixels du squelette qui connecte les deux points-clés. Nous supposons que deux points-clés doivent avoir une distance supérieure à ASW .

a) *Estimation de la largeur des traits(ASW)*

Pour chaque image de la base, la projection verticale est calculée. Etant donnée une image I de taille $(h \times w)$, sa projection verticale P est un vecteur de taille w calculé comme suit (équation 3.4) :

$$P(j) = \sum_{i=1}^h I(i, j) \quad (3.4)$$

Les minimas locaux sont déterminés dans la projection verticale de chaque image de la base. La figure 3.10 présente un exemple de la projection vertical d'une image et les minimas locaux correspondants. La valeur moyenne des minimas locaux est la largeur moyenne des traits de l'écriture ASW .

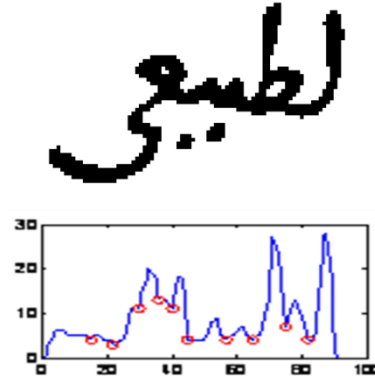


Figure 3.10. Minimas locaux dans la projection verticale d'une image.

b) *Nettoyage des points terminaux*

Soit $EP_{raw} = \{EP_i, i=1 \dots N_{ep}\}$ et $BP_{raw} = \{BP_i, i = 1 \dots N_{bp}\}$ l'ensembles des points terminaux et des points de branchements bruts respectivement, avec N_{ep} et N_{bp} sont les nombres des points terminaux et points de branchement. La distance de squelette D_s est calculée entre chaque point terminal EP_i et tous les points-clés connectés à EP_i par des pixels du squelette que ceux soit des EP ou des BP . Les points terminaux ayant une distance D_s inférieure à ASW sont considérés inutiles et donc supprimés. L'ensemble des points terminaux après nettoyage est défini par l'équation (3.5)

$$EP_{clean} = \{EP_i \in EP_{raw} / ds(EP_i, KP_j) > ASW, \forall KP_j \in \{EP_{raw} \cup BP_{raw}\}\} \quad (3.5)$$

c) *Nettoyage des points de branchement*

Dans le processus de nettoyage des points de branchement, l'ensemble des points terminaux nettoyés EP_{clean} sont considérés comme des points de branchement. L'ensemble EP_{clean} est incluse dans l'ensemble des points de branchement bruts $BP_{raw} = BP_{raw} \cup EP_{clean}$. Un poids w_i est calculé pour chaque points-clés KP_i de l'ensemble BP_{raw} . Le poids w_i est le cumul des distances D_s entre le point-clés KP_i et tous ses voisins. Les voisins du points-clés KP_i sont les autres points-clés qui lui sont connectés avec des pixels du squelette. Le poids w_i est calculé avec l'équation (3.6). Si deux points de branchement ont une distance inférieure à ASW , le point ayant un poids minimum est supprimé. L'ensemble des points de branchement après nettoyage est défini par l'équation (3.7)

$$w_i = \sum_j d_s(BP_i, KP_j), \forall KP_j \in Neighbors(BP_i) \quad (3.6)$$

$$BP_{clean} = \{BP_i \in BP_{raw} / ds(BP_i, KP_j) < ASW, \forall KP_j \in \{BP_{raw} \cup EP_{clean}\}\} \quad (3.7)$$

Avec $Neighbors(BP_i)$ est une fonction qui retourne l'ensemble des points-clés connectés au point-clés BP_i dans l'image de squelette. Le nettoyage des points-clés est effectué avec l'algorithme 2 dans 3 étapes : nettoyage des points terminaux, calcul des poids des points de branchement et finalement nettoyage des points de branchement. Dans l'algorithme 2, la fonction $Neighbors(EP_i)$ renvoie un seul point-clés lorsqu'elle est utilisée avec un point terminal en paramètre. Parce qu'un point terminal a un seul voisin dans l'image de squelette. Lorsque la fonction $Neighbors(BP_i)$ est utilisée avec un point de branchement en paramètre, elle renvoie l'ensemble de points-clés connecté au point-clés BP_i (trois ou plus). Un exemple des points-clés extrait avec l'algorithme 2 est affiché dans la figure 3.11.

Algorithm 2. Keypoints cleaning

Input: EP_{raw}, BP_{raw} : The raw sets of Endpoints and Branch points respectively

Output: EP_{clean}, BP_{clean} : The clean sets of Endpoints and Branch points respectively

```

1:  $EP_{clean} \leftarrow \emptyset;$ 
2: for all  $EP_i$  in  $EP_{raw}$  do
3:   if  $d(EP_i, Neighbors(EP_i)) > ASW$  then
4:      $EP_{clean} \leftarrow EP_{clean} \cup \{EP_i\}$ 
5:   end if
6: end for
7: for all  $BP_i$  in  $BP_{raw}$  do
8:    $Nbrs \leftarrow Neighbors(BP_i)$ 
9:    $W(i) \leftarrow 0$ 
10:  for all  $Bbr_j$  in  $Nbrs$  do
11:     $W(i) \leftarrow W(i) + d(BP_i, Nbr_j)$ 
12:  end for
13: end for
14:  $BP_{clean} \leftarrow BP_{raw}$ 
15: for all  $BP_i$  in  $BP_{clean}$  do
16:    $Nbrs \leftarrow Neighbors(BP_i)$ 
17:   for all  $Bbr_j$  in  $Nbrs$  do
18:     if  $d(BP_i, Nbr_j) < ASW$  then
19:        $BP_{clean} \leftarrow BP_{clean} - \{ \operatorname{argmin}(W(BP_i), W(Nbr_j)) \}$ 
20:     end if
21:   end for
22: end for

```

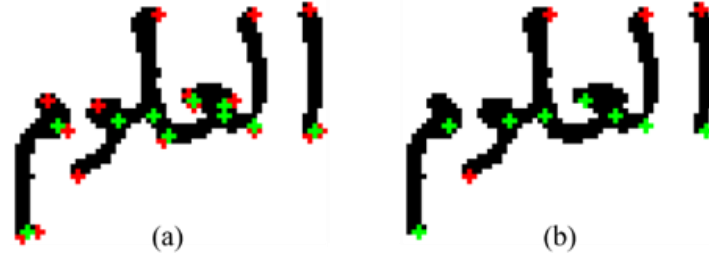


Figure 3.11. Un exemple d'extraction de points clés : (a) points terminaux et points de branchement bruts. (b) Points clés obtenus après nettoyage par l'algorithme 2.

3.3.2. Extraction des caractéristiques des points-clés

Après l'extraction des points-clés, un vecteur de caractéristiques est extrait à chaque point-clé de l'image. Les caractéristiques utilisées décrivent la position du point-clé dans l'image et la répartition spatiale de l'ancre dans son voisinage. Soit $P_i(x_i, y_i)$ un point-clé extrait d'une image ayant $C(x_c, y_c)$ comme centre. Les primitives extraites du point-clé P_i se définissent comme suit :

- La distance entre P_i et le centre de l'image : elle se calcule comme étant l'amplitude du vecteur $\overrightarrow{CP_i}$ qui se calcule comme suit (équation 3.8) :

$$\|\overrightarrow{CP_i}\| = \sqrt{(x_i - c_x)^2 + (y_i - c_y)^2} \quad (3.8)$$

- L'angle du vecteur $\overrightarrow{CP_i}$ avec l'axe horizontal calculé dans l'équation (3.9)

$$\theta = \tan^{-1}((x_i - c_x)/(y_i - c_y)) \quad (3.9)$$

- La densité de l'ancre dans le voisinage du point-clé : c'est le rapport entre le nombre de pixel d'encre et le nombre de pixels de l'arrière-plan dans un rayon défini par un cercle centré sur le point-clé. Nous avons calculé cette densité à 3 échelles, nous avons utilisé un cercle de rayon variant de $r=ASW$ jusqu'à $r=3xASW$ tel que illustré dans la figure 3.12. La densité de l'ancre est calculée par l'équation (3.10)

$$\text{Ink_density}_{s=1..3} = \frac{\sum I(x,y)}{\sum 1-I(x,y)} \cdot \sqrt{(x-x_c)^2 + (y-y_c)^2} \leq r^2, r = SxASW \quad (3.10)$$

Finalement, un vecteur de 5 caractéristiques est extrait de chaque point-clé de l'image.

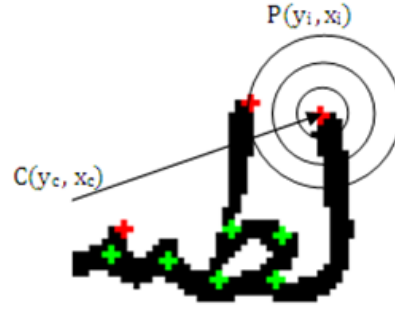


Figure 3.12. Extraction des caractéristiques de point-clés.

3.3.3. Mesure de dissimilarité (image matching)

Le repérage de mots se fait avec les composantes connexes. Lorsque le système reçoit une CC en requête, il calcule sa dissimilarité avec toutes les CCs appartenant à sa classe. Soit deux composantes connexes CC^a et CC^b décrites par les deux ensembles de points-clés $KP^a = \{kp_1^a \dots kp_{L_a}^a\}$ et $KP^b = \{kp_1^b \dots kp_{L_b}^b\}$ respectivement. Avec L_a et L_b le nombre de points-clés extraits à partir des composantes CC^a et CC^b . La dissimilarité entre deux composantes connexes est calculée en utilisant la distance Euclidienne entre les vecteurs de caractéristiques extraits à partir de leurs points-clés. Soit kp_i^a et kp_j^b deux points-clés appartenant aux composantes CC^a et CC^b , soit F_i^a et F_j^b les deux vecteurs de primitives correspondants. La distance euclidienne est définie par l'équation (3.11)

$$d_{euc}(F_i^a, F_j^b) = \sqrt{\sum_{k=1}^5 (F_i^a(k) - F_j^b(k))^2} \quad (3.11)$$

Pour chaque point-clé kp_i^a de la composante CC^a , nous déterminons son *match* dans la composante CC^b , $match(kp_i^a)$ est un point-clé de la composante CC^b ayant la distance minimale avec kp_i^a . Le *match* d'un point-clé est défini dans l'équation (3.12)

$$match(kp_i^a) = \underset{j}{\operatorname{argmin}}(d_{euc}(kp_i^a, kp_j^b)) \quad (3.12)$$

Finalement, la dissimilarité entre les composantes CC^a et CC^b est la somme des distances entre les points-clés de la composante CC^a et leurs *matches* correspondants. La figure 3.13 montre un exemple des *matches* des points-clés d'une CC. La dissimilarité $d(CC^a, CC^b)$ est définie ainsi par l'équation (3.13)

$$D(CC^a, CC^b) = \sum_{i=1}^{L_a} d_{euc}(kp_i^a, match(kp_i^a)) \quad (3.13)$$



Figure 3.13. Un exemple des matchs des points-clés d'une CC dans une autre CC.

3.4. Expérimentation et résultats

Les étapes de prétraitement des images ne font pas partie des objectifs de la présente thèse. Notre évaluation se focalise sur la phase repérage de mots, de ce fait nous avons choisi une base d'images binarisées est segmentées en CCs. Des détails concernant la base d'image, les métriques utilisées pour évaluer notre méthode ainsi les résultats obtenus sont donnés dans la suite de cette section.

3.4.1. La base d'image

La méthode proposée est évalué avec la base d'image IBN SINA [19]. La base est construite à partir d'un ancien manuscrit arabe contenant 50 feuilles. Les images de la base IBN SINA sont binarisée est nettoyées des bruits liés aux dégradations. Les images des feuilles sont segmentées en composantes connexes. Dans cette étude nous avons utilisé les dix premières pages pour l'évaluation, ces pages contiennent environ 5684 composantes connexes. Nous avons sélectionné 9 images pour les utiliser comme des requêtes. Le nombre de caractères des requêtes varie de 4 jusqu'à 7.

