

THÈSE

Présentée par

GAGAOUA Meriem

Pour l'obtention du grade de

DOCTEUR EN SCIENCES

Filière : Informatique

Option : Cloud computing

Thème

Reconnaissance de l'écriture arabe manuscrite

Soutenue le : 23/11/2021

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 Faïçal	Professeur	ESTIN, Bejaia	Examineur
Mr SEBAA Abderrazak	MCA	ESTIN, Bejaia	Examineur

Année Universitaire : 2021/2022

Remerciement

Je tiens tout d'abord à remercier Dieu le tout puissant, qui m'a donné la force et la patience d'accomplir ce modeste travail. « الحمد لله الذي بنعمته تتم الصالحات »

Je tiens à remercier Monsieur **TARI Abdelkamel**, mon directeur de thèse pour sa disponibilité, son écoute, ses conseils et ses encouragements durant toutes ces années. Et Merci De m'avoir encadré.

Je tiens à remercier **Mr CHERIET Mohamed** de m'avoir accueilli au sein de son laboratoire « SYNCHROMÉDIA – Laboratoire de communications multimédias en téléprésence » à Ecole des technologies supérieurs (ETS) CANADA. Et pour son écoute et ses conseils.

Je remercie Monsieur **AZNI Mohamed** Maître de conférences à l'université de Béjaia d'avoir accepté de présider le jury de cette thèse.

Je remercie aussi Monsieur **AZOUAOU Faycal** professeur à ESTIN de Béjaia de m'avoir fait l'honneur d'accepter de juger ce travail.

Je remercie aussi Monsieur **SEBAA ABDERRAZAK** Maître de conférences à ESTIN, Béjaia de m'avoir fait l'honneur d'accepter de juger ce travail.

Je remercie mes parents sans eux je n'arriverais jamais là.

Je remercie mon mari **GHILAS HAMZA** de m'avoir aidé et d'avoir fait équipe avec moi dans la réalisation de ce travail, et d'être passionné durant toutes ses années. Milles Mercis.

Je remercie Mon frère Mohamed de m'avoir aidé avec ses remarques pertinentes et de m'avoir soutenu.

Je remercie ma sœur, mes frères, et mes amies qui m'ont soutenu durant la période de préparation de ce travail.

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

Dédicace

Je dédie ce modeste travail à tous ceux qui me sont chères et j'aime...

Et spécialement à mon petit Mohamed (moumouh).

Liste des travaux

Revue internationale avec comités de lecture

GAGAOUA, Meriem, GHILAS, Hamza, TARI, Abdelkamel, et *CHERIET, Mohamed*. Histogram of marked background (HMB) feature extraction method for arabic handwriting recognition. *International Journal of Image and Graphics*, 2021, p. 2250015.

GHILAS, Hamza, GAGAOUA, Meriem, TARI, Abdelkamel, et *CHERIET, Mohamed*. Spatial Distribution of Ink at Keypoints (SDIK): a novel feature for word spotting in Arabic documents. *International Journal of Image and Graphics*, 2021, in press

Communications avec acte et comités de lecture

GAGAOUA, Meriem, GHILAS, Hamza, TARI, Abdelkamel, et *CHERIET, Mohamed*. 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. p. 70-75.

GHILAS, Hamza, GAGAOUA, Meriem, TARI, Abdelkamel, et *CHERIET, Mohamed*, "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.

Table des matières

Liste des figures	IV
Liste des tableaux.....	VII
Introduction Générale.....	10
1.Reconnaissance de l'écriture arabe manuscrite.....	14
1.1. Généralité sur le traitement d'images	14
1.1.1. Définition d'une image	14
1.1.2. Représentation d'une image numérique.....	15
1.1.3. Caractéristiques d'une image numérique.....	15
1.1.3.1. Dimension :.....	15
1.1.3.2. Résolution	15
1.1.3.3. Bruit	16
1.1.3.4. Luminance.....	16
1.1.3.5. Contraste.....	16
1.1.3.6. Images à niveaux de gris	16
1.1.3.7. Images en couleurs.....	16
1.1.4. Les traitements liés à l'image numérique.....	17
1.1.5. Applications de traitements d'images.....	17
1.2. Description et caractéristiques de l'écriture arabe	17
1.2.1. Caractéristiques de l'écriture arabe.....	19
1.3. Le processus de reconnaissance de l'écriture arabe.....	21
1.3.1. Les Approches de reconnaissance :.....	21
1.3.1.1. Approche globale (holistique) :.....	21
1.3.1.2. Approche analytique :.....	22
1.3.2. Les étapes de reconnaissance de l'écriture :.....	22
1.3.2.1. Acquisition.....	24

1.3.2.2.	Prétraitement	24
1.3.2.3.	Segmentation	27
1.3.2.4.	Extraction d'attributs	30
1.3.2.5.	Classification (reconnaissance) :	31
1.3.2.6.	Post traitement :	37
1.4.	Les problèmes rencontrés dans la reconnaissance de l'écriture arabe manuscrite	38
1.5.	Conclusion	39
2.	Etat de l'art sur l'extraction des Attributs de reconnaissance de l'écriture arabe.....	40
2.1.	GSC (gradient, structurel et concavités).....	40
2.2.	Attributs directionnels.....	41
2.3.	Histogrammes.....	42
2.4.	Motifs binaires locaux (LBP)	43
2.5.	Les moments	44
2.6.	Descripteurs de fourrier	45
2.7.	Transformation des caractéristiques invariantes à l'échelle.....	45
2.8.	Les attributs structurels	48
2.9.	Conclusion	51
3.	HMB nouvel attribut pour la reconnaissance de l'écriture arabe manuscrite.	52
3.1.	Extraction des attributs : Histogramme du fond marqué (HMB).....	52
3.1.1.	Marquage du fond.....	54
3.1.2.	Exemples sur le processus de marquage	57
3.1.2.1.	Calcul de la valeur décimale de la marque	58
3.1.2.2.	Exemples de marquage	59
3.1.3.	Calcul des histogrammes.....	63
3.1.3.1.	Préparation des images.....	63
3.1.3.2.	Calcul des vecteurs de caractéristiques	64
3.1.3.3.	Exemples de calcul des Histogrammes HMB	65
3.2.	Avantages de l'attribut HMB.....	67

3.3. Conclusion	67
4. Expérimentations et résultats	69
4.1. Bases de données	69
4.1.1. IBN SINA	69
4.1.2. IFHCDB	70
4.2. Les systèmes de références	71
4.2.1. Préparations des données	71
4.2.2. Les attributs de distributions	72
4.2.3. Les attributs directionnels	72
4.2.4. Les attributs structurels	73
4.2.5. Les attributs de concavités	73
4.3. Outil de reconnaissance	74
4.4. Configuration des paramètres du système	75
4.5. Discussions des résultats	76
4.6. Conclusion	80
Conclusion Générale	82
Bibliographie	85

Liste des figures

Figure 1.1. Caractéristiques de l'écriture arabe.....	19
Figure 1.2. Quelques lettres partageant le même corps et différentes par les points diacritiques	20
Figure 1.3. Exemple de variabilité dans l'écriture arabe manuscrite	20
Figure 1.4. Liste des diacritiques	21
Figure 1.5. L'organisation générale d'un système de reconnaissance de l'écriture.	23
Figure 1.6. Opérations de prétraitements.....	24
Figure 1.7. Les différentes étapes appliquées pour passer d'une image d'un document à un ensemble d'images de mots. 1) Segmentation en zones ; 2) Segmentation en lignes ; 3) Segmentation en mots	27
Figure 1.8. Segmentation par projection. b) horizontale. c) verticale	28
Figure 1.9. Segmentation basée sur le tracé du contour	29
Figure 1.10. Segmentation basée sur la squelettisation	29
Figure 1.11. Extraction manuelle et automatique des attributs dans un système de reconnaissance.	31
Figure 1.12. Représentations schématisées du principe de K-NN	32
Figure 1.13. Représentation schématisée d'un SVM.....	33
Figure 1.14. Représentation schématisée des topologies HMMs	34
Figure 1.15. Représentation schématisée des structures HMM pour la reconnaissance de (a) mot isolé et (b) mot connecté	34
Figure 1.16 La reconnaissance avec les modèles de Markov cachés. a) L'image d'un mot ; b) La séquence d'observations ; c) Un modèle de Markov caché utilisé pour calculer la probabilité d'un mot par rapport à la séquence d'observations	36
Figure 1.17. La reconnaissance avec les RNN	37
Figure 1.18. Problèmes liés à l'écriture manuscrite.....	39
Figure 2.1. Histogramme d'orientation de contour. a) l'image d'un caractère ; b) contour du caractère; c) plan d'orientation; d) histogrammes d'orientation local	42
Figure 2.2. Histogramme de gradients orientés.....	43
Figure 2.3. Histogramme de projection verticale et horizontale	43

Liste des figures

Figure 2.4. Exemple de calcul LBP : (a) une fenêtre de 3x3 centrée sur un pixel, (b) les niveaux de gris dans la fenêtre, (c) seuillage du voisinage par rapport au pixel x, (d) l'image d'un caractère après prétraitement, (e) résultat de l'application de LBP sur l'image dans (d).	44
Figure 2.5. Une représentation schématique du descripteur SIFT pour un patch de 16×16 pixels et un tableau de descripteurs 4×4	47
Figure 2.6. Les points d'angle en appliquant le détecteur de Harris	47
Figure 2.7. Plusieurs descripteurs SIFT.....	48
Figure 2.8. Les types de points (a) points terminaux; (b) points de branchement; (c) points de croisement	48
Figure 2.9. Extraction de : a)points diacritiques. b) boucles	49
Figure 2.10. Extraction ascendants (Stem) de deux mots différents	49
Figure 2.11 Extraction de descendeurs. a) point d'intersection du mot avec la ligne de base. b) quelques descendants (Leg)	50
Figure 3.1. Représentation de la distribution spéciale des pixels d'une image ...	53
Figure 3.2. Schéma général de l'extraction des attributs HMB. a) image originale. b) image marquée. c) Les attributs HMBs	54
Figure 3.3. Les différentes configurations de directions utilisées dans le processus de marquage.....	55
Figure 3.4. Configuration avec 4 directions.....	57
Figure 3.5 : Toutes les possibilités de marques avec quatre directions et leur code binaire ainsi que leurs signes décimaux correspondants.....	58
Figure 3.6. Exemple de calcul de la marque décimale d'un pixel de fond ayant un seul voisin.	58
Figure 3.7. Exemple de calcul de la marque décimale d'un pixel de fond ayant un plusieurs voisins.	59
Figure 3.8. Exemple de marquage d'un pixel de fond ayant un seul voisin d'encre	59
Figure 3.9. Exemple de marquage d'un pixel de fond ayant un voisin d'encre dans toutes les directions.	60
Figure 3.10. Exemple de marquage d'un pixel de fond ayant un voisin d'encre dans deux directions.	60
Figure 3.11. Exemple de marquage d'un pixel de fond ayant aucun voisin d'encre.....	61

Liste des figures

Figure 3.12. Processus de marquage. a) L'image originale. b) Partie de l'image sélectionnée dans (a). c) Résultat du marquage de l'image dans (b).	61
Figure 3.13. Exemple d'une image marquée.....	62
Figure 3.14. Images marquées avec des couleurs. a) Quelques images marquées avec différentes valeurs du rayon r (2, 4 et 6). b) Tableau des couleurs utilisées pour imprimer les pixels marqués en (a).	63
Figure 3.15. Divisions en quatre blocks horizontaux des images de texte marquées.	64
Figure 3.16 Calcul d'histogramme. a) Une image marquée divisée horizontalement en quatre blocs. h) Le HMB extrait de la frame f1 dans l'image marquée (a). h1), h2), h3) et h4) sont des HMBs extraits respectivement des blocs 1, 2, 3 et 4 de la frame f1.	65
Figure 3.17. Calcul d'histogramme. a) Une image marquée divisée horizontalement en quatre blocs. h1), h2) Les HMBs extraient du bloc1 et bloc2 respectivement de la frame f1 dans l'image marquée (a).....	66
Figure 3.18 Calcul d'histogramme. a) Une image marquée divisée horizontalement en quatre blocs. h2), h4) Le HMB extrait du bloc 2 et bloc 4 respectivement de la frame f2 dans l'image marquée (a).	67
Figure 4.1. Exemples de composantes connexes de la base IBN SINA.....	70
Figure 4.2 La Page 0008 de la base IBN SINA.	70
Figure 4.3. Exemples de caractères de la base de données IFHCDB.	71
Figure 4.4. Exemple d'image divisée en 4 blocs.	71
Figure 4.5. Les quatre types d'attributs de concavités.....	74
Figure 4.6 Topologie des caractères HMMs utilisé.....	75

Liste des tableaux

Tableau 1.1 L'alphabet arabe et ses différentes formes dans le mot	18
Tableau 4.1. Comparaison des résultats de l'attribut proposé à ceux des méthodes implémentées en utilisant la base de données IBN SINA et la base de données IFHCDB.	77
Tableau 4.2. Les résultats de l'attribut proposé dans cette étude en utilisant la base de données IFHCDB et des caractères isolés de la base de données IBN SINA.....	78
Tableau 4.3. Les taux de reconnaissance avec l'attribut proposé avec un rayon de 4 pixels et différentes combinaisons de frames et chevauchement.	79
Tableau 4.4. Taux reconnaissance de l'attribut proposé avec différentes valeurs de rayon et différentes configurations de directions en utilisant la base de données IBN SINA.	80
Tableau 4.5. Taux reconnaissance de l'attribut proposé avec différentes valeurs de rayon et différentes configurations de directions en utilisant la base de données IFHCDB.	80

Liste des Abréviations

C

CNN	
Convolutional Neural Networks ·	31
CTC,	
Connectionist Temporal Classification ·	36

H

HMB	
Histogram of Marked Background ·	11
HMMs	
Hidden Markov Models ·	12
HOG	
Histogram of Oriented Gradients ·	42
HTK	
Hidden markov model ToolKit ·	74

K

K-NN	
K-Nearest Neighbor ·	32

L

LBP	
Local Binary Patterns ·	43
LSTM	
Long Short Term Memory ·	36

O

OCR	
Optical Character Recognition ·	10

Liste des abréviations

P

PAW

Part of Arabic Word ·

49

R

RNN

Recurrent Neural Networks. ·

36

S

SIFT

Scale Invariant Feature Transform ·

45

SVM

Support Vector Machine ·

32

Introduction Générale

L'écriture qu'elle soit imprimée ou manuscrite est le moyen le plus utilisé pour la communication et la transmission d'informations entre les individus ainsi que la conservation d'informations et du savoir dans les archives. L'automatisation d'accès et du traitement des documents manuscrits archivés ou en cours d'utilisation s'impose. En effet avec l'arrivés des nouvelles technologies de l'information et les avancées technologiques dans les domaines d'informatique, d'électronique et d'automatique, les chercheurs du domaine de la reconnaissance de forme notamment de la reconnaissance d'écriture se préoccupent du développement d'application visant la reconnaissance automatique de l'écriture (OCR).

La reconnaissance automatique de l'écriture (OCR) est un processus informatique complexe, qui vise à traduire un texte imprimé ou manuscrit en un texte codé numériquement. La reconnaissance de l'écriture concerne toutes les tâches en relation avec le traitement de masse des documents papiers. Par conséquent, elle porte sur les applications répétitives volumineuses avec des bases de données de taille importantes à savoir : le traitement automatique des dossiers administratifs, le tri automatique du courrier postal, la lecture des montants et des chèques bancaires, le traitement des adresses postales, le traitement des formulaires, les interfaces sans clavier, l'analyse du geste écrit, la lecture des documents patrimoniaux, l'indexation d'archives bibliothécaires et la recherche d'information dans de bases de documents et etc.

Malgré les travaux réalisés dans le domaine de la reconnaissance de l'écriture manuscrite, en particulier l'écriture arabe manuscrite, il reste toujours l'un des problèmes les plus difficiles dans le domaine de reconnaissance de forme. Cette difficulté est liée à la complexité morphologique de l'écriture arabe (cursivité, contient des loupes, point diacritique, même lettre peut avoir plusieurs formes, etc.) et aux grandes variations des styles d'écriture ainsi que la dégradation des documents manuscrits.

La reconnaissance automatique de l'écriture passe par plusieurs étapes : acquisition, prétraitement, segmentation, extractions d'attributs, classification (reconnaissance) et posttraitement. La reconnaissance peut être globale ou analytique. Dans la reconnaissance analytique, l'image du mot ou du caractère est segmenté en sous unités avant de passer à la phase d'extraction d'attribut contrairement à la reconnaissance holistique où les attributs sont extraits sans passer par l'étape de segmentation.

Ainsi, le choix d'une méthode d'extraction de caractéristiques appropriée reste une étape cruciale et délicate pour obtenir une grande précision de reconnaissance de l'écriture arabe, en particulier pour les documents historiques.

Les attributs de classification utilisés dans la littérature peuvent être classés en deux groupes principaux : les attributs structurels et les attributs statistiques. Lors de l'analyse de ces méthodes d'extraction de caractéristiques, nous avons constaté qu'elles se concentrent sur l'analyse des pixels noirs (encre) dans les images de textes. Alors qu'une image de texte est un ensemble de pixels d'encres ayant une certaine distribution spatiale dans le fond de l'image. Ceci dit que le fond de l'image est important et ne doit pas être négligé lors de l'extraction des caractéristiques. D'où l'idée de proposer un nouvel attribut qui prend en compte les pixels de fond en plus des pixels d'encre dans l'image du texte pour la reconnaissance de l'écriture arabe manuscrite.

L'objectif principal de cette thèse est de développer une technique pour la reconnaissance de l'écriture arabe manuscrite de manière efficace. Afin d'atteindre l'objectif principal de cette recherche :

- Une nouvelle méthode d'extraction de caractéristique efficace qui prend en considération les spécificités de la langue arabe est proposée.
- Mapper les caractéristiques extraites dans un classifieur pour mesurer la performance du système de reconnaissance.

Dans cette thèse, nous proposons, un nouvel attribut (pour la reconnaissance de l'écriture arabe manuscrite) appelé HMB (Histogram of Marked Background). La nouveauté avec notre contribution (HMB) est que, en plus des pixels d'encre nous utilisons également le fond de l'image du texte pour extraire les caractéristiques. Le principe générale de l'attribut HMB est d'injecter des

informations discriminantes dans l'image du texte et cela en attribuant des marques aux pixels du fond des images. Les pixels de fond sont marqués selon la répartition spatiale des pixels d'encre dans leur voisinage. Par conséquent, ces marques permettent de capturer la forme et l'orientation de l'écriture (lignes verticales, lignes horizontales, boucles, points, concavités, et autres). L'attribut HMB permet ainsi de capturer les caractéristiques complexes de l'écriture arabe manuscrite.

Dans le but de faire la reconnaissance de mots à vocabulaire large et afin de pallier aux problèmes liés à la segmentation nous avons utilisé les HMMs comme outil de reconnaissance. Les HMMs ont pour avantage la faculté de faire la reconnaissance sans faire recours à la segmentation ainsi que la capacité de faire la reconnaissance de mots en faisant la concaténation des modèles de caractères.

Le reste de la thèse est organisé comme suit :

- **Chapitre 1** : dans ce chapitre nous avons décrit l'écriture arabe et ses caractéristiques, les approches de reconnaissances (holistique et analytiques), les étapes de reconnaissance de l'écriture dans lesquelles nous avons présenté les méthodes de prétraitements, les méthodes de segmentations, les méthodes d'extraction d'attributs, les méthodes de classifications et les méthodes de post traitements, enfin, nous avons présenté les problèmes rencontrés dans la reconnaissance de l'écriture arabe manuscrite,
- **Chapitre 2** : dans ce chapitre nous avons présenté un état de l'art sur les techniques d'extraction des attributs de reconnaissance appliqués à la reconnaissance de l'écriture arabe manuscrite.
- **Chapitre 3** : dans ce chapitre nous avons présenté notre contribution à la reconnaissance de l'écriture arabe qui est la proposition d'un nouvel attribut appelé HMB. Nous avons présenté les étapes suivies pour l'extractions des attributs HMB à savoir le processus de marquage des images ainsi que l'algorithme utilisé pour le marquage, et l'étape de calcul des histogrammes HMB ainsi que les vecteurs de caractéristiques.

- **Chapitre 4** : dans ce chapitre nous avons présenté les bases de données utilisées pour le test de notre attribut (HMB), les méthodes d'extractions d'attributs implémentées utilisées pour la comparaison de notre méthode, la méthode de reconnaissance (HMM) utilisée pour l'évaluation de notre méthode, les différents résultats obtenus pour différentes valeurs des paramètres du système ainsi que la discussion des résultats obtenus avec notre méthode et les méthodes implémentées.
- **Conclusion générale** : dans laquelle nous avons présenté une synthèse du travail réalisé dans cette thèse ainsi que les perspectives et les travaux futurs envisagés.

Reconnaissance de l'écriture arabe manuscrite.

Toute information écrite peut être reprise et informatisée à différentes fins et dans plusieurs domaines (bureautique, reconnaissance de montants littéraux de chèques bancaires, tri du courrier dans les postes etc.), d'où on trouve la reconnaissance optique des caractères (OCR) qui dérive du domaine de la reconnaissance de forme et qui occupe une place importante dans la recherche scientifique. Contrairement au latin la reconnaissance de l'écriture arabe imprimée ou manuscrite reste encore aujourd'hui un domaine non complètement exploré. A. nazif [1] fut le premier à travailler sur la reconnaissance de l'écriture arabe dans sa thèse de master en 1975.

1.1. Généralité sur le traitement d'images

Le traitement d'images est un domaine très vaste. On désigne par traitement d'images numériques l'ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations. De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes.

1.1.1. Définition d'une image

Une image peut être définie comme une fonction bidimensionnelle, $f(x,y)$, où x et y sont des coordonnées spatiales, et l'amplitude de f à n'importe quelle paire de coordonnées (x, y) est appelée l'intensité ou le niveau de gris de l'image à ce point. Lorsque les valeurs de l'amplitude de f et des coordonnées (x,y) sont

des quantités discrètes, l'image est appelée image numérique. Une image numérique est donc composée d'un nombre fini d'éléments, ayant chacun un emplacement et une valeur donnés. Ces éléments sont appelés pixels.[2]

1.1.2. Représentation d'une image numérique

Une image est une fonction d'au moins deux variables réelles :

Image : $f(x, y)$	image 2D
Volume : $f(x, y, z)$	image 3D
Séquence d'image : $f(x, y, t)$	image 3D
Séquence de volumes : $f(x, y, z, t)$	image 4D

Les valeurs prises par $f(.)$ peuvent être:

- Scalaires (intensité lumineuse)
- Vectorielles (couleur (RVB),...), imagerie multi spectrale, image de paramètres...)
- Réelles ou complexes

1.1.3. Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants[3] :

1.1.3.1. Dimension :

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image[3].

1.1.3.2. Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur ; plus grand est ce nombre, meilleure est la résolution [3].

1.1.3.3. Bruit

Le bruit dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur [3].

1.1.3.4. Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet [3].

1.1.3.5. Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images [3].

1.1.3.6. Images à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet[3].

1.1.3.7. Images en couleurs

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans

d'ordinateurs,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.)[3].

1.1.4. Les traitements liés à l'image numérique

Plusieurs traitements peuvent être appliqués à une image après son acquisition. Parmi ces traitements on cite l'amélioration qui permet de mettre en relief les détails ou faire ressortir certaines caractéristiques. La restauration d'images qui permet d'améliorer les images ayant subies des dégradations. La compression qui vise à réduire la quantité d'informations véhiculées par une image. Les traitements morphologiques qui sont un ensemble d'outils pour extraire des composantes de l'image. La Segmentation qui est une procédure de partitionnement de l'image en ses composantes ou objets. La Représentation et la description, elles interviennent généralement après une segmentation. La conversion des résultats obtenus sous une forme convenable pour la suite du traitement. La description peut être vue comme une sélection de caractéristiques pour la classification et la reconnaissance de formes.

1.1.5. Applications de traitements d'images

Le traitement d'images est utilisé dans plusieurs domaines tels que la robotique en industrie (assemblage, reconnaissance de pièces, véhicule autonome et etc.). La télédétection (météo, cartographie, analyse des ressources terrestres, astronomie et etc.). Les applications militaires (guidage de missile, reconnaissance (aérienne, sous-marine et etc.). L'imagerie médicale (tomographie, aide au diagnostic, suivi de formes anatomiques et etc.). La sécurité (reconnaissance (d'empreintes, visages, signatures), détection de mouvement et etc.). Dans les administrations pour la reconnaissance de l'écriture (les adresses de courriers postaux, les montants de chèques bancaires, les documents d'archives et etc.)

1.2. Description et caractéristiques de l'écriture arabe

La langue arabe est une langue universelle et c'est la langue officielle de plus de 20 pays. L'écriture arabe s'écrit horizontalement de droite et à gauche. Avec l'expansion de l'islam, différentes langues ont adopté le système d'écriture de l'arabe, langue du Coran, et l'usage de l'alphabet arabe s'est progressivement répandu. C'est ainsi que cet alphabet sert également à écrire certaines langues

indo-européennes comme le persan, le kashmiri, le sindhi, l'ourdou, le kurde le farsi, le jawi ect.

L'alphabet arabe ainsi que ses différentes formes dans le mot sont décrit dans le tableau 1.1.

No	Name	Isolated	Connected		
			Beginning	Middle	End
1	Alif	ا	ا	ا - ا	ا
2	Baa	ب	ب	ب	ب
3	Taa	ت	ت	ت	ت
4	Thaa	ث	ث	ث	ث
5	Jeem	ج	ج	ج	ج
6	Haa	ح	ح	ح	ح
7	Khaa	خ	خ	خ	خ
8	Daal	د	د	د	د
9	Thaal	ذ	ذ	ذ	ذ
10	Raa	ر	ر	ر	ر
11	Zaay	ز	ز	ز	ز
12	Seen	س	س	س	س
13	Sheen	ش	ش	ش	ش
14	Saad	ص	ص	ص	ص
15	Shaad	ض	ض	ض	ض
16	Ttaa	ط	ط	ط	ط
17	Dthaa	ظ	ظ	ظ	ظ
18	Ain	ع	ع	ع	ع
19	Ghen	غ	غ	غ	غ
20	Faa	ف	ف	ف	ف
21	Qaf	ق	ق	ق	ق
22	Kaf	ك	ك	ك	ك
23	Lam	ل	ل	ل	ل
24	Mem	م	م	م	م
25	Noon	ن	ن	ن	ن
26	Haa	ه	ه	ه	ه
27	Wow	و	و	و	و
28	Yaa	ي	ي	ي	ي

Tableau 1.1 L'alphabet arabe et ses différentes formes dans le mot

1.2.1. Caractéristiques de l'écriture arabe

Les caractéristiques de l'écriture arabe peuvent être résumées en plusieurs points comme illustré dans la figure 1.1.