3.4.2. Métriques d'évaluation

La méthode proposée est évaluée à deux niveaux à savoir : la classification des CCs dans des méta-classes et le repérage de mots. Pour la création des méta-classes nous avons utilisé la précision pour évaluer notre algorithme. Nous avons classifié les CCs de manière souple, une CC peut être classifiée dans plusieurs classes. Pour calculer la précision de notre algorithme pour une CC_i donnée, nous avons considéré la classe ayant un maximum d'occurrences de le CC_i . La précision de la méthode pour la composante CC_i est le taux de ses occurrences qui se trouvent dans la même classe que CC_i . La précision de la classification d'une composante CC_i est définie par l'équation (3.14).

$$\text{Precision}(\text{CC}_i) = \frac{\text{Max}_{k=1}^m (|\text{class}(\text{CC}_j) = k|)}{|\text{CC}_j|}, j = 1..n \quad (3.14)$$

Avec, m est le nombre de méta-classes et n est le nombre d'occurrence de la composante CC_i .

Pour l'évaluation de notre méthode dans le repérage de mots, nous avons utilisé le taux de *rappel* et la *précision*, ce sont les métriques les plus utilisées pour l'évaluation des systèmes de recherche d'information dans la littérature [87].

Le *rappel* mesure la capacité du système à retrouver toutes les occurrences pertinentes d'une requête. La *précision* mesure le taux de résultats pertinents retrouvé par le système. Les deux métriques sont calculées dans l'équation (3.15).

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.15)$$

Avec TP est le nombre de vrais positifs, FP est le nombre de faux positifs et FN et le nombre de faux négatifs.

La *précision* et le *rappel* sont calculés pour chacune des requêtes utilisées dans l'évaluation, ils reflètent l'efficacité du système dans la recherche d'une requête particulière. L'évaluation doit tenir compte de la qualité du système pour toutes les requêtes, pour cela la précision moyenne AP (Average Precision) et la moyenne des précisions moyennes MAP (Mean Average Precision) sont utilisées. Soit $Q = \{q_j, j = 1..n\}$ l'ensemble des requêtes utilisées pour l'évaluation, à chaque fois qu'une occurrence d'une requête q_j est récupérée, la précision est calculée. La valeur moyenne de ces précisions est la précision moyenne AP qui est calculée par l'équation (3.16). Dans l'équation (3.16) m_j est le nombre d'occurrences de la requête q_j . La précision MAP est la moyenne des précisions AP, elle est calculée par l'équation (3.17).

$$\text{AP}(q_j) = \frac{1}{m_j} \sum_{k=1}^{m_j} \text{precision}(R_{jk}) \quad (3.16)$$

$$\text{MAP}(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \text{AP}(q_j) \quad (3.17)$$

3.4.3. Résultat et discussions

Pour la classification des CCs nous avons considéré un nombre de classes $k=8$, ce choix est fait en raison du nombre de caractère que contient les CCs de la base d'image. La longueur des CCs en nombre de caractères varie de 1 à 8. Le nombre de classes maximum attribuées à une CC est $m=3$ qui est déterminé de manière empirique. La précision pour chaque CC est calculée par l'équation (3.14) et la valeur moyenne de la précision est calculée pour chaque catégorie de CC selon leurs nombres de caractères. La précision moyenne selon le nombre de caractères des CCs est affichée dans la figure 3.14. La classification souple ($m=3$) améliore la précision de classification des CCs par rapport à la classification dure ($m=1$).

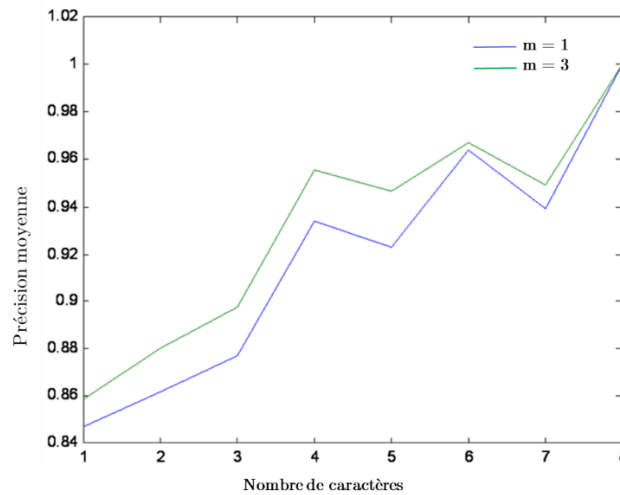


Figure 3.14. La précision de la classification des CCs.

Nous avons comparé notre méthode de repérage de mots avec les caractéristiques des profils de mots qui sont très populaire dans la littérature. Les profils des mots sont des séquences de tailles variables, de ce fait nous avons utilisé l'algorithme DTW pour le calcul de dissimilarité tel est le cas de plusieurs méthodes [56, 62].

Le tableau 3.2 présente les résultats obtenus pour le repérage de mots par notre méthode et le tableau 3.3 présente les résultats obtenus par les profils de mots avec l'algorithme DTW. La méthode proposée est plus performante que la méthode DTW en terme de précision et ainsi en terme de temps de réponse. Une valeur de **77.63%** pour la précision MAP avec un temps de réponse moyen de **5.94s** sont obtenus pour notre méthode. Pour la méthode DTW, un MAP de 60% avec un temps de réponse de **15.78%** sont obtenus.

Tableau 3.2. Résultats de la méthode proposée.

Requêtes	Nombre d'instances	Méthode proposée			
		Top 10		AP (%)	Temps de réponse en seconds
		Précision (%)	Rappel (%)		
لنكر	22	70	31.8182	51.6452	5.9204
لعلو	13	60	46.1538	52.1361	4.5012
لسبج	12	90	75.0000	95.9366	5.6067
لغلط	8	60	75.0000	83.6310	7.1940
لظز	27	100	37.0370	92.6246	6.2512
لظلو	5	30	60.0000	68.3333	6.1887
لطسعي	11	90	81.8182	91.7063	4.7317
لنظنه	6	50	83.3333	89.5833	4.9860
لطصه	5	40	80.0000	73.0952	7.7338
Moyenne				77.63	5.9492

Tableau 3.3. Résultats des profils de mots avec DTW.

Requêtes	Nombre d'instances	Profils de mots avec DTW			
		Top 10		AP (%)	Temps de réponse en seconds
		Précision (%)	Rappel (%)		
لنكر	22	80	36.3636	44.8626	12.3687
لعلو	13	90	69.2308	79.5586	5.0611
لسبج	12	40	33.3333	45.4300	9.6093
لغلط	8	50	62.5000	66.2957	23.0787
لظز	27	90	33.3333	70.8364	22.1770
لظلو	5	30	60.0000	66.8421	14.5825
لطسعي	11	90	81.8182	95.3604	7.3350
لنظنه	6	40	66.6667	48.6652	7.3791
لطصه	5	10	20.0000	22.8383	40.4709
Moyenne				0.6008	15.7847

3.5. Conclusion

L'étude préliminaire a démontré l'efficacité de l'utilisation des points-clés à savoir les points terminaux et les points de branchement pour l'extraction des caractéristiques de l'écriture arabe manuscrite. Les caractéristiques des points-clés sont prometteuses dans la comparaison des images de documents arabes et impose ainsi une étude approfondie.

Spatial Distribution of Ink at Keypoints (SDIK) : Un nouveau descripteur pour le repérage de mots dans les documents arabes.

Dans ce chapitre, nous présentons une nouvelle méthode d'extraction de caractéristiques adaptée à la nature complexe de l'écriture arabe. Nous avons appelé la caractéristique proposée Spatial Distribution of Ink at Keypoints (SDIK). Une étude préliminaire est présentée dans le chapitre 3 sur l'utilisation des points-clés pour l'extraction des caractéristiques. Dans cette première étude, nous avons constaté que les points terminaux et les points de branchement sont des zones prometteuses pour l'extraction des caractéristiques de l'écriture arabe. Le descripteur SDIK vise à capturer la répartition spatiale des pixels d'encre dans l'image de texte dans le voisinage des points-clés. Avec leurs représentations de longueurs fixes, nos descripteurs sont très rapides à comparer. Le descripteur SDIK permet la comparaison d'un mot avec une ligne de texte entière et évite la segmentation d'un document arabe en mots.

4.1. Contributions

Les principales contributions de l'étude présentée dans ce chapitre sont résumées comme suit :

- Nous avons proposé un nouveau descripteur appelé SDIK, qui capture la répartition spatiale de l'encre au voisinage d'un point-clé. Les descripteurs SDIKs sont de longueurs fixes et ils sont est très rapide à comparer.
- Un algorithme d'appariement (matching) est proposé, il permet de calculer la similarité d'un mot avec une ligne entière et donc, nous évitons de segmenter le document cible en mots.

- Les expérimentations effectuées sur deux bases d'images ont montré l'efficacité du descripteur SDIK pour le repérage des mots dans les documents modernes et historiques.

4.2. Méthode proposée

La méthode proposée est conçue pour le repérage de mots dans les documents arabes. Elle suit les méthodes QBE (Query by Example) et ne nécessite pas de données étiquetées. L'écriture arabe est cursive, les caractères d'un mot sont connectés par des lignes horizontales. Ces connexions sont parfois prolongées excessivement et prolonge l'écriture horizontalement. La figure 4.1 montre un exemple de cette caractéristique dans un mot arabe écrit de différentes manières. Ainsi, les principales caractéristiques discriminantes de l'écriture arabe sont concentrées aux points de branchement et aux extrémités des traits. Nous avons remarqué que pour comparer deux images de texte arabe, il n'est pas nécessaire de comparer la totalité des traits.

Un nouveau descripteur appelé SDIK (Spatial Distribution of Ink at Keypoints) est proposé. Le descripteur proposé utilise le voisinage des points de branchement et des points terminaux pour représenter l'écriture manuscrite arabe. Il est à noter que les points-clés ne sont pas considérés comme des caractéristiques (primitives) mais comme des zones adéquates pour extraire les descripteurs SDIKs. Le descripteur SDIK capture la répartition spatiale des pixels d'encre dans le voisinage d'un point-clé. Le descripteur SDIK est un histogramme qui quantifie les pixels d'encre qui remplissent des régions prédéfinies au voisinage d'un point-clé. Comme la caractéristique SDIK est un histogramme, le temps de calcul pour comparer deux vecteurs de caractéristiques est très faible. Cet avantage nous a permis de calculer la dissimilarité entre un mot de requête et des lignes de texte entières. Par conséquent, le document cible est segmenté en lignes et nous avons évité la segmentation en mots qui est connue comme étant une tâche très complexe. La méthode proposée suit principalement trois étapes, comme le montre la figure 4.2. Premièrement, les points-clés sont extraits de la requête et des lignes du document cible. Deuxièmement, les caractéristiques SDIK sont extraites des lignes et du mot de requête. Enfin, un processus de d'appariement ordonne les lignes cibles en fonction de leur similarité avec la requête.

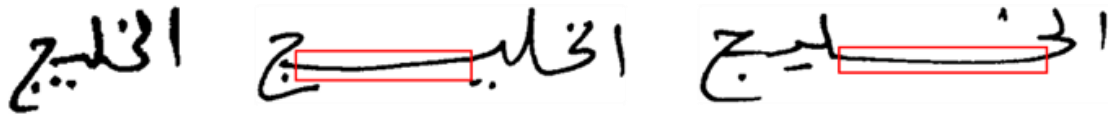


Figure 4.1. Trois instances d'un mot arabe écrit de différentes manières. Les rectangles montrent des parties non discriminantes de l'écriture.