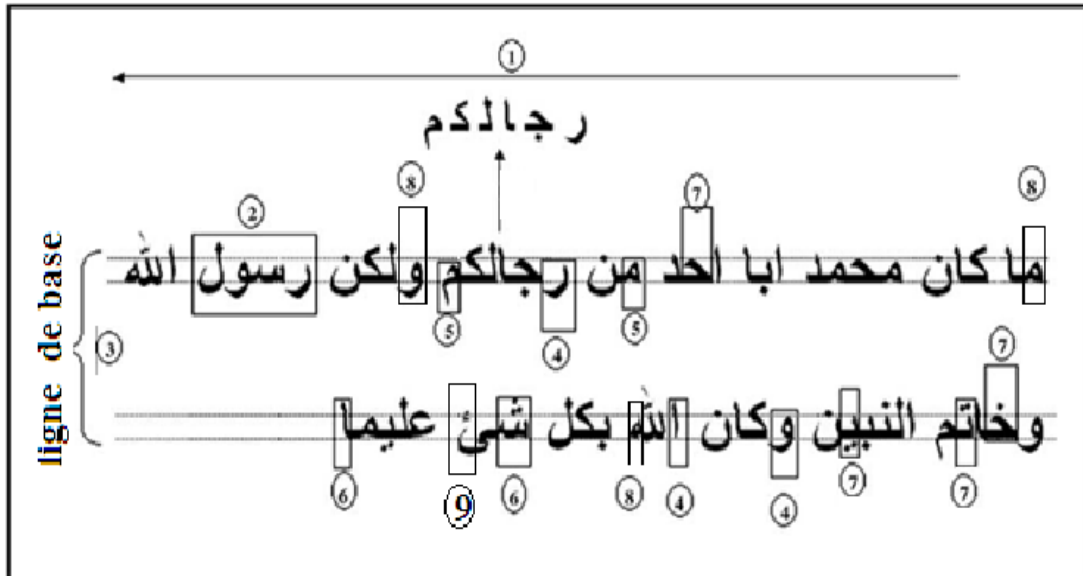


Figure 1.1. Caractéristiques de l'écriture arabe.

1. L'arabe s'écrit de droite à gauche.
2. Un mot peut être constitué de deux ou plusieurs sous mots.
3. Les mots sont écrits suivant une ligne imaginaire appelée ligne de base.
4. L'arabe est toujours écrit cursivement et les mots sont séparés par des espaces, sauf six caractères qui ne peuvent être connectés qu'à droite [ر، ز، و، ا، د، ذ] lorsque ils apparaissent dans le mot il sera coupé en deux ou plusieurs sous mots séparés par des petits espaces.
5. La forme du caractère change selon sa position dans le mot, chaque caractère a deux ou quatre formes tableau 1.
6. La taille des caractères diffère de l'un à l'autre et du même caractère selon sa position dans le mot.
7. Quinze caractères ont des points qui peuvent être au-dessous ou au-dessus de la partie principale. Quelques caractères partagent la même partie principale sauf les points peuvent faire la différence entre eux figure 1.2.

8. Quelques caractères contiennent des boucles, la boucle est une caractéristique importante pour décrire le mot, le caractère [ه] contient deux boucles.
9. Le caractère Hamza [ء], n'est pas vraiment une lettre c'est une forme complémentaire qui est utilisée avec la lettre [ك] dans la forme isolée ou à la fin et elle peut être utilisée isolée ou avec les lettres [ا ، و ، ي].
10. L'écriture manuscrite diffère d'un scribe à un autre, cela signifie qu'il existe une variabilité dans l'écriture d'un même mot ou phrase par le même scribe ou deux scribes différents figure 1.3.
11. Elle utilise des diacritiques qui sont utiles des fois pour montrer le vrai sens d'un mot et sa prononciation figure 1.4.

ب ت ث ح ج خ د ذ س ش ص ض ف ق ه ة و ز ر ع غ

Figure 1.2. Quelques lettres partageant le même corps et différentes par les points diacritiques



Figure 1.3. Exemple de variabilité dans l'écriture arabe manuscrite

Diacritics Name			Shape	Examples	Diacritics Name			Shape	Examples
Fatha	فتحة	ـَ	مُنَجِّدٌ	Sukoon	سكون	ـْ	مَعْرُوفًا		
Thamma	ضممة	ـُ	تَشْكُرُونَ	Shadda	شدة	ـّ	عَطَا		
Kasra	كسرة	ـِ	أُولِيَاؤِكُمْ	Madda	مدة	ـً	أَبَاؤُكُمْ		
Tanween Al-Fath	تنوين الفتح	ـً	قَوْلًا	Alif- Khinjariah	الف خنجرية	ا	أُولَئِكَ ، إِلَى إِلَهِ		
Tanween Al-Thamm	تنوين الضم	ـً	كَبِيرٌ	Hamzatul- Wasl	همزة وصل	ـِ			
Tanween Al-Kasr	تنوين الكسر	ـٍ	كَأَخٍ						

Figure 1.4. Liste des diacritiques [4]

1.3. Le processus de reconnaissance de l'écriture arabe

La reconnaissance de l'écriture est une branche très importante du domaine de reconnaissance de formes. C'est un processus informatique complexe, qui vise à traduire un texte imprimé ou manuscrit en un texte codé numériquement.

1.3.1. Les Approches de reconnaissance :

La reconnaissance de l'écriture peut être faite sur la totalité du mot suivant l'approche holistique ou sur des parties du mot suivant l'approche analytique :

1.3.1.1. Approche globale (holistique) :

Cette approche se base sur une description unique de l'image du mot (vu comme une entité indivisible) et tente de le reconnaître en utilisant les caractéristiques du mot entier. Disposant de beaucoup d'informations (un vocabulaire large) rend la discrimination des mots proches très difficile et l'apprentissage des modèles nécessitant une grande quantité d'échantillons qui est souvent difficile à réunir pour l'apprentissage d'un classifieur holistique. Cette

approche reste parfaitement envisageable pour les vocabulaires réduits et distincts (exemple : reconnaissance de montants littéraux de chèques bancaires).

1.3.1.2. Approche analytique :

Cette approche traite le mot comme une collection de sous unités simples comme les caractères et procède en segmentant ces mots en unités. Cette approche est applicable dans le cas de grands vocabulaires. Elle présente cependant des problèmes, surtout dans la segmentation de l'écriture manuscrite tel que : l'ambigüité dans la détermination des points de segmentation et la détermination de l'identité de chaque segment à cause de la variabilité de la forme des segments.

1.3.2. Les étapes de reconnaissance de l'écriture :

La reconnaissance de l'écriture passe par plusieurs étapes à savoir : l'acquisition, le prétraitement, segmentation, extraction des attributs, la classification (reconnaissance) et le poste traitement. Le processus de reconnaissance peut passer directement de la phase de prétraitement à la phase d'extraction d'attribut sans passer par la phase de segmentation (approche globale). Contrairement à l'approche analytique où la phase de segmentation est primordiale avant de passer à la phase d'extraction d'attributs.

La figure 1.5 présente le schéma général d'un système de reconnaissance de l'écriture. Pour assurer une bonne reconnaissance, après l'acquisition des données, ces dernières peuvent passer par un ensemble d'opérations de prétraitement, puis une étape de segmentation peut avoir lieu. Dans le cas de reconnaissance en utilisant l'approche holistique le système passe directement à l'étape d'extraction de caractéristiques sans passer par la phase de segmentation. Les attributs extraient dans la phase d'extraction d'attributs seront utilisés dans l'étape de classification (apprentissage et test), puis à la fin une étape d'amélioration des résultats (posttraitement) pourra avoir lieu.

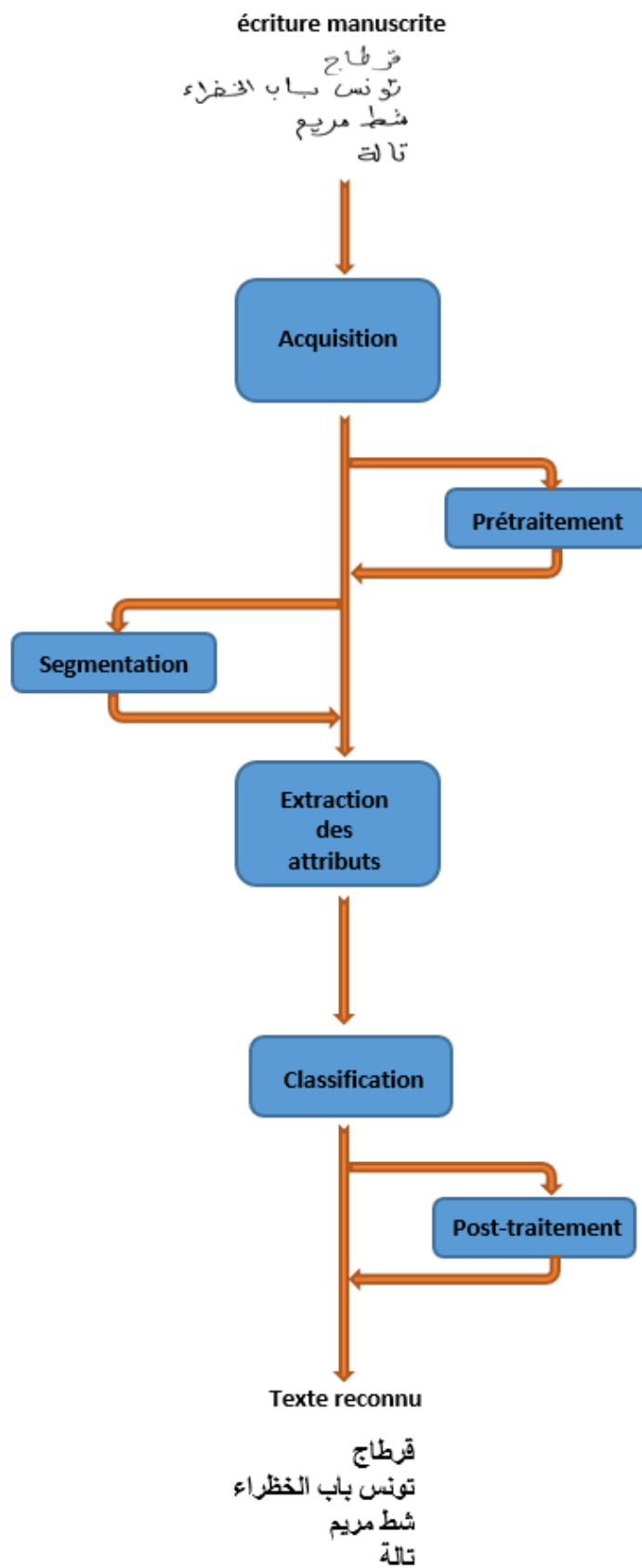


Figure 1.5. L'organisation générale d'un système de reconnaissance de l'écriture.

1.3.2.1. Acquisition

L'acquisition des données peut être faite par une tablette en cas de reconnaissance en ligne (temps réel) où la numérisation des scripts dépend de la résolution de la tablette et de la vitesse d'échantillonnage. Dans le cas de la reconnaissance hors ligne, l'acquisition est effectuée par un scanner où la qualité des données acquises dépend de la résolution de ce dernier.

1.3.2.2. Prétraitement

La qualité de reconnaissance dépend de la qualité du texte en entrée et la qualité du texte en entrée dépend de plusieurs facteurs tel que : la résolution du scanner utilisé, le type du document ancien ou récent (faxé ou copié), etc. Le prétraitement c'est l'ensemble d'opérations qui apportent des transformations à une image dans le but de réduire le bruit superposé aux données. Parmi les opérations de prétraitements généralement utilisées on trouve : la binarisation, le lissage, la squelettisation, le redressement de l'écriture, la normalisation et la détection de la ligne de base. La figure 1.6 présente quelques opérations de prétraitements réalisés sur une image.

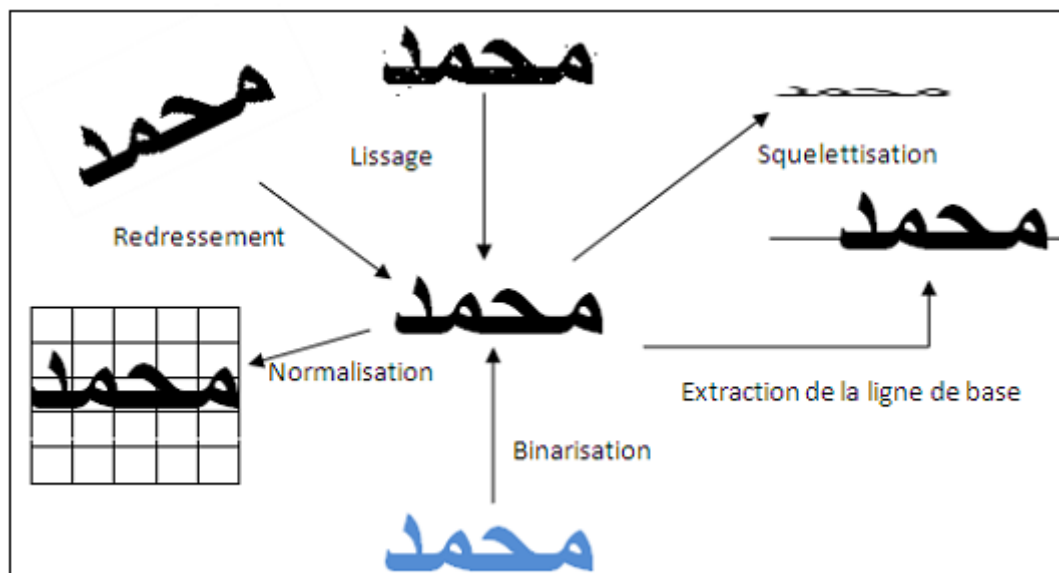


Figure 1.6. Opérations de prétraitements.

- **La binarisation** : c'est la transformation d'une image en couleur ou en niveaux de gris en une image binaire (composée de deux valeurs 0 et 1) qui permet de distinguer entre le fond d'une image (support papier) et la forme (traits des gravures et des caractères en noir). Il existe plusieurs méthodes de binarisations [5]. Parmi ces méthodes on trouve les méthodes basées sur le seuillage [6], [7] où les valeurs des niveaux de gris dans l'image qui sont supérieurs à un certain seuil sont mises à 1 et celles inférieurs à ce seuil sont mises à zéro. Le seuil de binarisation doit être choisi attentivement : s'il est très élevé les mots peuvent devenir très mince et peuvent se couper et s'il est très petit les mots peuvent devenir plus épais et des caractères isolés peuvent être connectés. On trouve aussi les méthodes basées les technique de traitement d'images [8 , 9], les méthodes basées sur les modèles statistiques [10] et les méthodes basées sur la classification des pixels [11].
- **Le lissage** : les techniques de lissage permettent de résoudre les problèmes dû aux artefacts de l'acquisition, tel que les bruits qui conduisent soit à une absence ou à une surcharge de points. Ceci en utilisant les opérations de bouchage et de nettoyage. Les opérations de bouchage égalisent les contours et bouchent les trous internes et celles de nettoyage permettent de supprimer les petites tâches et les excroissances de la forme [12]. Une méthode qui est souvent utilisée pour réduire le bruit dans une image 2D est de parcourir l'image pixel par pixel et changer la valeur du pixel suivant ses 8 voisins (nord, nord-ouest, est,...), un pixel qui est à "1" est mis à "0" si il n'y a pas assez de pixels à "1" dans son entourage pour le supporter et vice versa [13]. Plusieurs autres techniques similaires sont utilisées citant les méthodes basées sur la morphologie mathématique [14] .
- **Squelettisation** : la squelettisation (amincissement) est une technique qui permet de réduire l'épaisseur des caractères dans le but de simplifier la représentation des caractères et réduire le temps de traitement. Parmi les algorithmes de squelettisation on trouve dans [15] l'algorithme qui supprime itérativement les points des frontières de la forme jusqu'à obtention d'un squelette d'épaisseur d'un pixel. N.J. Naccache et al dans [16] présentent une comparaison

de performance de 14 algorithmes de squelettisation. L. Lam, S et al dans [17] présentent un état de l'art sur les méthodologies de squelettisation. Parmi les inconvénients des algorithmes de squelettisation, l'altération de la forme des caractères tels que la réduction des points diacritiques doubles et uniques à une même forme.

- **Extraction de la ligne de base** : la ligne de base contient l'information sur l'orientation du texte et les positions des points de connexion entre les caractères. La méthode la plus utilisée pour trouver la ligne de base est la projection horizontale, qui est un vecteur de taille égale au nombre de lignes de l'image, chaque entrée du vecteur contient le nombre de pixels à "1" dans sa ligne correspondante. La ligne de base apparaît dans l'histogramme comme l'ensemble des entrées consécutives à valeurs maximales. Quelques systèmes OCR utilisent la ligne de base pour l'alignement des pages penchées, la séparation des lignes dans un bloc de texte et la segmentation des mots en caractères. A. AL-Shatnawi et al dans [18] présentent une étude et une comparaison des méthodes de détection de la ligne de base et les difficultés rencontrées en traitant l'écriture manuscrite arabe.
- **Normalisation** : la normalisation de la taille des images des textes consiste à définir dans des matrices de même taille les images de ces derniers, dans le but de faciliter les traitements ultérieurs. Cette opération introduit généralement de légères déformations sur les images, certains traits caractéristiques tels que la hampe dans les caractères peuvent être éliminés à la suite de la normalisation, ce qui peut entraîner des confusions entre certains caractères [19].
- **Redressement** : l'un des problèmes rencontrés en OCR est l'inclinaison des lignes du texte, qui introduit des difficultés pour la segmentation. L'inclinaison peut être intrinsèque au texte ou provenir de la saisie, si le document a été placé en biais. [19].

1.3.2.3. Segmentation

Dans cette phase les différentes parties logiques de l'image sont séparées en blocs graphiques et blocs de textes, puis à partir des blocs de texte sont extraites les lignes, puis les mots, ensuite les caractères ou même des unités plus simples figure 1.7. C'est l'une des phases les plus critiques, difficiles et qui prend plus de temps de traitement surtout pour l'écriture cursive comme l'arabe.

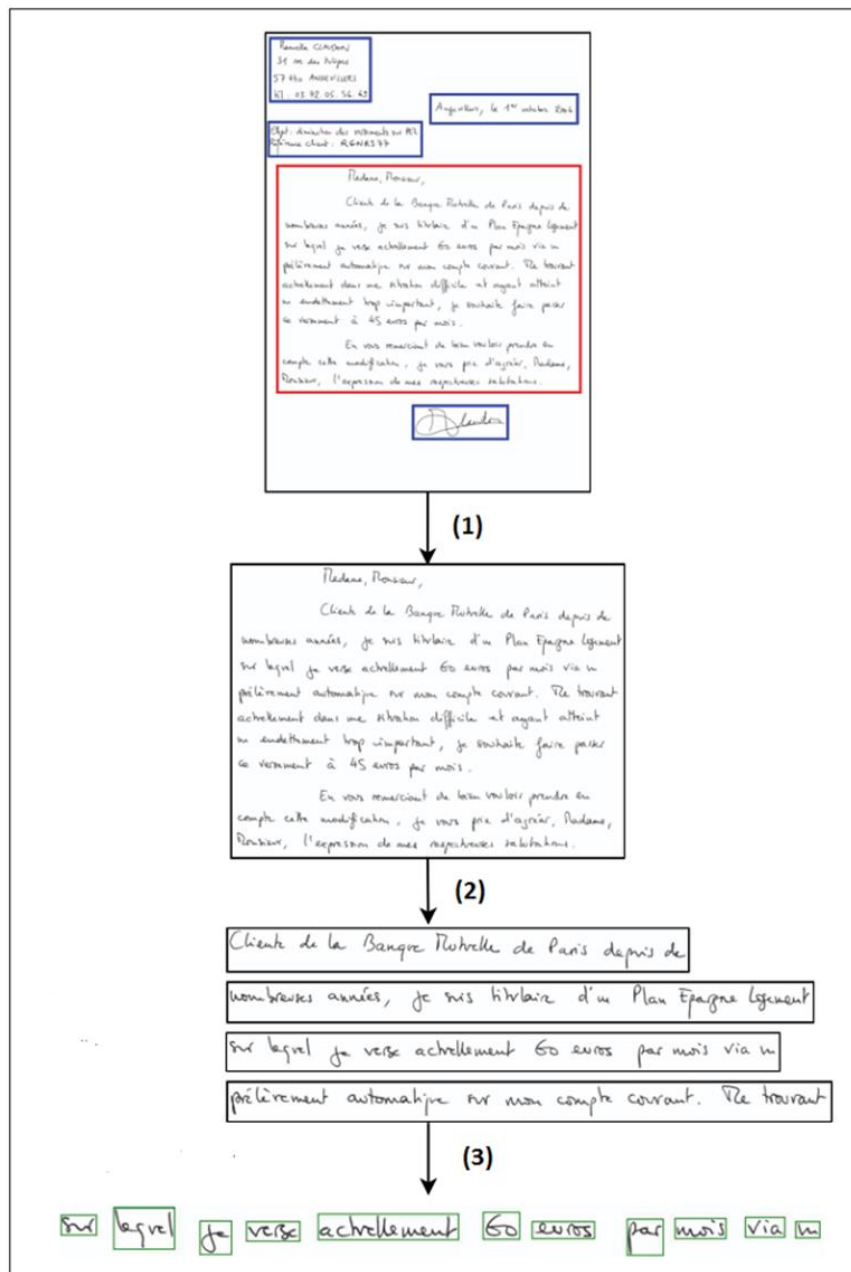


Figure 1.7. Les différentes étapes appliquées pour passer d'une image d'un document à un ensemble d'images de mots. 1) Segmentation en zones ; 2) Segmentation en lignes ; 3) Segmentation en mots [20].

La segmentation peut être implicite ou explicite. Dans la segmentation implicite les caractères sont segmentés lors de la reconnaissance, des segments de textes et de points de segmentation provisoires seront confirmés ou non par la classification. L'avantage principal de cette technique est qu'elle évite les problèmes rencontrés dans la séparation des caractères et n'engendre pas d'erreur de segmentation. Y.M. Alginahi dans [21] a fait un état de l'art sur les méthodes de segmentation de l'écriture arabe. Nous présentons quelques méthodes de segmentation explicite à savoir :

1.3.2.3.1. Les méthodes basées sur les histogrammes de projection et la ligne de base

Le But de la projection est de simplifier le système de reconnaissance par réduction de l'information de deux dimensions à une dimension. La projection horizontale permet la séparation des lignes de texte et détection de la ligne de base (la plus grande pique de la figure 1.8.b représente la ligne de base) alors que la projection verticale aide dans la segmentation du mot en sous mots et caractères. Les figures 1.8.b et 1.8.c représentent respectivement la projection horizontale et verticale de la phrase de la figure 1.8.a. Les travaux dans [12, 13] utilisant les histogrammes de projection.

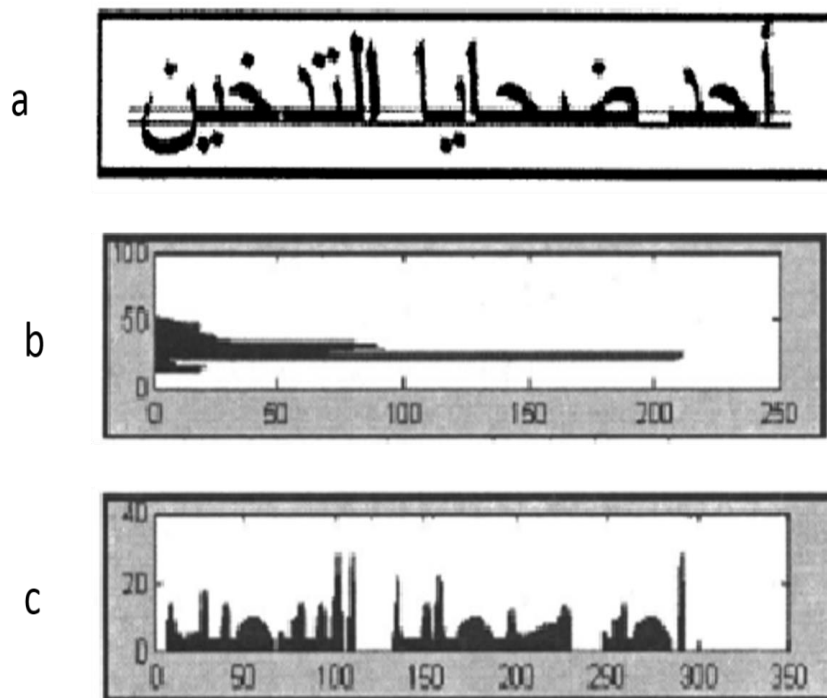


Figure 1.8. Segmentation par projection. b) horizontale. c) verticale [22]

1.3.2.3.2. Les méthodes basées sur le tracé du contour

Ces méthodes analysent la forme de l'écriture et se basent sur l'analyse du tracé du contour (supérieur et inférieur) des mots pour trouver des points de segmentations. Dans la figure 1.9.b des points de segmentation sont détectés sur le contour supérieur du mot dans la figure 1.9.a.

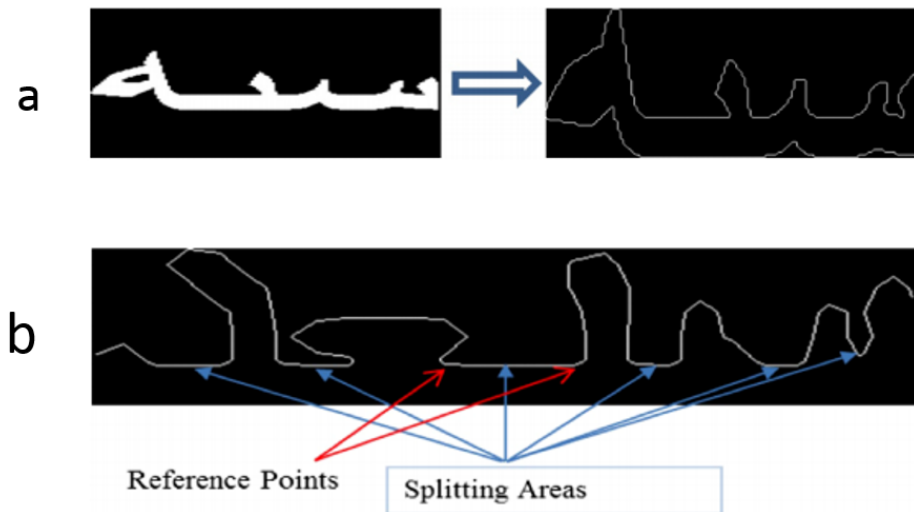


Figure 1.9. Segmentation basé sur le tracé du contour [23]

Les travaux dans [24],[25] utilisent le contour pour la segmentation des mots en caractères.