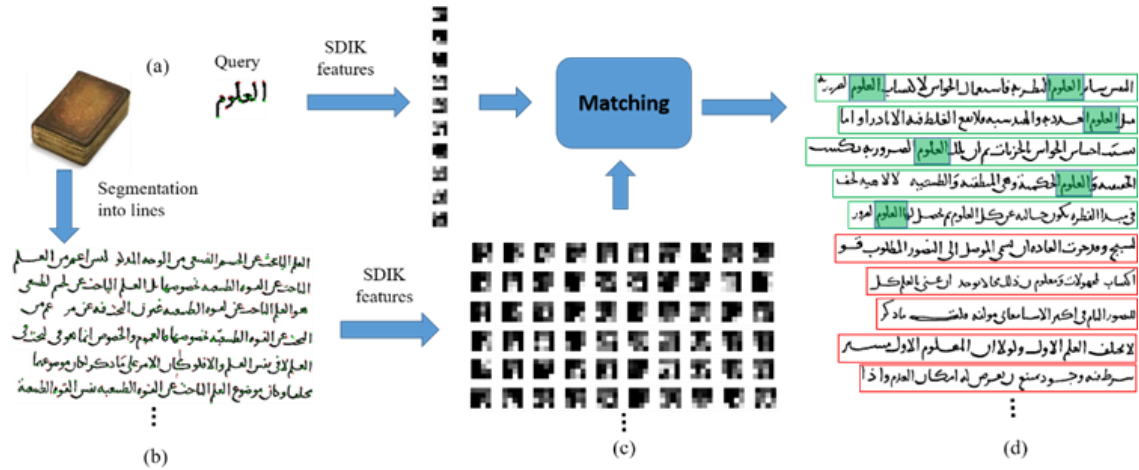


Figure 4.2. Diagramme de la méthode proposé : (a) un document cible et un mot de requête. (b) Le document cible segmenté en lignes. (c) Caractéristiques SDIK extraites de chaque point-clé dans les lignes du document et le mot de requête. (d) Lignes de texte ordonnées en fonction de leur similarité avec le mot de requête.

4.3. Extraction des caractéristiques SDIK

Après avoir effectué le processus d'extraction des points-clés, le vecteur de caractéristiques SDIK est extrait à chaque point-clé. Nous avons utilisé l'algorithme proposé dans l'étude préliminaire (chapitre 3) pour l'extraction et le nettoyage des points-clés. Le descripteur SDIK a pour objectif de capturer la répartition spatiale de l'encre dans le voisinage d'un point-clé. Les points-clés ne sont pas utilisés comme des caractéristiques mais comme des emplacements appropriés pour extraire les descripteurs SDIKs. Soit un ensemble de n points clés $P = \{KP_1, \dots, KP_n\}$ extraits d'une image I binarisée. L'image binaire I est définie par l'équation (4.1).

$$I(x, y) = \begin{cases} 1, & \text{if } I(x, y) \text{ is ink pixel} \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

Nous représentons chaque point clé KP_i par un vecteur SDIK qui capture la répartition spatiale des pixels d'encre autour du point-clé KP_i . Pour ce faire, des zones directionnelles et circulaires sont considérées pour partitionner le voisinage du point-clé. Pour les zones directionnelles, le voisinage du point clé est découpé

de manière régulière en plusieurs directions (figure 4.3). Pour les régions circulaires, un cercle de rayon r est centré au point-clé KP_i pour circonscrire chaque zone. Le rayon des zones circulaires varie de r jusqu'à $r \times c$ avec c est le nombre de zones circulaires considérées. La figure 4.3 montre différentes configurations de zones directionnelles avec trois zones circulaires.

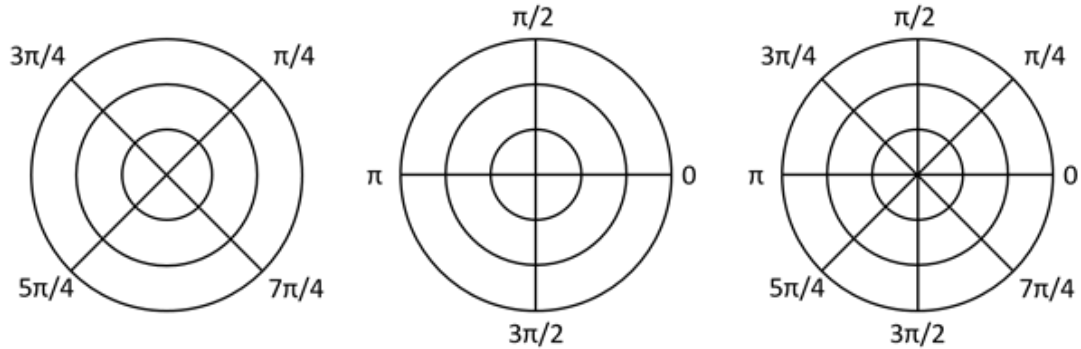


Figure 4.3. Les zones directionnelles et circulaires utilisées pour partitionner le voisinage d'un point-clé.

Finalement, le descripteur SDIK extrait d'un point clé KP_i est un histogramme h_i défini par l'équation (4.2). L'histogramme h_i est un vecteur de longueur $K=d \times c$, avec d est le nombre de zones directionnelles et c est le nombre de zones circulaires. Ainsi, $h_i(k)$ est le nombre de pixels d'encre dans la $k^{ième}$ zone du voisinage du point-clé.

$$h_i(k) = |\{I(y, x) = 1, \quad I(y, x) \in bin(k)\}| \quad (4.2)$$

La figure 4.4 montre un exemple de calcul des caractéristiques SDIK. Dans cet exemple, nous avons considéré trois zones circulaires et quatre directions. Un vecteur de 12 bins est obtenu (3 zones circulaires x 4 zones directionnelles). Pour simplifier, nous avons étiqueté les zones directionnelles par gauche (L), droite (R), haut (U) et bas (D), et les zones circulaires par C1, C2 et C3 comme le montre la figure 4.4(a). Le descripteur SDIK extrait du point clé mis en évidence sur la figure 4.4(c) est affiché en niveaux de gris dans la figure 4.4(b), plus un bin est sombre plus il a une grande valeur.

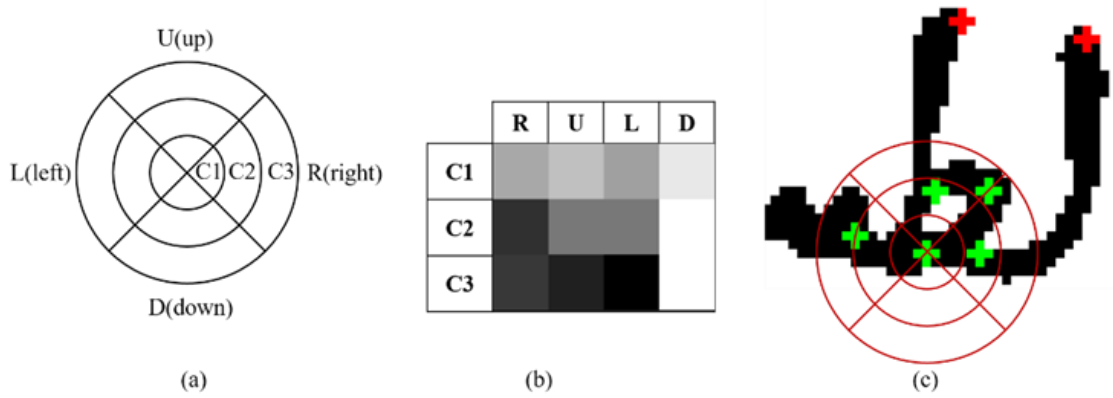


Figure 4.4. Calcul des descripteurs SDIK : (a) Le modèle de zones directionnelles et circulaires utilisé. (b) Descripteur SDIK affiché en niveaux de gris (sombre = valeur élevée). (c) Une partie d'une image avec les points-clés extraits.

La figure 4.5 montre un autre exemple d'extraction des caractéristiques SDIK. Dans cet exemple, nous avons utilisé la même configuration des zones circulaires et directionnelles présenté dans la figure 4.4(a). Nous avons affiché quelques vecteurs SDIK extraits des points-clés sélectionnés dans la ligne de texte. Il est à noter la similarité entre les vecteurs SDIK extraits des points clés similaires (points-clés numéro 1 et 5, 2 et 6, 3 et 7, 4 et 8).

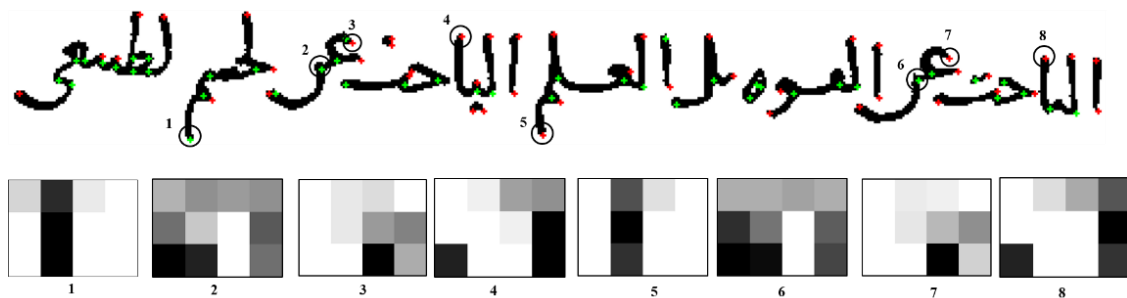


Figure 4.5. Exemples de vecteurs SDIK extraits à partir des points-clés d'une ligne de texte.

4.4. Calcul de similarité (Image Matching)

Étant donné une image de requête q , notre objectif est d'ordonner les lignes de texte dans le document cible en fonction de leur similarité avec l'image de requête. Considérons $KP^q = \{KP_1^q \dots KP_n^q\}$ et $KP^l = \{KP_1^l \dots KP_m^l\}$ les ensembles de points-clés extraits respectivement d'une requête q et d'une ligne de texte cible l . Soit C_q la dissimilarité de la requête q avec la ligne l . Pour calculer ce coût, nous devons chercher un sous-ensemble de points-clés de la ligne qui minimise la dissimilarité avec la requête. Les points-clés extraits de la requête et la ligne cible sont triés selon leur coordonnée x dans l'image. Ensuite, nous avons calculé une

matrice de dissimilarité D de taille $(n \times m)$ entre chacun des deux points clés de la requête q et la ligne cible l . Les points-clés de la requête correspondent aux lignes et les points-clés de la ligne cible correspondent aux colonnes. Alors, D_{ij} est la dissimilarité de deux points clés KP_i^q et KP_j^l de la requête et de la ligne cible respectivement. Étant donné que les descripteurs SDIK sont des vecteurs de longueurs fixes, il est naturel d'utiliser le test X^2 pour calculer cette distance. La dissimilarité $D_{ij}(KP_i^q, KP_j^l)$ est calculée par l'équation (4.3). Où h_i et h_j sont les vecteurs de caractéristiques SDIK de K bins extraits respectivement des points-clés KP_i^q et KP_j^l .

$$D_{ij}(KP_i^q, KP_j^l) = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \quad (4.3)$$

Le but est de trouver une région de la ligne l qui minimise le coût C_{ql} . Pour cela, nous avons proposé un algorithme pour chercher d'une séquence de points-clés de la ligne l qui minimise ce coût (Algorithme 3). Une fois la matrice de dissimilarité calculée (Algorithme 3 : ligne 3), l'algorithme parcourt la matrice D sur l'index des colonnes. À partir de chaque colonne sur la matrice D , l'algorithme sélectionne une partie de n colonnes de j à $j+n$. Ensuite, il détermine dans cette partie le *match* de chaque point-clé KP_i^q de la requête q (Algorithme 3 : ligne 8). $match(KP_i^q)$ est le meilleur point-clé correspondant à KP_i^q . Une dissimilarité S_d est calculée pour chaque partie sélectionnée comme la somme des dissimilarités de chaque point clé de la requête et de son *match* dans cette partie (Algorithme 3 : ligne 10). Enfin, l'algorithme renvoie la valeur minimale des distances S_d comme étant le coût C_{ql} .