1.3.2.3.3. Les méthodes basées sur la squelettisation

Le squelette d'un mot fournit ses informations essentielles. Pour cela plusieurs auteurs se sont servis de l'analyse du squelette de mots pour faire la segmentation en caractères [26, 27]. La figure 1.10 montre des points de segmentation détectés sur le squelette d'un mot.

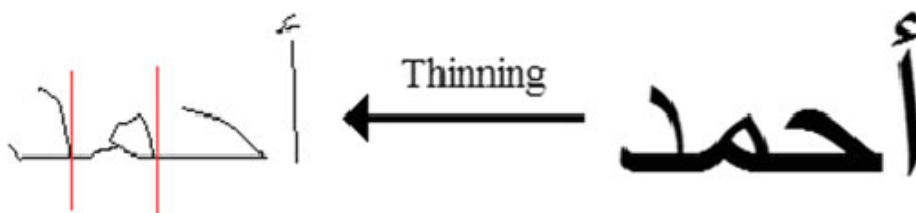


Figure 1.10. Segmentation basé sur la squelettisation [21]

1.3.2.3.4. Les méthodes basées sur les réseaux de neurones

Ces méthodes utilisent les réseaux de neurones dans la segmentation. Les auteurs dans [28] extraits des points de segmentation en utilisant des heuristiques. Ces points de segmentation sont passés à un réseau de neurones qui est préalablement entraîné avec des points de segmentation extraits manuellement pour la vérification et la validation.

1.3.2.4. Extraction d'attributs

Cette phase est une étape importante et cruciale dans le processus de la reconnaissance de l'écriture. Cette phase analyse les segments de textes et extrait un ensemble de caractéristiques qui servent à son identification. Elle consiste à représenter les données d'entrées (texte, mots, caractères, etc.) sous la forme d'un vecteur de primitives, où chaque vecteur se comporte comme une signature unique de la donnée en entrée. Un bon attribut doit être caractérisé par le pouvoir discriminant c.-à-d. minimise la variabilité intra classe et maximise la variabilité interclasse. Autrement dit un bon attribut dans lequel les valeurs du vecteur de primitives sont similaires pour les données appartenant à la même classe et distincts pour les données appartenant à différentes classes. Le choix d'un bon attribut dépend de la nature du texte (cursif,), les types des scripts qui peuvent être manuscrits ou imprimés et la langue de l'écriture (arabe, latin, etc.).

L'extraction d'attributs à partir d'images peut être effectuée de deux manières comme indiqué dans la figure 1.11. Premièrement, la manière la plus classique consiste à concevoir des attributs spécialement adaptées à la tâche et parfois même adaptées à un ensemble de données spécifique (écriture, formes, etc.). Ces attributs sont appelées manuels (handcrafted) puisqu'un algorithme a été conçu manuellement pour les extraire, incorporant des informations à priori sur les spécificités des données. Certains des attributs handcrafted sont très simples, tandis que certains ensembles d'attributs plus récents sont très complexes et généralement très dimensionnels [20]. Selon [29] ces méthodes sont classées en méthodes statistiques tel que le LBP [30], SIFT[31], basé sur les lignes droites[32], le gradient[33] et en méthodes structurels [34], [35].

La deuxième catégorie d'attributs consiste à apprendre automatiquement les attributs à partir des images en utilisant l'apprentissage automatique. C'est

une solution qui est de plus en plus utilisée depuis l'avènement du Deep Learning avec les Réseau neuronal convolutif (CNN) [36-38].

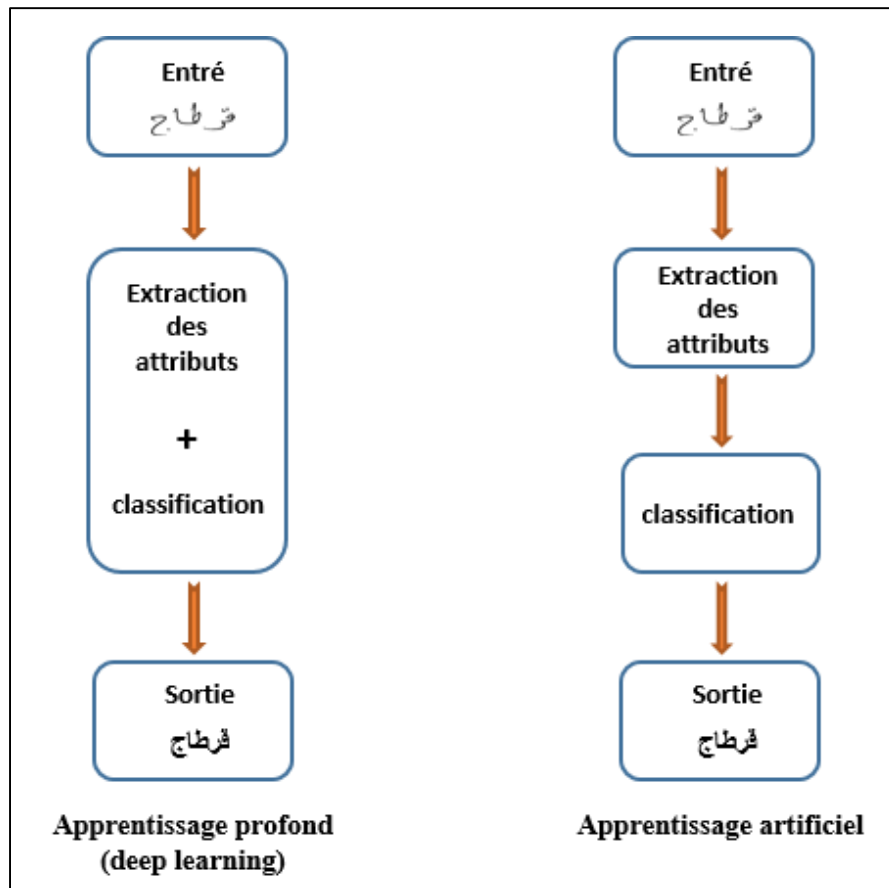


Figure 1.11. Extraction manuelles et automatique des attributs dans un système de reconnaissance.

1.3.2.5. Classification (reconnaissance) :

C'est la phase de prise de décision dans un système de reconnaissance de l'écriture, elle utilise les caractéristiques extraites dans la phase d'extraction d'attributs pour identifier des segments de texte, des mots ou des caractères suivant certaines règles. Cette phase peut utiliser des modèles de références obtenus lors d'une phase d'apprentissage pour classer les données.

Plusieurs méthodes de reconnaissance ont été reportées dans la littérature [39]. Les premiers travaux étaient limités à la reconnaissance d'images de caractères et de chiffres [1, 40]. Par la suite, les modèles de reconnaissance se sont améliorés pour reconnaître des mots entiers ou des lignes [19, 41, 42]. Parmi les méthodes de reconnaissance de l'écriture on trouve les K-plus proches voisins, les

machines à vecteurs de support, les Modèles de Markov cachés, les réseaux de neurones récurrents et etc.

1.3.2.5.1. Les K-plus proches voisins (KNN) :

La méthode des KNN est une technique de classification utilisant des vecteurs dans un espace de caractéristiques multidimensionnel, chacun avec une étiquette de classe en tant qu'échantillons d'apprentissage qui sont stockés lors de la phase d'apprentissage. Dans la phase de classification, les distances (la distance euclidienne est plus populaire) entre chaque échantillon d'apprentissage et l'échantillon de teste sont calculées. K est une constante définie par l'utilisateur. Les K échantillons d'apprentissage qui ont les plus petites distances (proches) par rapport à l'échantillon de test sont trouvés et leurs étiquettes sont identifiées. En utilisant le vote majoritaire sur les échantillons voisins, la classe de l'échantillon de test sera indiquée. La figure 1.12 montre une représentation schématique du principe des K plus proche voisin. Parmi les travaux utilisant les KNN pour la reconnaissance de l'écriture arabe manuscrite on cite [43, 44].

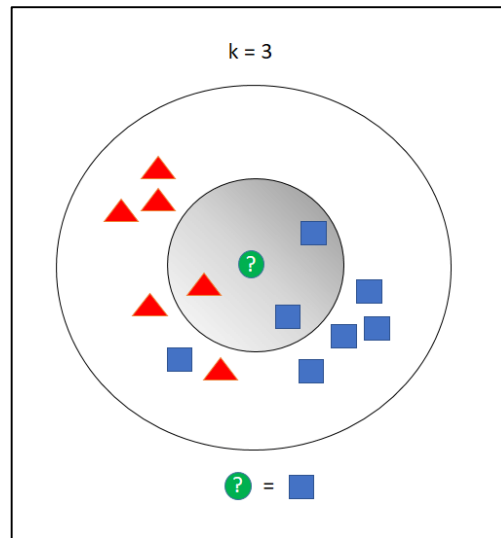


Figure 1.12. Représentations schématique du principe de K-NN

1.3.2.5.2. Les machines à vecteurs de support (SVM) :

Développé par Vapnik [45] et Cortes [46] les SVM sont utilisés comme outil pour la résolution des problèmes de classification et de régression. Leur principe générale est la séparation optimale des données en classes à l'aide d'une frontière (séparateur) de telle sorte à maximiser la distance qui sépare entre les différents

groupes de données et la frontière (voir figure 1.13) . Les SVMs reposent souvent sur l'utilisation de noyaux sigmoïdes ou polynomiaux lorsque les données sont linéairement inséparables. La figure 1.13 montre une représentation schématique du principe des SVM. Plusieurs travaux ont utilisé les SVM pour la reconnaissance de l'écriture arabe manuscrite [47, 48].

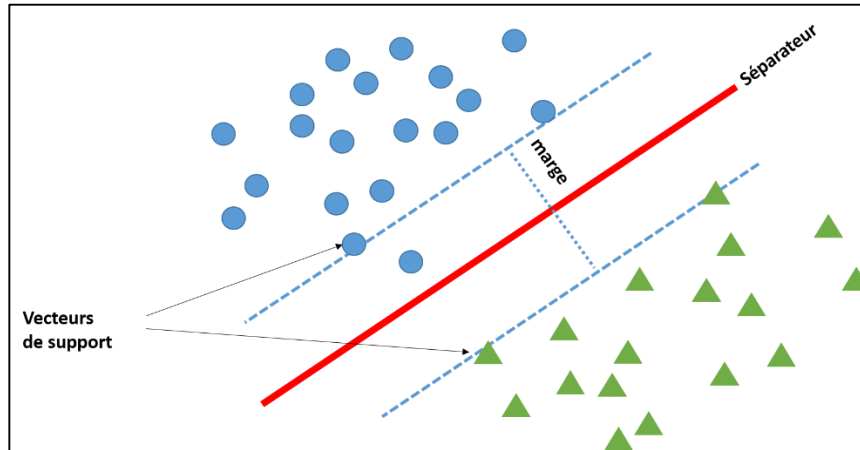


Figure 1.13. Représentation schématique d'un SVM

1.3.2.5.3. Les modèles de Markov cachés :

Un modèle de Markov caché (Hidden Markov Model ou HMM) décrit un processus stochastique composé d'une séquence d'états cachés qui sont observés indirectement à travers un autre processus [49]. La figure 1.14 présentent certaines topologies des HMMs. Figure 1.15 montre une représentation schématique des structures HMM pour la reconnaissance de mot isolé (voir figure1.15(a)) et de mot connecté (voir figure1.15(b)). Les états du modèle sont représentés par des cercles et les états non émetteurs par des carrés

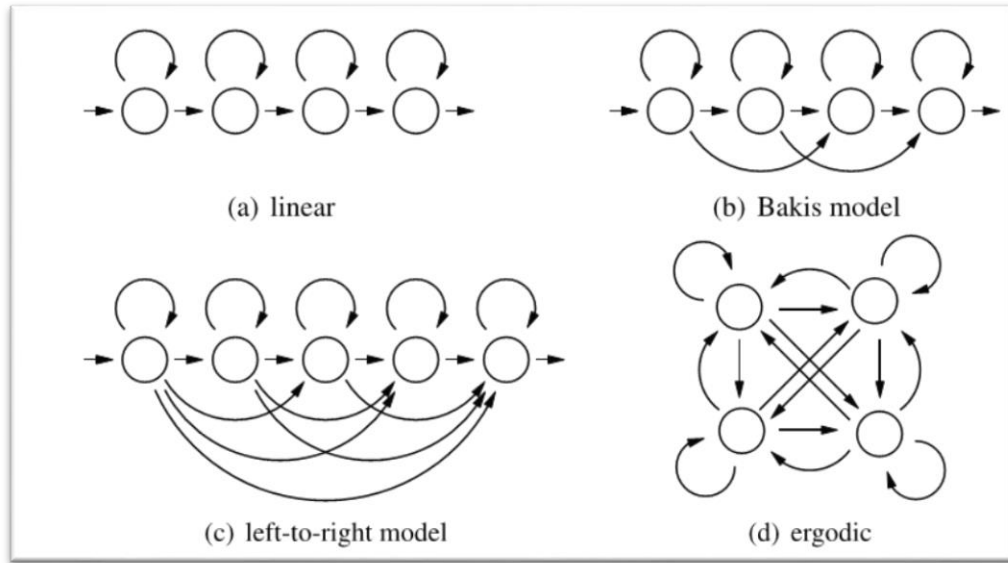


Figure 1.14. Représentation schématique des topologies HMMs [49]

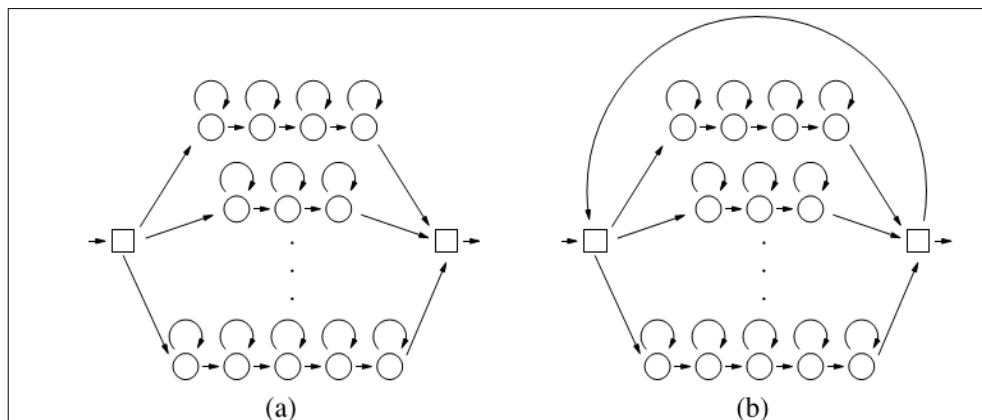


Figure 1.15. Représentation schématique des structures HMM pour la reconnaissance de (a) mot isolé et (b) mot connecté [49].