Enfin, les lignes de texte du document cible sont ordonnées selon leurs dissimilarités C_{ql} avec l'image de la requête. La figure 4.6 fournit un exemple du processus de calcul de dissimilarité. La figure 4.6 (a) montre un exemple d'image de requête et les *matches* de ses points-clés sélectionnés par l'algorithme 3 dans la ligne cible. La figure 4.6 (b) met en évidence la matrice de dissimilarité, la partie sélectionnée par le rectangle correspond aux *matches* des points-clés de la requête.

Algorithm 3. Keypoints matching.

- 1: **Input:** $KP^q = \{KP_1^q \dots KP_n^q\}$ and $KP^l = \{KP_1^l \dots KP_m^l\}$: keypoints extracted from the query q and the target line l respectively.
- 2: **Output:** C_{ql} : the best matching cost of the query q and the target line l .
 $best_matches(KP^q)$: the best matching keypoints of the query q .
- 3: $D(i,j) = X^2(KP_i^q, KP_j^l)$, **For** $i = 1 \rightarrow n$ and $j = 1 \rightarrow m$.
- 4: $C_{ql} = inf$;
- 5: **for** $j = 1 \rightarrow m-n$ **do**
- 6: $S_d \leftarrow 0$;
- 7: **for** $i = 1 \rightarrow n$ **do**
- 8: $match(i) \leftarrow argmin(D(i,k))$, **for** $k = j \rightarrow j+n$;
- 9: **end for**;
- 10: $S_d \leftarrow \sum(D(i,match(i)))$, **for** $i = 1 \rightarrow n$;
- 11: **if** $S_d < C_{ql}$ **then**
- 12: $C_{ql} \leftarrow S_d$;
- 13: $best_matches(i) = match(i)$, **for** $i = 1 \rightarrow n$;
- 14: **end if**
- 15: **end for**

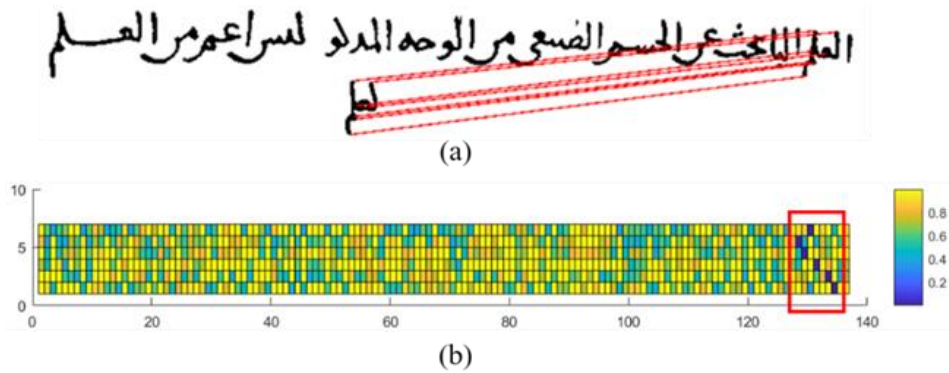


Figure 4.6. Un exemple de calcul de dissimilarité entre un mot de requête et une ligne. (a) L'image de la requête et les *matches* de ses points-clés dans la ligne cible. (b) La matrice de dissimilarité et les points-clés sélectionnés comme *matches* des points-clés de la requête.

4.4.1. Expérimentation et résultats

4.4.2. Les bases d'images

Nous avons utilisé deux ensembles de données pour l'évaluation de notre méthode IBN SINA [19] et IFN/ENIT [81]. La base d'images IBN SINA consiste en un manuscrit arabe historique intitulé "Kitab Kashf al-tamwihat sharh al-Tanbihat". Nous avons concentré nos évaluations sur cet ensemble de données car il offre la possibilité de montrer l'efficacité de notre méthode dans l'appariement des requêtes de mot avec des lignes de texte. L'ensemble de données IFN/ENIT est une base d'images des noms des villes tunisiennes. La base IFN/ENIT contient

des images de mots manuscrits qui ne nécessitent pas de segmentation. Nous avons utilisé cette base pour comparer notre méthode avec la littérature.

L'ensemble de données IBN SINA se compose de 50 feuilles d'un manuscrit arabe historique. Chaque page comprend en moyenne 19 lignes. Par conséquent, un total de 927 lignes est considéré dans l'évaluation de notre méthode.

IFN/ENIT est un ensemble de mots arabes des noms des villes tunisiens écrits par des centaines de scripteurs différents. Il comprend 26459 mots répartis dans quatre ensembles A, B, C et D. L'ensemble de données IFN/ENIT est très utilisé par les méthodes de repérage de mots nécessitant une segmentation en mots. Dans ce travail, nous l'avons utilisé pour comparer notre méthode avec la littérature.

4.4.3. Segmentation du document en lignes

Les images de la base IBN SINA sont segmentées en composantes connexes, dans notre cas les images du document doivent être segmentées en lignes. Pour cela, nous avons suivi une méthode ascendante basée sur la projection horizontale [88-92]. Premièrement, nous avons estimé l'emplacement des lignes de texte en utilisant l'histogramme de projection horizontale de l'image binaire de la page entière. Les maxima locaux (pics) dans la projection horizontale indiquent les emplacements des lignes dans la page (figure 4.7). Les pics dans la projection horizontale sont filtrés en utilisant la hauteur des pics et la distance entre deux pics. Ensuite, les pics supérieurs à un seuil minimal h_{\min} et séparés par une distance supérieure à un seuil de distance entre les pics d_{\min} sont considérés comme étant les emplacements des lignes. Ces deux paramètres sont choisis empiriquement. La figure 4.7 montre des pics sélectionnés dans la projection horizontale d'une page de document. Une fois les emplacements des lignes de texte estimés, chaque CC de la page du document est affecté à sa ligne la plus proche.

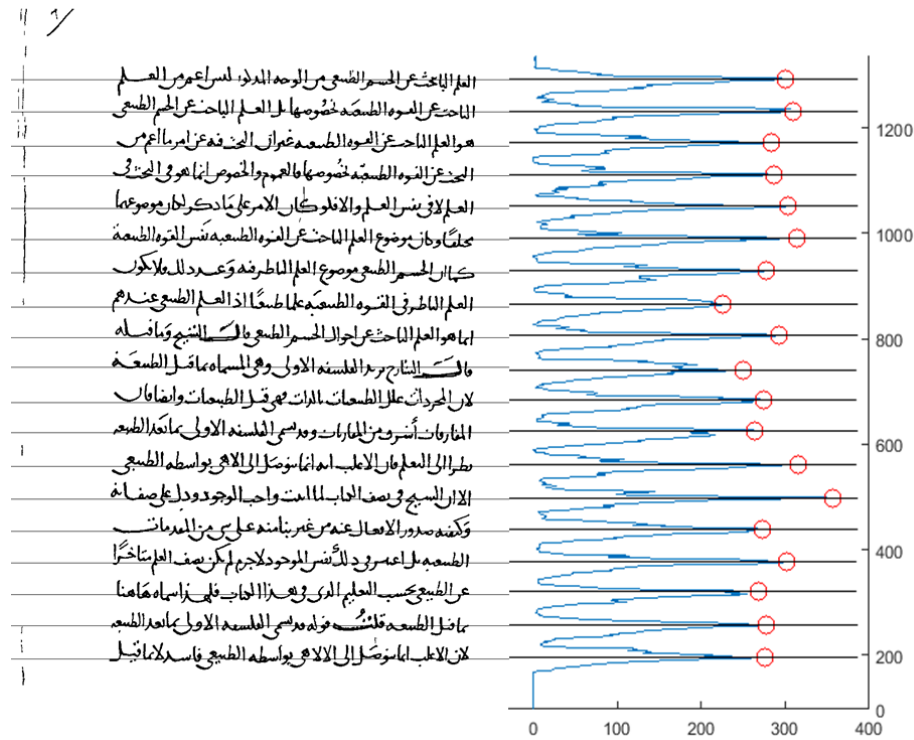


Figure 4.7. L'image binarisée de la page 11 de la base d'image IBN SINA et sa projection horizontale, les pics entourés dans la projection horizontale sont sélectionnés comme emplacements des lignes de texte.

4.4.4. Résultats qualitatifs

Les requêtes soumises au système de repérage peuvent être des sous-mots, des mots ou des phrases de plusieurs mots. Trois groupes de requêtes sont utilisés, un ensemble de 30 sous-mots, un ensemble de 18 mots et enfin 2 phrases. Ces requêtes sont sélectionnées au hasard dans la base d'image IBN SINA.

Lorsqu'une image requête est soumise au système pour repérage, elle est directement comparée avec les images des lignes du document cible et il n'y a aucune tâche de segmentation à effectuer dans l'étape de recherche. Les lignes du document cible sont ordonnées en fonction de leurs similarités avec l'image de la requête. Dans la figure 4.8, nous avons affiché les 10 meilleurs résultats récupérés pour quelques requêtes.

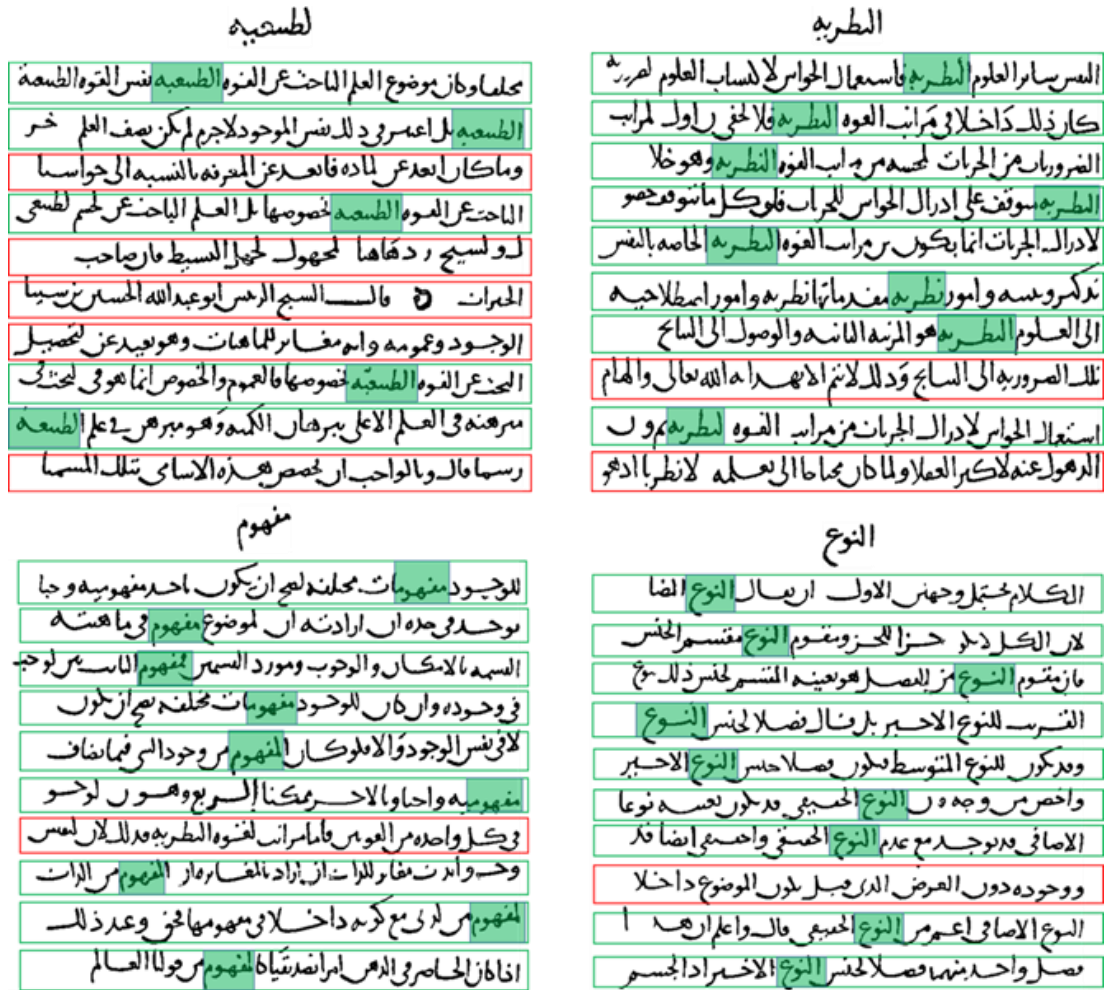


Figure 4.8. Les 10 premières lignes récupérées pour quelques requêtes. Les images en haut sont les requêtes.