L'état du modèle à un instant t ne dépend que de l'état à l'instant $t-1$. A chaque instant, des observations sont émises selon une fonction de densité de probabilité dépendant de l'état courant. Formellement, un modèle de Markov caché de K états $\{S_1, \dots, S_K\}$ est défini par trois mesures de probabilités $\lambda = (\pi, A, B)$ tel que :

- π : les probabilités initiales, où π_i est la probabilité que S_i soit l'état initial.
- A : la matrice de transition d'états, où « a_{ij} » est la probabilité de transition de l'état S_i à l'état S_j .

– B : la probabilité d'émission, où $b_i(o)$ est la probabilité d'émettre l'observation o étant dans l'état S_i .

Une fois les paramètres $\lambda = (\pi, A, B)$ appris, le modèle peut par la suite être utilisé pour calculer la probabilité $P(O | \lambda)$ d'une séquence donnée d'observations $O = \{o_1, \dots, o_T\}$.

Pour la reconnaissance de mots, un modèle de Markov caché est défini pour chaque caractère de l'alphabet. Ces modèles sont ensuite combinés pour produire des modèles de mots. Étant donné l'image d'un mot représentée sous la forme d'une séquence d'observations (figure 1.16), le mot reconnu correspond au modèle donnant la plus grande probabilité pour la séquence [20].

La reconnaissance à l'aide de modèles de Markov cachés a l'avantage de reconnaître des mots sans avoir recours à la segmentation, et il présente aussi l'avantage de faire la reconnaissance d'un grand vocabulaire vu qu'il peut reconnaître des mots en se servant des modèles de caractères. Plusieurs travaux traitant la reconnaissance de l'écriture arabe manuscrite en utilisant les HMMs ont été fait parmi lesquels on cite [42, 50-55]

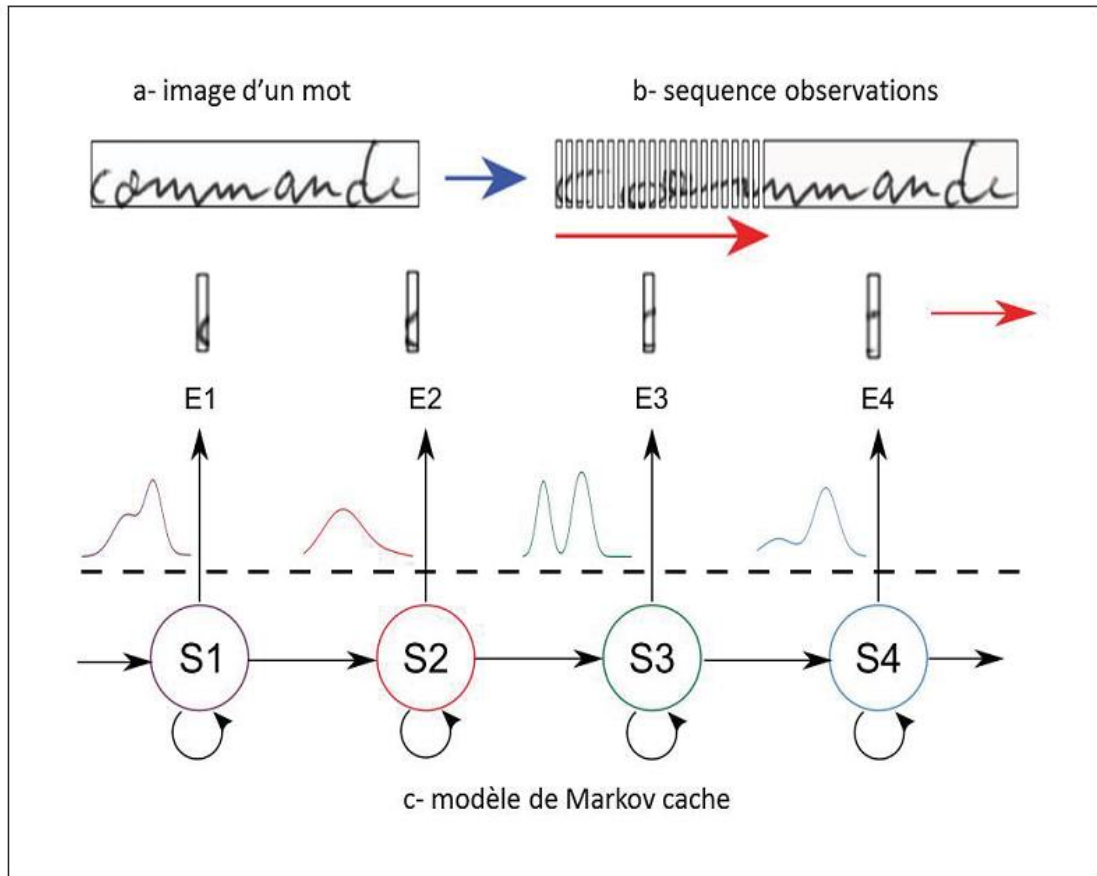


Figure 1.16 La reconnaissance avec les modèles de Markov cachés. a) L'image d'un mot ; b) La séquence d'observations ; c) Un modèle de Markov caché utilisé pour calculer la probabilité d'un mot par rapport à la séquence d'observations [20].

1.3.2.5.4. Les réseaux de neurones récurrents (RNN)

Les RNN sont des réseaux de neurones adaptés pour traiter les séquences de données grâce à des connexions récurrentes. Les réseaux récurrents avec mémoires à court et long termes (Long Short-Term Memory ou LSTM) sont une approche populaire pour la reconnaissance de l'écriture. Grâce à leur architecture, ces modèles sont particulièrement efficaces pour préserver l'information contextuelle dans les séquences. Pour entraîner un réseau LSTM à étiqueter une séquence d'entrée sans segmentation explicite de ses caractères, la méthode de classification temporelle connexionniste (CTC)[56] est souvent utilisée. Cette méthode repose sur une procédure avant-arrière (forward backward) pour transformer une séquence S en une séquence d'étiquettes L qui représente le texte reconnu. Un réseau LSTM produit une séquence S contenant différentes étiquettes parmi les caractères d'un alphabet [20]. La figure 1.17 montre un exemple de reconnaissance de mots avec les RNN. Plusieurs travaux traitant la

reconnaissance de l'écriture arabe manuscrite ont été reportés dans la littérature parmi lesquels on cite [57-59].

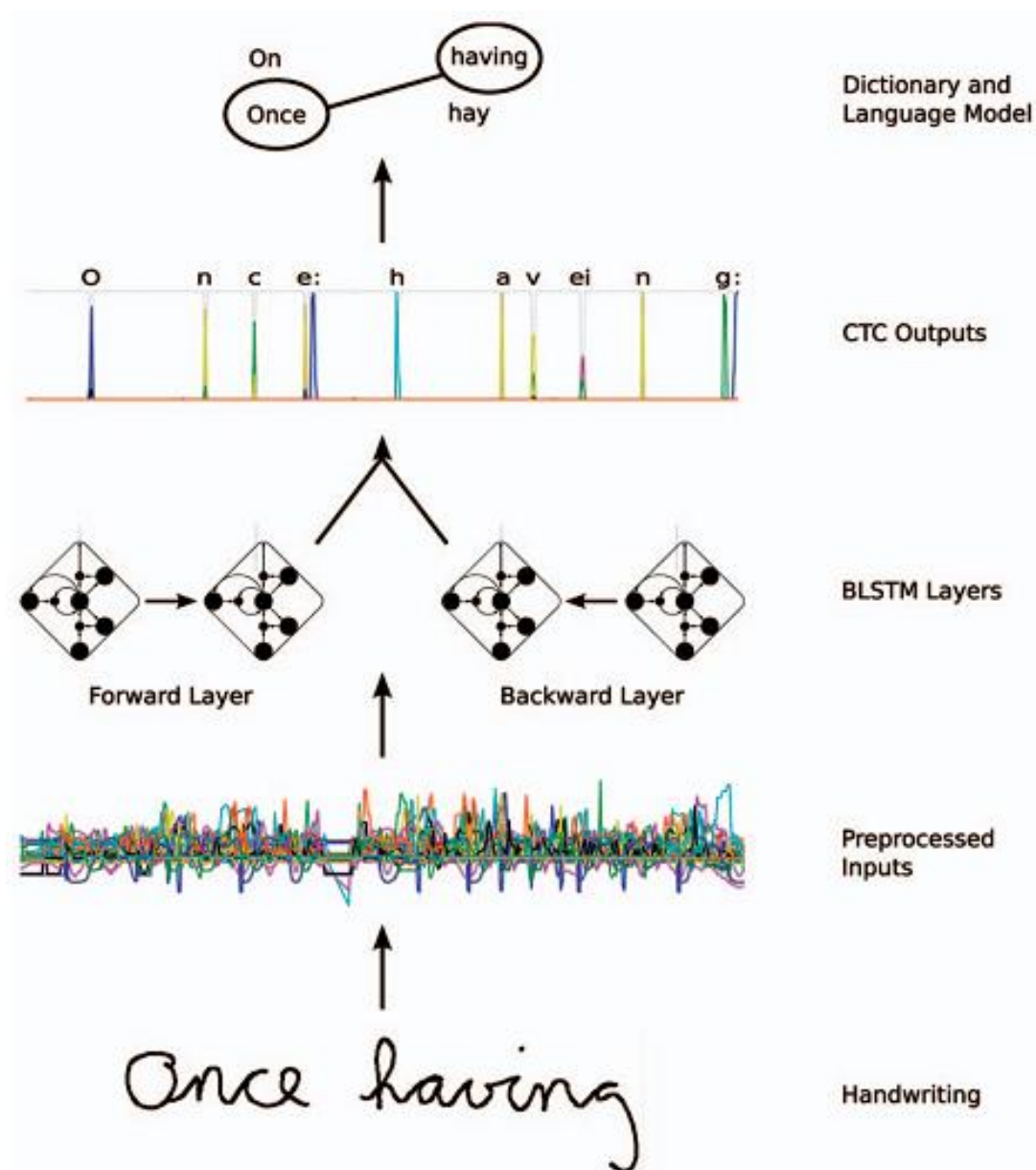


Figure 1.17. La reconnaissance avec les RNN [60]

1.3.2.6. Post traitement :

C'est l'ensemble des techniques qui permettent d'améliorer la reconnaissance par le raffinement des décisions prises dans la phase précédente.

Différentes méthodes sont utilisées dans le post-traitement y compris la correction d'erreur manuelle, correction d'erreur basée sur le dictionnaire et la correction d'erreurs basée sur le contexte [61].

1.4. Les problèmes rencontrés dans la reconnaissance de l'écriture arabe manuscrite

La forme d'écriture de la langue arabe présente quelques problèmes au développeur des OCR qui peuvent être résumé en :

- En texte arabe, les lettres consécutives d'un mot sont connectées suivant la ligne de base utilisée. Pour s'adapter à la ligne de base utilisée l'alphabet arabe a quatre formes pour chaque lettre (isolé, initial, médial, final). plusieurs lettres ne suivent pas cette règle et peuvent avoir différentes formes médial et final et quand l'un des caractères non joignable est inclus dans le mot alors la lettre la précédant décide de sa forme final (ou isolé) et elle-même décide de sa forme initial (ou final). Cette caractéristique de connectivité cause un niveau de difficulté dans l'étape de segmentation. Par rapport aux autres langues l'écriture arabe est toujours cursive et la vitesse de reconnaissances de ses caractères est faible.
- La plupart des caractères contient des points en plus de leur corps par exemple ش qui a le corps de la lettre س avec trois points au-dessus, les points peuvent être effacés par erreur dans la phase de prétraitement ce qui conduira à des erreurs de reconnaissance.
- En générale le style de l'écriture arabe peut être imprimé (Naskh) ou manuscrit (Ruqq'a), comme on peut trouver d'autre qui sont utilisé pour la calligraphie décorative (Kofi, Thuluth et Diwani). Appart les professionnels, des gens simples ne peuvent pas suivre toutes les règles de l'écriture manuscrite ce qui cause plus de difficulté pour la reconnaissance et rend les bases de données du système volumineuses.
- Quelques caractères dans l'écriture manuscrite peuvent être écrits l'un sur l'autre formant des blocs connecté (ligatures) ou déconnecté créant des chevauchements verticaux ou partagés le même espace horizontal (overlaps) ce qui augmente la difficulté de segmentation des caractères (voir figure 1.18).