Les positions des occurrences pertinentes dans les listes ordonnées pour l'ensemble des requêtes utilisées dans l'évaluation sont indiquées dans la figure 4.9. Les points noirs représentent des occurrences pertinentes dans la liste ordonnée retournée par l'algorithme. Notre méthode réussisse à récupérer la plupart des images pertinentes dans les rangs initiaux et il ne reste que peu d'occurrences à récupérer dans les rangs suivants.

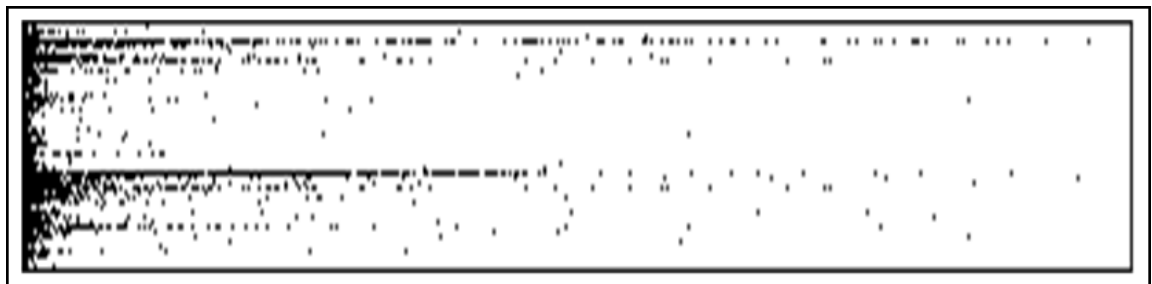


Figure 4.9. Positions des occurrences pertinentes dans les résultats. Chaque ligne représente les résultats récupérés pour une image de requête. Les points noirs représentent des occurrences pertinentes.

4.4.5. Résultats quantitatifs

Pour l'évaluation quantitative, nous avons adopté les métriques les plus utilisées dans la littérature pour l'évaluation des systèmes de repérage de mots [87]. Nous avons utilisé la précision, le rappel, la précision moyenne (AP) et la moyenne des précisions moyennes (MAP). Ces métriques sont présentées dans la section 3.4.2 du chapitre 3 et nous rappelons leurs définitions. Étant donné une requête q , sa dissimilarité avec chaque ligne du document cible est calculée. Les lignes sont ordonnées selon leurs dissimilarités avec la requête q . Le rappel (R) est défini comme la fraction des mots pertinents qui sont récupérés avec succès dans les résultats renvoyés à l'utilisateur (équation (4.4)).

$$R = \frac{|\{\text{relevant words}\} \cap \{\text{retrieved words}\}|}{|\{\text{relevant words}\}|} \quad (4.4)$$

La précision Pr est le taux de mots pertinents dans les résultats retournés par le système (équation (4.5)).

$$Pr = \frac{|\{\text{relevant words}\} \cap \{\text{retrieved words}\}|}{|\{\text{retrieved words}\}|} \quad (4.5)$$

Pour les systèmes de recherche d'information, l'utilisateur est satisfait tant que les résultats pertinents sont en début de la liste retournée. La précision dans les k premiers mots récupérés ($P@k$) est utilisée pour refléter cette propriété. $P@k$ est la précision de la méthode dans les k premières positions de la liste des résultats (équation (4.6)).

$$P@k = \frac{|\{\text{relevant words}\} \cap \{\text{best } k \text{ results}\}|}{k} \quad (4.6)$$

La moyenne des valeurs de la précision obtenue après la récupération de chaque occurrence pertinente est la précision moyenne (AP) qui est définie par l'équation (4.7).

$$AP = \frac{\sum_{k=1}^n (P@k \times R(k))}{|\{\text{relevant words}\}|} \quad (4.7)$$

Où $R(k)$ est une fonction binaire qui est égale à 1 si le $k^{\text{ième}}$ élément de la liste résultat est pertinent et 0 sinon. Enfin, la moyenne des précisions moyennes MAP est la valeur moyenne des AP obtenues pour chacune des requêtes utilisées dans l'évaluation.

Afin de calculer ces mesures, une ligne de la liste est considérée comme pertinente si elle contenait l'étiquette de la requête dans son propre étiquette. Le tableau 4.1 montre les résultats obtenus pour chaque groupe de requêtes utilisées dans l'évaluation. Nous avons reporté la moyenne des précisions moyennes MAP et la précision moyenne dans les k meilleures lignes dans la liste ordonnée (P@k). Nous avons considéré k comme le nombre de lignes de la liste ordonnée qui comprenait 50 % des occurrences pertinentes. Par conséquent, P@k représente la précision de la méthode lorsqu'elle récupère la moitié des occurrences pertinentes pour une requête donnée. Enfin, la figure 4.10 présente les courbes précision/rappel pour les requêtes des sous-mots et des mots. Ces courbes sont obtenues en considérant la précision interpolée pour des valeurs régulières de rappel. Les courbes précision/rappel pour les requêtes de sous-mots et les requêtes de mots sont très proches. Par conséquent, notre méthode est capable de récupérer les requêtes complexes avec la même efficacité que les requêtes simples.

Tableau 4.1. Résultats obtenus en termes de MAP et P@k moyenne.

Type de Requêtes	Nombre de requêtes	MAP (%)	Mean P@k (%)
sous-mots	30	75.15	91.79
mots	18	75.19	91.05
phrase	2	82.50	100
Toutes les requêtes	50	75.46	91.85

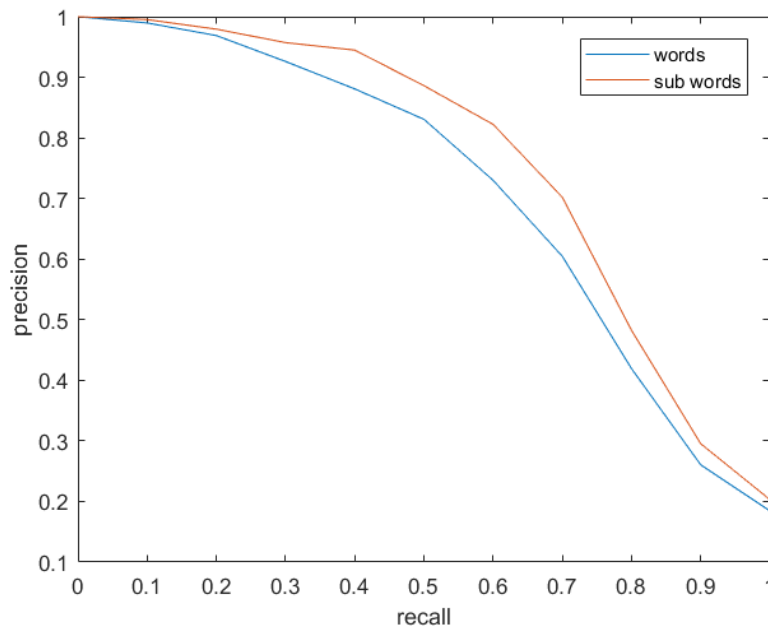


Figure 4.10. Courbes précision/rappel pour les requêtes de sous-mots et de mots.

Comme mentionné dans la section 4.3, nous avons déployé différentes configurations de zones directionnelles et circulaires pour l'extraction des descripteurs SDIK. Nous avons testé chacune des configurations illustrées dans la figure 4.3 (section 4.3). Les résultats obtenus pour ces configurations sont présentés dans le tableau 4.2. Nous avons constaté que les caractéristiques SDIK sont plus efficaces avec 8 zones directionnelles et avec quatre zones directionnelles avec les angles ($\pi/4$, $3\pi/4$, $5\pi/4$ et $7\pi/4$). Comme il n'y a pas de différence importante dans les résultats obtenus entre ces configurations, il est préférable d'extraire les caractéristiques SDIK avec quatre directions ($\pi/4$, $3\pi/4$, $5\pi/4$, $7\pi/4$) pour sa simplicité et pour avoir des vecteurs de caractéristiques de petites tailles. Cette configuration de zones directionnelles capture très bien la présence de lignes verticales et horizontales qui sont très fréquentes dans l'écriture arabe.

Tableau 4.2. Résultats obtenus en terme de MAP (%) pour différentes configurations de zones directionnelles et circulaires.

c	3	4	5	6
Zones directionnelles				
$(\pi/4, 3\pi/4, 5\pi/4, 7\pi/4)$	56.95	69.30	75.46	70.97
$(0, \pi/2, \pi, 3\pi/2)$	49.17	60.41	67.08	69.23
8 directional zones	62.97	71.76	75.70	75.82

Concernant les zones circulaires, nous avons testé notre méthode par plusieurs valeurs du rayon r et du nombre de zones circulaires c . Le tableau 4.3 montre la précision MAP obtenue pour différentes valeurs des deux paramètres r et c avec les quatre zones directionnelles ($\pi/4$, $3\pi/4$, $5\pi/4$, $7\pi/4$). Notez que la différence dans la précision MAP obtenue pour différentes valeurs de ces paramètres est insignifiante. De ce fait, le choix du rayon r et du nombre de zones circulaires c n'affectera pas les résultats de notre méthode. Les résultats fournis dans la section 4.4.4 (résultats qualitatifs) sont obtenus par un rayon de 4 pixels et 5 zones circulaires.

Tableau 4.3. Résultats obtenus en terme de MAP (%) pour différentes valeurs du rayon r et du nombre de zones circulaires c .

r	3	4	5	6
c				
3	45.82	56.95	64.77	67.59
4	58.89	69.30	72.96	70.01
5	68.02	75.46	72.86	69.76

4.4.6. Comparaison avec la littérature

Nous avons comparé notre méthode avec la littérature dans deux scénarios, avec les documents historiques de la base de données IBN SINA et avec l'écriture arabe moderne de la base IFN/ENIT. Suite au manque de travaux de repérage de mots pour l'écriture arabe, nous avons implémenté les deux types de caractéristiques très populaires dans la littérature, à savoir : les profils de mot (word profiles) et les histogrammes de gradient orienté (HOG) pour les comparer avec notre descripteur (SDIK).

Les caractéristiques de profil de mot (word profiles) ont d'abord été introduites pour la reconnaissance de l'écriture manuscrite par Marti et Bunke [93] et utilisées pour le repérage de mots pour l'écriture arabe et autres [13, 94, 95]. Pour chaque colonne de l'image d'entrée, nous avons calculé la projection verticale, le profil supérieur, le profil inférieur et le nombre de transitions entre l'arrière-plan et l'encre. Un vecteur de 4 caractéristiques est obtenu pour chaque colonne.

Les histogrammes du gradient orienté (HOG) sont très populaires dans les systèmes de repérage de mots [5, 7, 73]. Nous avons utilisé la caractéristique HOG en combinaison avec la caractéristique d'intensité comme fait par Wshah et al [5]. Pour extraire ces caractéristiques, nous avons utilisé une fenêtre glissante de droite à gauche sur l'image d'entrée. La fenêtre glissante a été divisée verticalement en deux cellules et la direction du gradient a été calculée à chaque pixel. Les directions du gradient ont été quantifiées uniformément dans huit directions pour chaque cellule. Ainsi, un histogramme de 16 bins (8 directions x 2 cellules) est obtenu pour les caractéristiques du gradient. Pour les caractéristiques d'intensité, la fenêtre glissante est divisée horizontalement en quatre régions et le taux de pixels noir est calculé. Le nombre total de caractéristiques d'intensité est de 8 (2 cellules verticales x 4 cellules horizontales). Ainsi, un histogramme de 24 bins est obtenu par trame (16 gradients + 8 intensités). Nous avons testé plusieurs valeurs de la largeur de la fenêtre glissante et de chevauchement et nous avons considéré les meilleurs résultats obtenus pour ces différentes configurations.

Comme les histogrammes HOG et les profils extraits d'une image de mot ont une longueur variable, nous avons utilisé l'algorithme DTW (Dynamic Time

Warping) pour mesurer la dissimilarité entre deux vecteurs de caractéristiques. DTW est un algorithme de programmation dynamique pour le calcul de la distance entre deux séries temporelles. Il est très populaire dans le repérage de mots, il a été utilisé pour la première fois dans ce domaine par Rath et Manmatha [82].

Le tableau 4.4 présente la précision moyenne MAP et le temps de réponse obtenus pour chaque méthode. Notre descripteur SDIK est plus discriminant par rapport aux autres méthodes en termes de précision moyenne et de temps de réponse. Le descripteur SDIK est plus efficace que les autres caractéristiques, que ce soit dans les documents historiques avec IBN SINA ou dans l'écriture manuscrite moderne avec IFN/ENIT. Par sa simplicité, notre descripteur est très rapide à comparer par rapport aux autres caractéristiques. Ceci est démontré par le temps moyen de réponse (Tr moyen). Nous avons obtenu de faibles résultats avec la base IFN/ENIT par rapport à IBN SINA. Cela est dû à la grande variabilité de cet ensemble de données, il est écrit par plus de 400 scripteurs. Notre descripteur a l'avantage de capturer les caractéristiques de l'écriture arabe sans consommer beaucoup d'espace mémoire. Il utilise uniquement le voisinage des points clés plutôt que l'ensemble des pixels de l'image. Notre descripteur a occupé 52,30 Mo en mémoire pour l'ensemble A de la base IFN/ENIT, mais les histogrammes HOG et les profils de mot ont pris respectivement 103,14 Mo et 88,66 Mo. Les méthodes marquées d'un astérisque dans le tableau 4.4 ne sont pas implémentées dans ce travail. Par conséquent, ils ne partagent pas le même protocole d'évaluation et ne peuvent donc pas être comparés directement à notre méthode. Cependant, nous les avons inclus pour donner une idée générale sur la compétitivité de notre méthode.

Tableau 4.4. La précision moyenne et le temps de réponse moyen (en secondes) du descripteur SDIK et les autres méthodes.

Méthode	IBEN SINA		IFN/ENIT	
	MAP (%)	Tr moyen	MAP (%)	Tr moyen
HOG & intensité	31.47	6.02 s	31.27	13.92 s
Profils de mot	41.42	6.08 s	42.25	23.95 s
Descripteur proposé (SDIK)	64.44	5.25 s	53.21	4.41 s
Local Gradient Histogram* [52]	-	-	46.1	-
Deep CNN* [11]	-	-	28.13	-

4.5. Conclusion

Dans ce chapitre, nous avons présenté notre contribution dans le repérage de mots dans les documents arabes. Nous avons proposé un nouveau descripteur pour l'extraction des caractéristiques des images de documents arabes. L'étude présentée dans ce chapitre a démontré l'efficacité de la caractéristique SDIK extraite à partir des points-clés dans le repérage de mots arabes que ce soit dans des documents modernes ou anciens. L'étude comparative effectuée a démontré que notre descripteur SDIK est compétitif avec la littérature.

Conclusion générale

Les anciens manuscrits sont un patrimoine à préserver soigneusement. Les nouvelles technologies sont venues à leur secours à travers la numérisation. La numérisation des anciens manuscrits a générée des banques d'images de grande taille et le développement d'applications pour faciliter l'utilisation de ces images s'impose. Le repérage de mots dans les images de document est parmi les applications les plus intéressantes, il permet de localiser un mot de requête dans une image de document. La problématique de repérage de mots est largement traitée dans la littérature, mais la majorité des méthodes proposées sont conçues pour les langages à caractères latins. Les méthodes proposées pour les langages latins ne peuvent être performantes avec les documents arabes en raison des caractéristiques particulières de la langue arabe. Nous citons le problème des frontières des mots qui ne sont pas facilement reconnaissable dans l'écriture arabe. La proposition d'une méthode de repérage de mots dans des images de document arabe s'impose. Un système de recherche de mot comprend plusieurs étapes à savoir : l'acquisition des images, la binarisation, l'extraction des caractéristiques et la comparaison de la requête avec les mots du document cible. Une image de document est une matrice à valeurs numériques et ne peut être utilisée de manière efficace par les algorithmes de recherche. L'extraction des caractéristiques permet de projeter les images dans un espace plus restreint et plus significatif. Des caractéristiques conçues pour représenter une image de document de langue latine ne peuvent représenter de manière efficace un document de la langue arabe. Dans cette thèse nous avons proposé une nouvelle méthode pour l'extraction des caractéristiques adaptée aux propriétés particulières de l'écriture arabe.

Notre contribution dans cette thèse consiste dans la proposition d'une nouvelle méthode d'extraction de caractéristiques d'une image de document arabe. Nous sommes partis de l'idée que pour comparer deux images de mots arabes, il n'est pas nécessaire de comparer la totalité des pixels. Par contre, il faut comparer les parties les plus discriminantes. Les traits de l'écriture arabes peuvent se prolonger horizontalement et créer ainsi des traits horizontaux parfois très longs. Nous avons supposé que l'information discriminante dans l'écriture arabe

se concentre dans les alentours des points de branchement et aux extrémités des traits.

Dans un premier temps, nous avons fait une étude préliminaire pour vérifier la faisabilité de l'utilisation des points de branchement et des points terminaux pour la comparaison des images de mots arabes. Dans cette étude nous avons proposé un algorithme pour l'extraction des points-clés. L'algorithme utilise le squelette de l'image pour extraire les points terminaux et les points de branchement, ces points sont ensuite nettoyés en utilisant deux paramètres : la largeur moyenne des traits de l'écriture ASW (Average Stroke width) et la distance en nombre de pixels de squelette entre les points-clés. La segmentation d'un document arabe en mots est une tâche complexe et vue qu'elle n'est pas l'objectif de notre étude, nous avons utilisé une base d'images de document segmentées en composantes connexes CC (sous-mots) et nous avons évalué notre méthode dans la recherche de sous-mots. Pour réduire le nombre de CCs à comparer avec une requête, nous avons proposé un algorithme pour segmenter les CCs en méta-classes. L'algorithme effectue une classification des CCs de manière souple avec un modèle statistique de type mélange de Gaussiennes, de sorte qu'une CC peut appartenir à plusieurs classes. Un vecteur de caractéristiques est extrait à partir de chaque points-clés des CCs et la distance euclidienne est utilisée pour calculer la dissimilarité entre les vecteurs de caractéristiques. Nous avons utilisé des requêtes de différentes tailles pour l'évaluation. La dissimilarité est calculée entre la requête et toutes les CCs appartenant à sa classe, les CCs sont ensuite ordonnés selon leur similarité avec la requête. Notre étude préliminaire a démontré que l'utilisation des points-clé pour l'extraction des caractéristiques est prometteuse pour le repérage de mots dans les images de documents arabes. Les résultats de notre étude préliminaire sont concrétisés par une communication orale dans la conférence ICPRS 2017.

Dans une seconde étude, nous avons proposé un descripteur appelé SDIK (Spatial Distribution of Ink at Keypoints). Le descripteur est conçu pour capturer les informations discriminantes de l'écriture arabe et faciliter la comparaison des images de mots arabes avec un document cible. Le descripteur SDIK capture la distribution spatiale de l'ancre dans le voisinage des points-clés. Les descripteurs SDIKs sont des vecteurs de tailles fixes et sont très rapides à comparer. Nous avons proposé un algorithme pour le calcul de similarité entre une image de mot requête et une ligne entière du document cible. L'algorithme calcule la distance

X^2 entre les descripteurs SDIKs de la requête et ceux de la ligne dans une matrice. L'algorithme effectue ensuite une recherche dans la matrice des distances pour retrouver un sous-ensemble de points-clés de la ligne qui minimise la dissimilarité avec la requête. Finalement, l'algorithme calcule la dissimilarité de chaque ligne du document avec la requête, les lignes sont ordonnées selon leurs similarités avec le mot requête. Notre méthode est évaluée en utilisant des images de document ancien (IBN SINA) et des images modernes des noms des villes tunisiennes (IFN/ENIT). Les résultats obtenus ont démontré l'efficacité de notre descripteur dans la comparaison des images de documents arabes. La comparaison avec deux types de méthodes d'extraction de caractéristiques très populaires dans la littérature est en faveur de notre descripteur SDIK.

Les travaux réalisés dans cette thèse ouvrent diverses perspectives de recherche. Dans un premier temps nous envisageons d'appliquer notre descripteur sur d'autres types de documents (documents Latins, Amazighs et autres). Nous envisageons aussi d'utiliser notre descripteur pour la reconnaissance de caractères avec un algorithme d'apprentissage automatique.

Bibliographie

1. E. Al-Wajih and R. Ghazali, An enhanced LBP-based technique with various size of sliding window approach for handwritten Arabic digit recognition, *Multimedia Tools and Applications*. **80**(16), 24399-24418 (2021), doi: 10.1007/s11042-021-10762-x
2. M. Gagaoua, H. Ghilas, A. Tari and M. Cheriet, Histogram of Marked Background (HMB) Feature Extraction Method for Arabic Handwriting Recognition, *International Journal of Image and Graphics*, 2250015 (2021), doi: 10.1142/S0219467822500152
3. H. Lamtougui, H. E. Moubtahij, H. Fouadi, A. Yahyaouy and K. Satori. Offline Arabic Handwriting Recognition Using Deep Learning: Comparative Study, in *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)*, 2020), p. 1-8
4. H. Q. Ghadhban, M. Othman, N. A. Samsudin, M. N. B. Ismail and M. R. Hammoodi. Survey of Offline Arabic Handwriting Word Recognition, in *Recent Advances on Soft Computing and Data Mining*, (°Springer International Publishing, 2020), p. 358-372
5. S. Wshah, G. Kumar and V. Govindaraju, Statistical script independent word spotting in offline handwritten documents, *Pattern Recognition*. **47**(3), 1039-1050 (2014), doi: <http://dx.doi.org/10.1016/j.patcog.2013.09.019>
6. A. Fischer, A. Keller, V. Frinken and H. Bunke, Lexicon-free handwritten word spotting using character HMMs, *Pattern Recognition Letters*. **33**(7), 934-942 (2012), doi: <http://dx.doi.org/10.1016/j.patrec.2011.09.009>
7. J. Almazán, A. Gordo, A. Fornés and E. Valveny, Segmentation-free word spotting with exemplar SVMs, *Pattern Recognition*. **47**(12), 3967-3978 (2014), doi: <http://dx.doi.org/10.1016/j.patcog.2014.06.005>
8. X. Zhang and C. L. Tan, Handwritten word image matching based on Heat Kernel Signature, *Pattern Recognition*. **48**(11), 3346-3356 (2015), doi: <http://dx.doi.org/10.1016/j.patcog.2014.10.028>
9. J. A. Rodriguez-Serrano and F. Perronnin, Synthesizing queries for handwritten word image retrieval, *Pattern Recognition*. **45**(9), 3270-3276 (2012), doi: <http://dx.doi.org/10.1016/j.patcog.2012.02.015>
10. Y. Liang, M. C. Fairhurst and R. M. Guest, A synthesised word approach to word retrieval in handwritten documents, *Pattern Recognition*. **45**(12), 4225-4236 (2012), doi: <http://dx.doi.org/10.1016/j.patcog.2012.05.024>
11. M. Mhiri, C. Desrosiers and M. Cheriet, Word spotting and recognition via a joint deep embedding of image and text, *Pattern Recognition*. **88**, 312-320 (2019), doi: <https://doi.org/10.1016/j.patcog.2018.11.017>