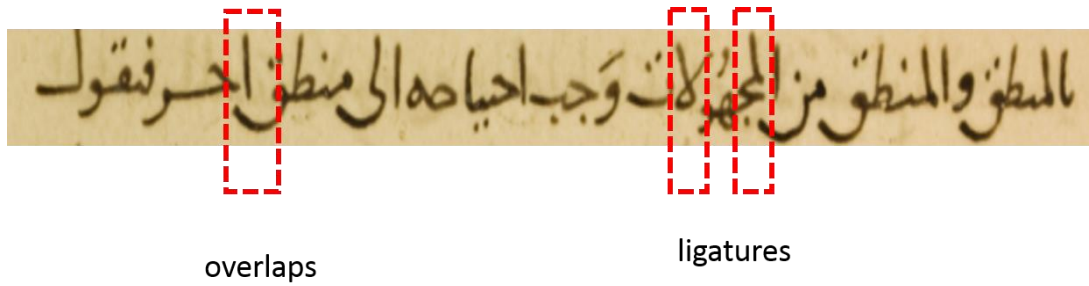


Figure 1.18. Problèmes liés à l'écriture manuscrite.

1.5. Conclusion

Dans ce chapitre nous avons présenté l'écriture arabe, ses caractéristiques, le processus de reconnaissance de l'écriture qui est composé de plusieurs phases à savoir l'acquisition, le prétraitement, la segmentation, l'extraction d'attributs et la classification, ainsi que les problèmes rencontrés lors de reconnaissance de l'écriture arabe. La phase d'extraction d'attributs est une phase importante dans les systèmes de reconnaissance. Alors le choix des attributs adéquats est une étape cruciale surtout dans la reconnaissance de l'écriture arabe manuscrite. Ainsi, le prochain chapitre présentera un état de l'art sur les méthodes d'extraction d'attributs pour la reconnaissance de l'écriture arabe manuscrite.

Etat de l'art sur l'extraction des Attributs de reconnaissance de l'écriture arabe

L'extraction d'attributs est l'une des étapes les plus importantes dans les systèmes de reconnaissance de l'écriture manuscrite, notamment l'écriture arabe. Dans la littérature, les chercheurs ont classé les attributs en deux catégories : statistiques et structurelles [29]. Les attributs statistiques sont des mesures numériques calculées sur des images ou des régions d'images. Elles comprennent les densités de pixels, les histogrammes des directions de code de chaîne, les moments, les descripteurs de Fourier, etc. [29]. Les attributs structurels sont des aspects intuitifs de l'écriture, tels que les boucles, les points de branchement, les points terminaux, les points diacritiques, etc. [29]. Dans ce qui suit nous présentons une liste non exhaustive des méthodes d'extraction d'attributs appliquées à la reconnaissance de l'écriture arabe manuscrite.

2.1. GSC (gradient, structurel et concavités)

Il s'agit d'un attribut multi résolution qui combine trois attributs différents de la forme du caractère, le gradient (représentant l'orientation locale des traits), les attributs structurels (qui étendent le gradient à de plus grandes distances et fournissent des informations sur les trajectoires des traits) et les attributs de concavité (qui capturent les relations entre les traits à de grandes distances) [62].

C'est un attribut qui s'adapte bien à l'écriture arabe vu qu'il combine trois attributs complémentaires pour la représentation de l'écriture arabe. Il est utilisé par plusieurs auteurs sur l'écriture en générale. Plusieurs auteurs l'ont appliqué sur l'écriture arabe et sur les chiffres [63-65].

Dans [64] les attributs GSC sont calculés comme suit : tout d'abord, la direction du gradient à chaque pixel de l'image est calculée afin de construire une carte de gradient à partir de l'image normalisée. Les attributs du gradient sont obtenus en quantifiant les pixels ayant le même gradient. Les attributs structurels détectent des motifs complexes dans le contour. Pour calculer les attributs de concavité, l'image est divisée en Bines. Dans chaque Bine, les pixels de certaines régions, comme les trous, sont détectés et comptés.

2.2. Attributs directionnels

Les caractères comprennent des traits qui sont des lignes, des courbes ou des poly lignes orientées. L'orientation ou la direction des traits joue un rôle important dans la différenciation des divers caractères. Depuis longtemps, l'orientation ou la direction du trait est prise en compte dans la reconnaissance des caractères basée sur l'analyse du trait. Dans un premier temps, la reconnaissance de caractères à l'aide de caractéristiques de direction était appelée "correspondance des formes directionnelles"(directional pattern matching).[66]

L'orientation (direction) locale du trait d'un caractère peut être déterminée de différentes manières : orientation du squelette [67],segment de trait [68] , code de chaîne du contour [33, 55, 69, 70], direction du gradient[33, 42, 63-65] , etc. Les attributs code de chaîne du contour et la direction du gradient sont largement utilisé parce qu'ils sont simples à mettre en œuvre et qu'ils sont à peu près invariables à la variation de la largeur du trait.[66]

Dans [55], les caractéristique extraites sont basées sur l'analyse des contours de l'image du mot et la détection morphologique des limites de l'image du mot. Une fenêtre glissante est déplacée de gauche à droite pour extraire l'histogramme des directions du code de chaîne du contour de chaque image du mot. La figure 2.1 montre un exemple d'histogramme d'orientation de contour (quatre orientations, 4×4 zones).

Dans [63], les auteurs appliquent à l'écriture arabe manuscrite l'attribut directionnel (gradient) proposé par favata et al [65]. L'attribut est calculé comme suit : après avoir divisé l'image du chiffre en grille de $N \times M$, le gradient est extrait pour chaque segment du chiffre. Les attributs basés sur le gradient sont calculées par convolution de deux opérateurs de Sobel 3×3 avec l'image binaire. Le gradient

d'un pixel central est calculé en fonction de ses huit plus proches voisins. La direction du gradient est utilisée dans le calcul d'un vecteur d'attributs de la carte de caractéristiques de gradient. La direction est divisée en 12 régions sans chevauchement ($1^\circ - 30^\circ, 31^\circ - 60^\circ \dots 331^\circ - 360^\circ$). Dans chaque région d'échantillonnage, un histogramme des directions de gradient est estimé, ce qui donne 12 entités de gradient pour chaque segment d'image. Cela correspond au décompte de chaque direction du gradient dans la région. Ces compteurs sont concaténés pour donner l'attribut gradient de l'image du chiffre.

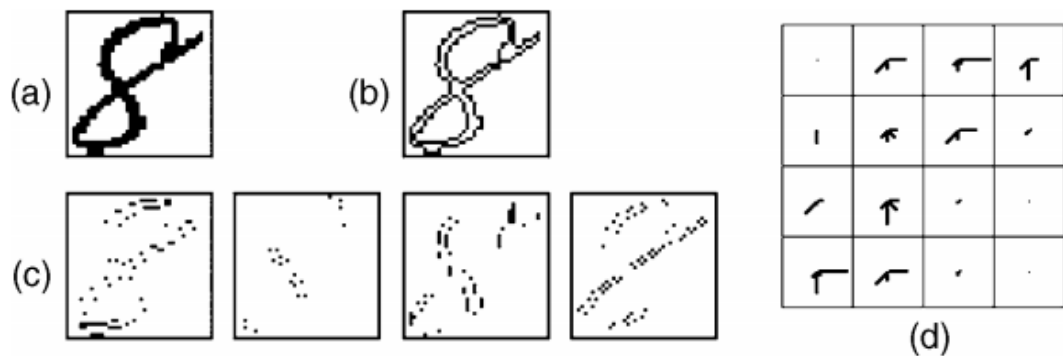


Figure 2.1. Histogramme d'orientation de contour. a) l'image d'un caractère ; b) contour du caractère; c) plan d'orientation; d) histogrammes d'orientation local [66]

2.3. Histogrammes

L'histogramme d'une image mesure la distribution des intensités dans cette image. Pour une image monochrome (une seule composante,) l'histogramme est défini comme une fonction discrète qui associe à chaque valeur d'intensité (k) le nombre de pixels prenant cette valeur (k). Les histogrammes sont très utilisés dans les méthodes d'extraction d'attributs. On trouve, Histogrammes de Gradients Orientés (HOG) [33, 71] (voir figure 2.2), Histogrammes de code de chaine du contour [33], les histogrammes de projections [72] etc. Les histogrammes de projections sont obtenus par projections horizontale et verticale des pixels noirs (encre) de l'image binaire comme illustré dans la figure 2.3. Les attributs utilisés peuvent être directement les valeurs des histogrammes éventuellement normalisés, ou bien extraites de ces histogrammes en cherchant par exemple à détecter les pics [40].

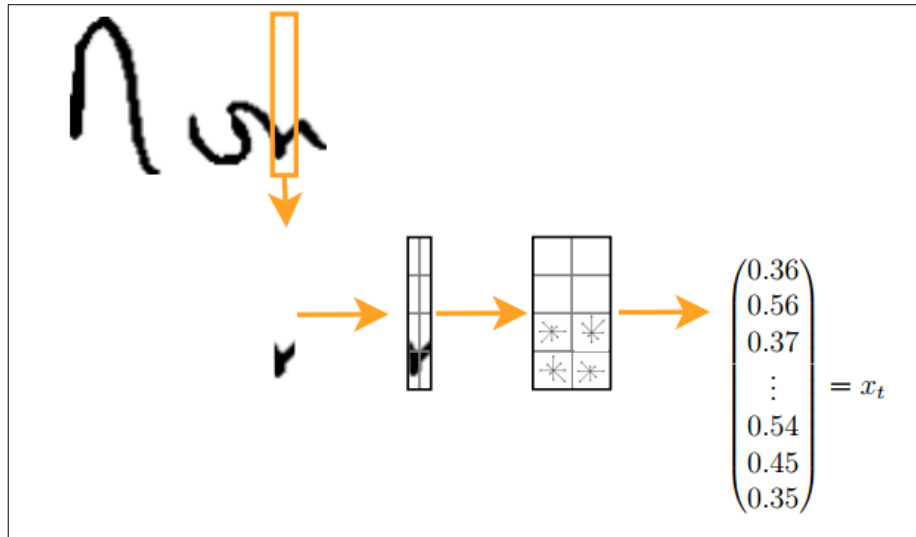


Figure 2.2. Histogramme de gradients orienté [73]

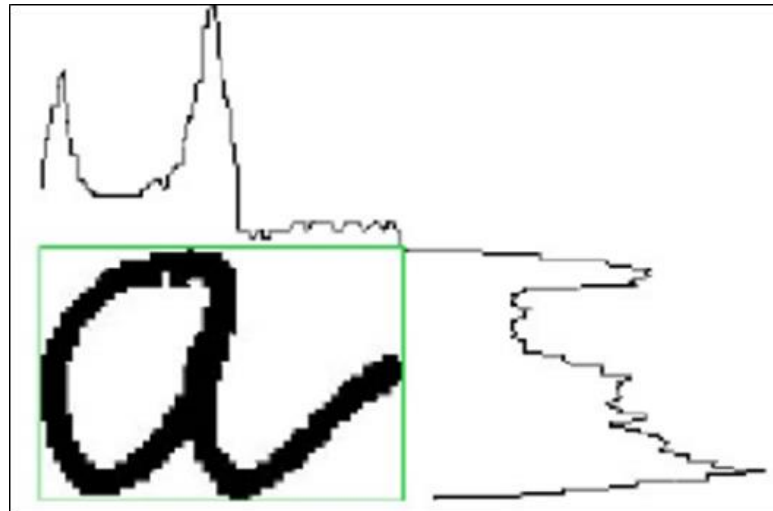


Figure 2.3. Histogramme de projection verticale et horizontale [74]

2.4. Motifs binaires locaux (LBP)

LBP est un descripteur conçu initialement pour l'analyse des textures. Il a été proposé initialement par Wang et al [75] puis modifié par T. Ojala et al [76] [77]. LBP est ensuite utilisé dans plusieurs applications notamment dans la reconnaissance de l'écriture arabe[30, 78] . Le principe général de la méthode basée sur les LBP consiste à assigner un code binaire à chaque pixel de l'image en fonction de ses 8 voisins, disposés sur une grille qui peut être de forme carrée. La valeur du niveau de gris du pixel central est utilisée pour le seuillage de ses pixels voisins pour ensuite générer un motif binaire comme illustré dans la figure 2.4. Les pixels de ce motif binaire sont alors transformés en un code LBP.

Dans [78], les auteurs extraient les attributs à partir de différentes régions de l'image du caractères, ce qui permet de détecter les variations dans l'écriture du caractère. Avant de calculer leur attribut ils divisent d'abord leurs images en considérant trois différentes configurations de divisions. Ensuite, les histogrammes LBP sont calculés dans chaque configuration.