12. M. Khayyat, L. Lam and C. Y. Suen, Learning-based word spotting system for Arabic handwritten documents, *Pattern Recognition*. **47**(3), 1021-1030 (2014), doi: <http://dx.doi.org/10.1016/j.patcog.2013.08.014>
13. R. Jain, V. Frinken, C. V. Jawahar and R. Manmatha. BLSTM Neural Network Based Word Retrieval for Hindi Documents, in *International Conference on Document Analysis and Recognition*, (2011), p. 83-87
14. M. Rusiñol, D. Aldavert, R. Toledo and J. Lladós, Efficient segmentation-free keyword spotting in historical document collections, *Pattern Recognition*. **48**(2), 545-555 (2015), doi: <http://dx.doi.org/10.1016/j.patcog.2014.08.021>
15. P. Riba, J. Lladàs and A. Fornés. Handwritten word spotting by inexact matching of grapheme graphs, in *13th International Conference on Document Analysis and Recognition (ICDAR)*, (2015), p. 781-785
16. P. Wang, V. Eglin, C. Garcia, C. Langeron, J. Lladós and A. Fornés. A Novel Learning-Free Word Spotting Approach Based on Graph Representation, in *2014 11th IAPR International Workshop on Document Analysis Systems*, (2014), p. 207-211
17. T. Mondal, N. Ragot, J. Ramel and U. Pal. Flexible Sequence Matching Technique: Application to Word Spotting in Degraded Documents, in *2014 14th International Conference on Frontiers in Handwriting Recognition*, (2014), p. 210-215
18. M. Rusinol, D. Aldavert, R. Toledo and J. Lladós. Browsing Heterogeneous Document Collections by a Segmentation-Free Word Spotting Method, in *2011 International Conference on Document Analysis and Recognition*, (2011), p. 63-67
19. R. F. Moghaddam, M. Cheriet, M. M. Adankon, K. Filonenko and R. Wisnovsky, IBN SINA: a database for research on processing and understanding of Arabic manuscripts images, in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. 2010, ACM: Boston, Massachusetts, USA. p. 11-18.
20. N. Otsu, A Threshold Selection Method from Gray-Level Histograms, *IEEE Transactions on Systems, Man, and Cybernetics*. **9**(1), 62-66 (1979), doi: 10.1109/TSMC.1979.4310076
21. A. Tarafdar, U. Pal, J. Ramel, N. Ragot and B. B. Chaudhuri. Word Spotting in Bangla and English Graphical Documents, in *2014 22nd International Conference on Pattern Recognition*, (2014), p. 3044-3049
22. K. Zagoris, K. Ergina and N. Papamarkos, A Document Image Retrieval System, *Engineering Applications of Artificial Intelligence*. **23**(6), 872-879 (2010), doi: <https://doi.org/10.1016/j.engappai.2010.03.002>
23. E. Ataer and P. Duygulu, Matching ottoman words: an image retrieval approach to historical document indexing, in *Proceedings of the 6th ACM international conference on Image and video retrieval*. 2007, Association for Computing Machinery: Amsterdam, The Netherlands. p. 341-347.
24. E. F. Can and P. Duygulu, A line-based representation for matching words in historical manuscripts, *Pattern Recognition Letters*. **32**(8), 1126-1138 (2011), doi: <https://doi.org/10.1016/j.patrec.2011.02.013>

25. C. Tensmeyer and T. Martinez, Historical Document Image Binarization: A Review, *SN Computer Science*. **1**(3), 173 (2020), doi: 10.1007/s42979-020-00176-1
26. J. Sauvola and M. Pietikäinen, Adaptive document image binarization, *Pattern Recognition*. **33**(2), 225-236 (2000), doi: [https://doi.org/10.1016/S0031-3203\(99\)00055-2](https://doi.org/10.1016/S0031-3203(99)00055-2)
27. G. Kumar, Z. Shi, S. Setlur, V. Govindaraju and S. Ramachandrupa. Keyword Spotting Framework Using Dynamic Background Model, in *2012 International Conference on Frontiers in Handwriting Recognition*, 2012), p. 582-587
28. H. Cao, V. Govindaraju and A. Bhardwaj, Unconstrained handwritten document retrieval, *International Journal on Document Analysis and Recognition (IJDAR)*. **14**(2), 145-157 (2011), doi: 10.1007/s10032-010-0139-z
29. R. F. Moghaddam and M. Cheriet. Application of Multi-Level Classifiers and Clustering for Automatic Word Spotting in Historical Document Images, in *10th International Conference on Document Analysis and Recognition*, 2009), p. 511-515
30. H. Wei, G. Gao and Y. Bao. A Method for Removing Inflectional Suffixes in Word Spotting of Mongolian Kanjur, in *2011 International Conference on Document Analysis and Recognition*, 2011), p. 88-92
31. J. Canny, A Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **PAMI-8**(6), 679-698 (1986), doi: 10.1109/TPAMI.1986.4767851
32. B. Su, S. Lu and C. L. Tan, Robust Document Image Binarization Technique for Degraded Document Images, *IEEE Transactions on Image Processing*. **22**(4), 1408-1417 (2013), doi: 10.1109/TIP.2012.2231089
33. M. Valizadeh and E. Kabir, An adaptive water flow model for binarization of degraded document images, *International Journal on Document Analysis and Recognition (IJDAR)*. **16**(2), 165-176 (2013), doi: 10.1007/s10032-012-0182-z
34. T. Lore and F. Bouchara, FAIR: A Fast Algorithm for Document Image Restoration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **35**(8), 2039-2048 (2013), doi: 10.1109/TPAMI.2013.63
35. R. Hedjam, R. F. Moghaddam and M. Cheriet, A spatially adaptive statistical method for the binarization of historical manuscripts and degraded document images, *Pattern Recognition*. **44**(9), 2184-2196 (2011), doi: <https://doi.org/10.1016/j.patcog.2011.02.021>
36. B. Su, S. Lu and C. L. Tan. A learning framework for degraded document image binarization using Markov Random Field, in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012), p. 3200-3203
37. J. Long, E. Shelhamer and T. Darrell. Fully convolutional networks for semantic segmentation, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015), p. 3431-3440
38. J. Pastor-Pellicer, S. España-Boquera, F. Zamora-Martínez, M. Z. Afzal and M. J. Castro-Bleda. Insights on the Use of Convolutional Neural Networks for

- Document Image Binarization, in *Advances in Computational Intelligence*, (°Springer International Publishing, 2015), p. 115-126
39. J. Calvo-Zaragoza and A.-J. Gallego, A selectional auto-encoder approach for document image binarization, *Pattern Recognition*. **86**, 37-47 (2019), doi: <https://doi.org/10.1016/j.patcog.2018.08.011>
 40. V. Papavassiliou, T. Stafylakis, V. Katsouros and G. Carayannis, Handwritten document image segmentation into text lines and words, *Pattern Recognition*. **43**(1), 369-377 (2010), doi: <https://doi.org/10.1016/j.patcog.2009.05.007>
 41. N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos and N. Papamarkos, Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths, *Image and Vision Computing*. **28**(4), 590-604 (2010), doi: <https://doi.org/10.1016/j.imavis.2009.09.013>
 42. M. Feldbach and K. D. Tönnies. Robust line detection in historical church registers, in *Joint Pattern Recognition Symposium*, (°Springer, 2001), p. 140-147
 43. G. Louloudis, B. Gatos, I. Pratikakis and C. Halatsis, Text line and word segmentation of handwritten documents, *Pattern Recognition*. **42**(12), 3169-3183 (2009), doi: <https://doi.org/10.1016/j.patcog.2008.12.016>
 44. J. A. Rodríguez-Serrano and F. J. P. R. Perronnin, Handwritten word-spotting using hidden Markov models and universal vocabularies. **42**(9), 2106-2116 (2009), doi:
 45. F. M. Wahl, K. Y. Wong, R. G. J. C. G. Casey and I. Processing, Block segmentation and text extraction in mixed text/image documents. **20**(4), 375-390 (1982), doi:
 46. T. Mondal, N. Ragot, J.-Y. Ramel and U. Pal. Flexible sequence matching technique: Application to word spotting in degraded documents, in *2014 14th International Conference on Frontiers in Handwriting Recognition*, (°IEEE, 2014), p. 210-215
 47. T. Mondal, N. Ragot, J.-Y. Ramel and U. Pal. Performance evaluation of DTW and its variants for word spotting in degraded documents, in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, (°IEEE, 2015), p. 1141-1145
 48. A. L. Kesidis, E. Galiotou, B. Gatos, I. J. I. J. O. D. A. Pratikakis and Recognition, A word spotting framework for historical machine-printed documents. **14**(2), 131-144 (2011), doi:
 49. M. Khayyat, L. Lam and C. Y. Suen. Arabic handwritten word spotting using language models, in *2012 International Conference on Frontiers in Handwriting Recognition*, (°IEEE, 2012), p. 43-48
 50. R. Saabni and J. J. T. El-Sana, Keyword searching for Arabic handwritten documents. **11**, 716-722 (2008), doi:
 51. T. Sari and A. Kefali, A search engine for Arabic documents, in *Actes du dixième Colloque International Francophone sur l'Écrit et le Document*. 2008. p. 97-102.
 52. J. A. Rodríguez-Serrano and F. Perronnin, A Model-Based Sequence Similarity with Application to Handwritten Word Spotting, *IEEE Transactions on Pattern*

- Analysis and Machine Intelligence*. **34**(11), 2108-2120 (2012), doi: 10.1109/TPAMI.2012.25
53. A. Fischer, A. Keller, V. Frinken and H. Bunke. HMM-based Word Spotting in Handwritten Documents Using Subword Models, in *2010 20th International Conference on Pattern Recognition*, 2010), p. 3416-3419
 54. Z. Zhong, W. Pan, L. Jin, H. Mouchère and C. Viard-Gaudin. Spottingnet: Learning the similarity of word images with convolutional neural network for word spotting in handwritten historical documents, in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, (°IEEE, 2016), p. 295-300
 55. V. Frinken, A. Fischer, R. Manmatha and H. Bunke, A Novel Word Spotting Method Based on Recurrent Neural Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **34**(2), 211-224 (2012), doi: 10.1109/TPAMI.2011.113
 56. T. M. Rath and R. Manmatha, Word spotting for historical documents, *International Journal of Document Analysis and Recognition (IJ DAR)*. **9**(2), 139-152 (2007), doi: 10.1007/s10032-006-0027-8
 57. A. Papandreou, B. Gatos and K. Zagoris. An adaptive zoning technique for word spotting using dynamic time warping, in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, (°IEEE, 2016), p. 387-392
 58. K. Khurshid, C. Faure and N. J. P. R. Vincent, Word spotting in historical printed documents using shape and sequence comparisons. **45**(7), 2598-2609 (2012), doi:
 59. J. A. Rodriguez and F. Perronnin. Local gradient histogram features for word spotting in unconstrained handwritten documents, 2008), p. 7-12
 60. T. Adamek, N. E. O'connor, A. F. J. I. J. O. D. A. Smeaton and Recognition, Word matching using single closed contours for indexing handwritten historical documents. **9**(2-4), 153-165 (2007), doi:
 61. K. Terasawa and Y. Tanaka. Slit Style HOG Feature for Document Image Word Spotting, in *2009 10th International Conference on Document Analysis and Recognition*, 2009), p. 116-120
 62. T. M. Rath and R. Manmatha. Word image matching using dynamic time warping, in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 2003), p. II-521-II-527 vol.2
 63. T. M. Rath and R. Manmatha. Features for word spotting in historical manuscripts, in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, 2003), p. 218-222 vol.1
 64. A. Fischer, V. Frinken, H. Bunke and C. Y. Suen. Improving hmm-based keyword spotting with character language models, in *2013 12th International Conference on Document Analysis and Recognition*, (°IEEE, 2013), p. 506-510
 65. V. Frinken, A. Fischer, H. Bunke and R. Manmatha. Adapting BLSTM neural network based keyword spotting trained on modern data to historical documents, in *2010 12th International Conference on Frontiers in Handwriting Recognition*, (°IEEE, 2010), p. 352-357

66. A. H. Toselli and E. Vidal. Fast HMM-filler approach for key word spotting in handwritten documents, in *2013 12th International Conference on Document Analysis and Recognition*, (°IEEE, 2013), p. 501-505
67. A. Papandreou, B. Gatos and G. Louloudis, An adaptive zoning technique for efficient word retrieval using dynamic time warping, in *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*. 2014, Association for Computing Machinery: Madrid, Spain. p. 147–152.
68. G. Louloudis, A. L. Kesidis and B. Gatos. Efficient word retrieval using a multiple ranking combination scheme, in *2012 10th IAPR International Workshop on Document Analysis Systems*, (°IEEE, 2012), p. 379-383
69. T. Konidakis, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, S. J. J. I. J. O. D. A. Perantonis and Recognition, Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. **9**(2), 167-177 (2007), doi:
70. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005), p. 886-893 vol. 1
71. D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*. **60**(2), 91-110 (2004), doi: 10.1023/B:VISI.0000029664.99615.94
72. J. Almazán, A. Fornés and E. Valveny. Deformable HOG-Based Shape Descriptor, in *2013 12th International Conference on Document Analysis and Recognition*, 2013), p. 1022-1026
73. J. A. Rodríguez and F. Perronnin, Local gradient histogram features for word spotting in unconstrained handwritten documents, in *International Conference on Frontiers in Handwriting Recognition*. 2008.
74. T. Ahonen, A. Hadid and M. Pietikainen, Face Description with Local Binary Patterns: Application to Face Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **28**(12), 2037-2041 (2006), doi: 10.1109/TPAMI.2006.244
75. S. Dey, A. Nicolaou, J. Lladós and U. Pal. Local binary pattern for word spotting in handwritten historical document, in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, (°Springer, 2016), p. 574-583
76. A. P. Giotis, D. P. Gerogiannis and C. Nikou. Word Spotting in Handwritten Text Using Contour-Based Models, in *2014 14th International Conference on Frontiers in Handwriting Recognition*, 2014), p. 399-404
77. S. Sudholt and G. A. Fink. PHOCNet: A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents, in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016), p. 277-282
78. J. Almazán, A. Gordo, A. Fornés, E. J. I. T. O. P. A. Valveny and M. Intelligence, Word spotting and recognition with embedded attributes. **36**(12), 2552-2566 (2014), doi:

79. J. T. Favata and G. Srikantan, A multiple feature/resolution approach to handprinted digit and character recognition, *International Journal of Imaging Systems and Technology*. **7**(4), 304-311 (1996), doi: [https://doi.org/10.1002/\(SICI\)1098-1098\(199624\)7:4<304::AID-IMA5>3.0.CO;2-C](https://doi.org/10.1002/(SICI)1098-1098(199624)7:4<304::AID-IMA5>3.0.CO;2-C)
80. A. Vinciarelli and J. Luettin, Off-Line Cursive Script Recognition Based on Continuous Density HMM, (2001), doi:
81. M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze and H. Amiri, IFN/ENIT-DATABASE OF HANDWRITTEN ARABIC WORDS, in *7th Colloque International Francophone sur l'Ecrit et le Document , CIFED*. 2002: Hammamet, Tunis.
82. T. M. Rath and R. Manmatha. Word image matching using dynamic time warping, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Proceedings.*, 2003), p. II-521-II-527 vol.2
83. M. Cheriet and R. F. Moghaddam, A Robust Word Spotting System for Historical Arabic Manuscripts, in *Guide to OCR for Arabic Scripts*, V. Märgner and H. El Abed, Editors. 2012, Springer London: London. p. 453-484.
84. R. Saabni and A. Bronstein. Fast Key-Word Searching via Embedding and Active-DTW, in *International Conference on Document Analysis and Recognition*, 2011), p. 68-72
85. R. Farrahi Moghaddam, M. Cheriet, M. Adankon, K. Filonenko and R. Wisnovsky, *IBN SINA: A database for research on processing and understanding of Arabic manuscripts images*. 2010. 11-18.
86. G. McLachlan and D. Peel, ML Fitting of Mixture Models, in *Finite Mixture Models*. 2000, John Wiley & Sons, Inc. p. 40-80.
87. A. P. Giotis, G. Sfikas, B. Gatos and C. Nikou, A survey of document image word spotting techniques, *Pattern Recognition*. **68**, 310-332 (2017), doi: <https://doi.org/10.1016/j.patcog.2017.02.023>
88. 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
89. U. Pal and S. Datta. Segmentation of Bangla unconstrained handwritten text, in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, 2003), p. 1128-1132
90. Z. Shi, S. Setlur and V. Govindaraju. A Steerable Directional Local Profile Technique for Extraction of Handwritten Arabic Text Lines, in *2009 10th International Conference on Document Analysis and Recognition*, 2009), p. 176-180
91. A. Zahour, L. Likforman-Sulem, W. Boussellaa and B. Taconet. Text Line Segmentation of Historical Arabic Documents, in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2007), p. 138-142

92. X. Zhang and C. L. Tan. Text Line Segmentation for Handwritten Documents Using Constrained Seam Carving, in *2014 14th International Conference on Frontiers in Handwriting Recognition*, 2014), p. 98-103
93. U. V. Marti and H. Bunke, Using a Statistical Language Model to Improve the Performance of An HMM-Based Cursive Handwriting Recognition System, *International Journal of Pattern Recognition and Artificial Intelligence*. **15**(01), 65-90 (2001), doi: 10.1142/S0218001401000848
94. J. A. Rodríguez-Serrano and F. Perronnin, Handwritten word-spotting using hidden Markov models and universal vocabularies, *Pattern Recognition*. **42**(9), 2106-2116 (2009), doi: <http://dx.doi.org/10.1016/j.patcog.2009.02.005>
95. H. Cao, A. Bhardwaj and V. Govindaraju, A probabilistic method for keyword retrieval in handwritten document images, *Pattern Recognition*. **42**(12), 3374-3382 (2009), doi: <https://doi.org/10.1016/j.patcog.2009.02.003>

Résumé - Le travail réalisé dans le cadre de cette thèse s'inscrit dans la restauration des anciens manuscrits arabes. La numérisation est la meilleure solution pour protéger ce type de documents des dégradations physiques et celles liées à la manipulation manuelle. L'indexation et la recherche d'informations dans les images de document devient une nécessité. Le repérage de mots (word spotting) consiste à repérer dans une collection d'images de document toutes les occurrences d'un mot de requête donné. La recherche de mots dans les images de document est très avancée pour les langages Latins mais il y a très peu de travaux pour la recherche de mots dans les documents arabes. Dans cette thèse nous avons proposé une nouvelle méthode pour la recherche de mots dans les images de manuscrits arabes. Nous avons contribué dans la phase d'extraction des caractéristiques par la proposition d'un nouveau descripteur appelé SDIK (Spatial Distribution of ink at Keypoints). Nous avons d'abord effectué une étude préliminaire qui a démontré l'efficacité de l'extraction des caractéristiques dans le voisinage des points-clés. Dans une deuxième étude nous avons élaboré notre méthode d'extraction des caractéristiques (SDIK). Le descripteur SDIK permet de capturer la répartition spatiale des pixels d'ancre dans le voisinage d'un point-clé. Nous avons utilisé les voisinages des points terminaux (endpoints) et des points de branchement (branch points) pour extraire les descripteurs SDIK et nous avons proposé un algorithme pour l'extraction de ces points-clés. Nos descripteurs SDIK sont des histogrammes de tailles fixes et sont très rapides à comparer. Ainsi notre méthode compare un mot requête avec des lignes entières du document cible par un algorithme que nous avons proposé. Par cet algorithme de recherche, nous avons évité la segmentation d'un document arabe en mots qui est une tâche très difficile. Les évaluations de notre méthode effectuées sur des images de document anciens (IBN SINA) et des images de l'écriture moderne (IFN/ENIT) ont démontré l'efficacité de notre descripteur par rapport à la littérature.

Mots-clés : repérage de mot, extraction des caractéristiques, images de document, anciens documents arabes.

Abstract - The proposed method in this thesis contributes in the restoration of historical Arabic documents. Digitization is the best solution to protect these documents from physical degradations and human artifacts. Indexing and searching for information in document images becomes a necessity. Word spotting aims at locating all occurrences of a given query word in a collection of document images. Word spotting is very advanced for Latin languages, but there is a few works for Arabic documents. In this thesis, we have proposed a new method for word spotting in Arabic manuscripts. We have contributed in the feature extraction step by proposing a new descriptor called SDIK (Spatial Distribution of ink at Keypoints). First, we have performed a preliminary study, in which we have proved the efficiency of extracting features in the neighborhood of key points. In a second study, we developed our feature extraction method (SDIK). The SDIK descriptor captures the spatial distribution of ink pixels in the neighborhood of a keypoint. We have used endpoints and branch points as salient locations to extract SDIK descriptors and we have proposed an algorithm to extract these keypoints. The resulting SDIK features are very fast to match; we take this advantage to propose an algorithm to match a query word with lines images rather than words. By this matching mechanism, we overcome the hard task of segmenting an Arabic document into words. The proposed method in this thesis is tested on historical Arabic document with IBN SINA dataset and on modern handwriting with IFN/ENIT database. The obtained results are great of interest for retrieving query words in an Arabic document.

Keywords: word spotting; information retrieval; image matching; features extraction; Arabic documents.

ملخص - العمل المنجز في هذه الأطروحة يساهم في الحفاظ على المخطوطات العربية القديمة. تعتبر الرقمنة هي الحل الأفضل لحماية هذه الوثائق من التدهور المادي ومخلفات العوامل البشرية. مع توفر قواعد بيانات للمخطوطات المصورة، أصبح توفير برامج للفهرسة والبحث عن المعلومات في هذه الصور ضرورة. تهدف برامج اكتشاف الكلمات (word spotting) إلى تحديد كل مواقع كلمة بحث معينة في مجموعة من صور ناتجة عن رقمنة المخطوطات. شهدت أنظمة اكتشاف الكلمات تقدماً ملحوظاً بالنسبة للغات ذات الأحرف اللاتينية، ولكن هناك نقص كبير جداً من أنظمة اكتشاف الكلمات بالنسبة للغة العربية. في هذه الأطروحة اقترحنا طريقة جديدة للبحث عن الكلمات في المخطوطات العربية. لقد ساهمنا في مرحلة استخراج الميزات من خلال اقتراح واصف جديد يسمى SDIK (Spatial Distribution of Ink at Keypoints). أولاً، أجرينا دراسة أولية، أظهرنا فيها كفاءة استخراج الميزات من النقاط الرئيسية. في دراسة ثانية، قمنا بتطوير طريقة استخراج الميزات (SDIK). يلتقط واصف SDIK التوزيع المكاني ليكسالات الحبر في المناطق المجاورة للنقاط الرئيسية. لقد استخدمنا نقاط النهاية ونقاط الفروع كمواقع بارزة لاستخراج واصفات SDIK واقترحنا خوارزمية لاستخراج هذه النقاط الرئيسية. ميزات SDIK الناتجة سريعة جداً في المطابقة؛ استفدنا من هذه الميزة لاقتراح خوارزمية لمطابقة كلمة البحث مع أسطر كاملة بدلاً من الكلمات. من خلال آلية المطابقة المقترحة، قمنا بتفادي عملية تقسيم مستند عربي إلى كلمات والتي تعتبر عملية معقدة. تم اختبار الطريقة المقترحة في هذه الأطروحة على المخطوطات العربية القديمة المتمثلة في قاعدة البيانات IBN SINA وعلى الكتابة اليدوية الحديثة بواسطة قاعدة البيانات IFN/ENIT. تعتبر النتائج التي تم الحصول عليها ذات أهمية كبيرة لاسترجاع الكلمات في مستند عربي.

الكلمات الرئيسية: اكتشاف الكلمات، استرجاع المعلومات، مطابقة الصور، استخراج الميزات من الصور، المخطوطات القديمة، المخطوطات العربية.