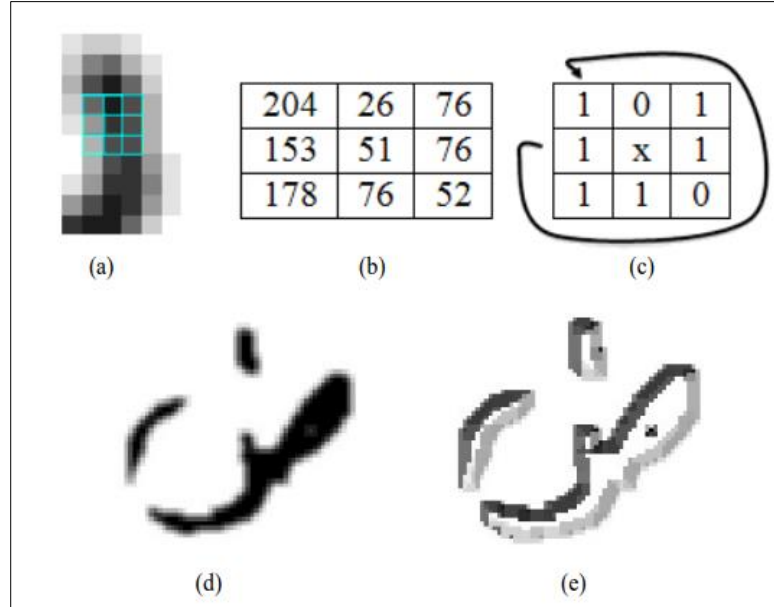


Figure 2.4. Exemple de calcul LBP : (a) une fenêtre de 3x3 centrée sur un pixel, (b) les niveaux de gris dans la fenêtre, (c) seuillage du voisinage par rapport au pixel x, (d) l'image d'un caractère après prétraitement, (e) résultat de l'application de LBP sur l'image dans (d) [78].

2.5. Les moments

Les moments et les fonctions des moments ont été utilisés comme attributs de formes dans de nombreuses applications pour reconnaître des formes d'image bidimensionnels. Ce type d'attributs extrait les propriétés globales de l'image telles que la zone de forme, le centre de la masse, le moment d'inertie etc. La définition générale des fonctions de moment m_{pq} de l'ordre $(p + q)$ pour une $X \times Y$ fonction d'intensité d'image continue $f(x, y)$ est défini dans l'équation 1 [66].

$$m_{pq} = \int_y \int_x \varphi_{pq}(x, y) f(x, y) dx dy \quad (1)$$

où p, q sont des nombres entiers entre $[0, \infty)$, x et y sont les coordonnées d'un point de l'image, et $\psi_{pq}(x, y)$ est la fonction de base. De même, la définition générale d'une image numérique $X \times Y$ peut être obtenue par le remplacement des intégrales par des sommations telles que définies dans l'équation 2

$$m_{pq} = \sum_x \sum_y \varphi_{pq}(x, y) f(x, y) \quad (2)$$

où p, q sont des nombres entiers entre $(0, \infty)$ et représentent l'ordre, x et y sont les x- et y-pixel de l'image numérique, et $\varphi_{pq}(x, y)$ est la fonction de base.

Les moments ont été utilisés comme descripteurs pour la reconnaissance de l'écriture notamment l'écriture arabe manuscrite. Plusieurs variantes ont été appliquées notamment les moments de Hu [79], les moments de Zernike [79, 80], les moments centrés [80], les moments de Tchebychev [81] etc.

Dans [80] les auteurs utilisent les moments de Zernike ainsi que les moments centré normalisé comme attributs. D'abord, les caractères sont segmentés et partitionnés en corps et composants secondaires. Ensuite, les moments sont extraits du corps du caractère, des composants secondaires ainsi que de tout le caractère. L'avantage d'utiliser ces attributs est qu'ils sont insensibles à la translation, rotation et changement d'échelle, ce qui permet de préserver les informations contenues dans les images.

2.6. Descripteurs de fourrier

Les descripteurs de Fourier sont utilisés pour décrire la forme (courbe fermée) de tout objet trouvé sur une image d'entrée. Comme les descripteurs de Fourier nécessitent une représentation 1D des informations de limite, des signatures de forme sont utilisées. Une signature de forme mappe une représentation 2D d'une forme à une représentation 1D. Il existe de nombreuses signatures de forme parmi lesquelles on trouve les positions complexes, la distance au centre de gravité de la courbe, la courbure et la fonction angulaire cumulée. [66]

Plusieurs travaux ont été réalisés en utilisant les descripteurs de Fourier pour la reconnaissance de l'écriture arabe [82] [83] [84].

2.7. Transformation des caractéristiques invariantes à l'échelle

Le descripteur de transformation de caractéristiques invariantes à l'échelle connue sous le nom anglais SIFT (Scale Invariant Feature Transform), a été proposé par Lowe [85], il transforme les données d'une image en coordonnées

invariables à l'échelle par rapport aux caractéristiques locales. L'extraction de cet attribut suit plusieurs étapes

- **Détection des extrema dans l'espace d'échelle** : La première étape du calcul recherche toutes les échelles et tous les emplacements de l'image. Elle est mise en œuvre en utilisant une fonction de différence des gaussienne pour identifier les points d'intérêt potentiels qui sont invariants à l'échelle et à l'orientation.
- **Localisation du point clé** : A chaque emplacement candidat, un modèle détaillé est ajusté pour déterminer l'emplacement et l'échelle. Les points clés sont sélectionnés en fonction des mesures de leur stabilité.
- **Attribution de l'orientation** : Une ou plusieurs orientations sont assignées à chaque point clé en fonction des directions du gradient de l'image. Toutes les opérations futures sont effectuées sur des données d'images qui ont été transformées par rapport à l'orientation, l'échelle et l'emplacement assignés pour chaque élément, ce qui permet d'assurer l'invariance de ces transformations.

Ainsi le descripteur SIFT construit un histogramme de gradient des orientations des points d'échantillonnage dans une région autour du point clé, trouve la valeur d'orientation la plus élevée et toutes les autres valeurs qui se situent à moins de 80 % de la plus élevée, et les utilise comme orientation dominante du point clé [86].

La figure 2.5 montre une représentation schématique de l'algorithme SIFT ; où les orientations et les amplitudes du gradient sont calculés à chaque pixel puis pondérés par une atténuation gaussienne (indiquée par cercle superposé). Un histogramme d'orientation du gradient pondéré est ensuite calculé pour chaque sous-région (4*4).

Le Descripteur SIFT est appliqué dans le domaine de reconnaissance de forme, et de reconnaissance de l'écriture notamment la reconnaissance de l'écriture arabe manuscrite [31, 87].

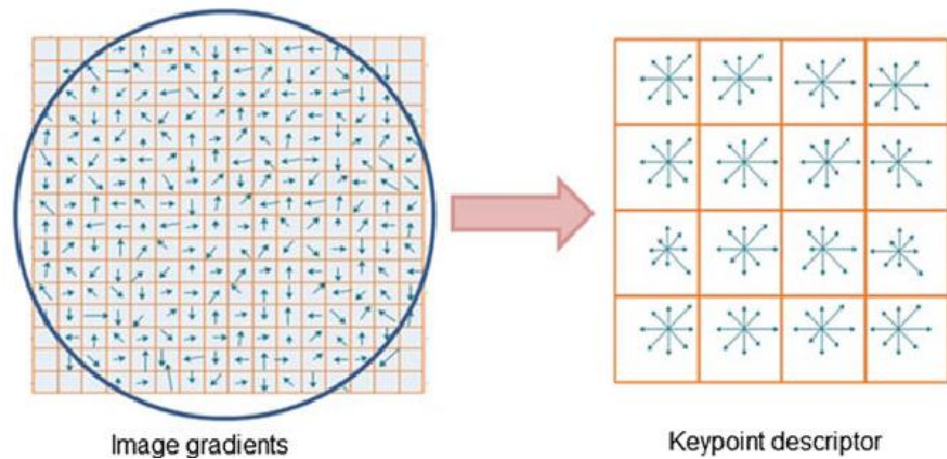


Figure 2.5. Une représentation schématique du descripteur SIFT pour un patch de 16×16 pixels et un tableau de descripteurs 4×4 [86]

Dans [87] les auteurs tente de détecter les parties courbées du trait dans les images de mots arabes. Pour ce faire, le détecteur de Harris [88] est utilisé pour détecter les angles (coins) dans les mots comme illustré dans la figure 2.6. Ensuite, les points clés sont détectés pour chaque coin en utilisant les descripteurs SIFT comme illustré dans la figure 2.7. Enfin, le sac d'attributs (Bag of features) est créé en ne considérant que les points clés intéressant en se servant d'une fenêtre glissante.

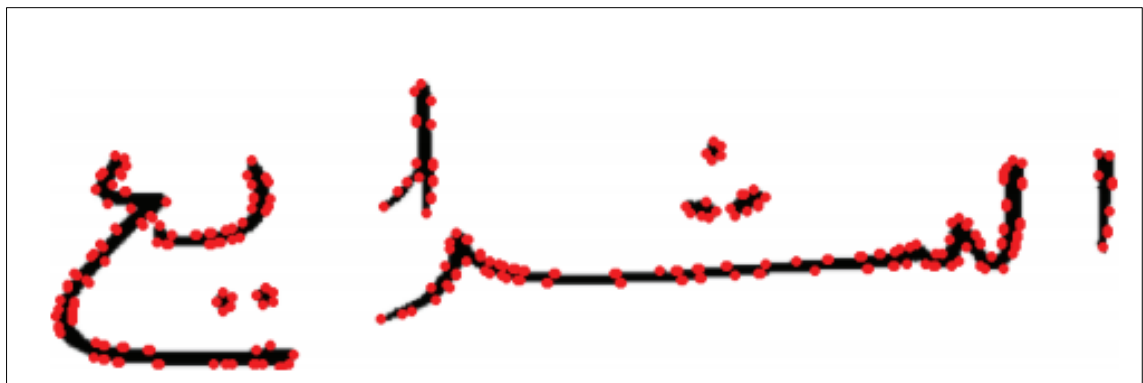


Figure 2.6. Les points d'angle en appliquant le détecteur de Harris [87]

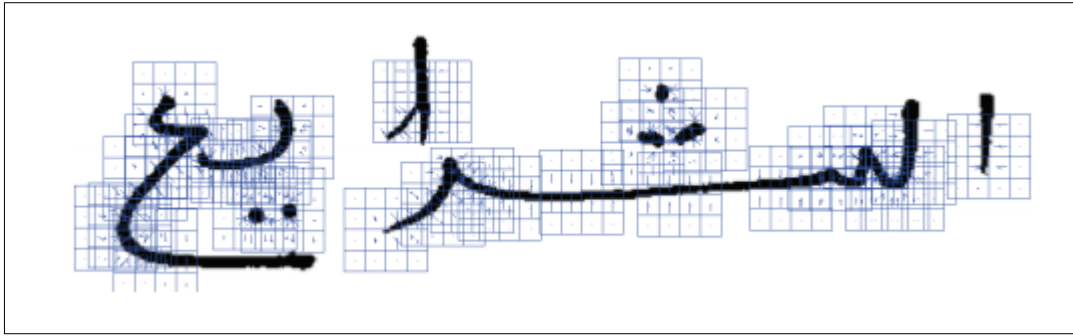


Figure 2.7. Plusieurs descripteurs SIFT [87]

2.8. Les attributs structurels

Les attributs structurels ont une relation directe avec les caractéristiques et les propriétés topologique de l'écriture. Parmi ces propriétés on cite la présence de boucle, des arcs, des concavités, des segments de ligne droites, de points diacritiques, des ascendants (jambes), et descendants (hampes). Ces attributs structurels opèrent aussi en utilisant des points clés tel que les points terminaux, les points de croisement, et les points de branchement. Ces caractéristiques peuvent être extraite directement de l'image du mot, ou sur les images prétraitées, tel que les squelettes des mots ou leurs contours.

Dans une image squelettisée, les pixels noirs qui ont un certain nombre de voisins noirs non égaux à 0 ou 2 reçoivent le nom de points caractéristiques. Les points caractéristiques sont classés en trois types : points terminaux, points de branchement et points de croisements.

Le point terminal, est défini comme le début ou la fin d'un segment de ligne. Un point de branchement est un point de jonction reliant trois branches. Un point de croisement est un autre point de jonction qui relie quatre branches [5]. La figure 2.8 illustre visuellement les trois types de points caractéristiques.

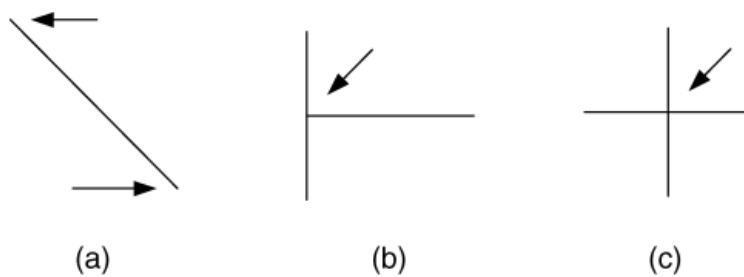


Figure 2.8. Les types de points (a) points terminaux; (b) points de branchement; (c) points de croisement [66]

Dans [89], les auteurs ont divisé les images des mots en trois colonnes et trois lignes. Par la suite, ils extraient les attributs structurels à savoir les boucles (figure 2.9(a)), les points diacritiques (figure 2.9(a)) en considérant leurs types, nombre et positions dans le mot, les ascendants (Stem)(figure 2.10), les descendants (Leg)(figure 2.11), la position et le nombre de PAW et la transition de pixels dans la représentation des mots. Ces attributs sont ensuite passés aux classifieurs pour la reconnaissance.



Figure 2.9. Extraction de : a)points diacritiques. b) boucles [89]

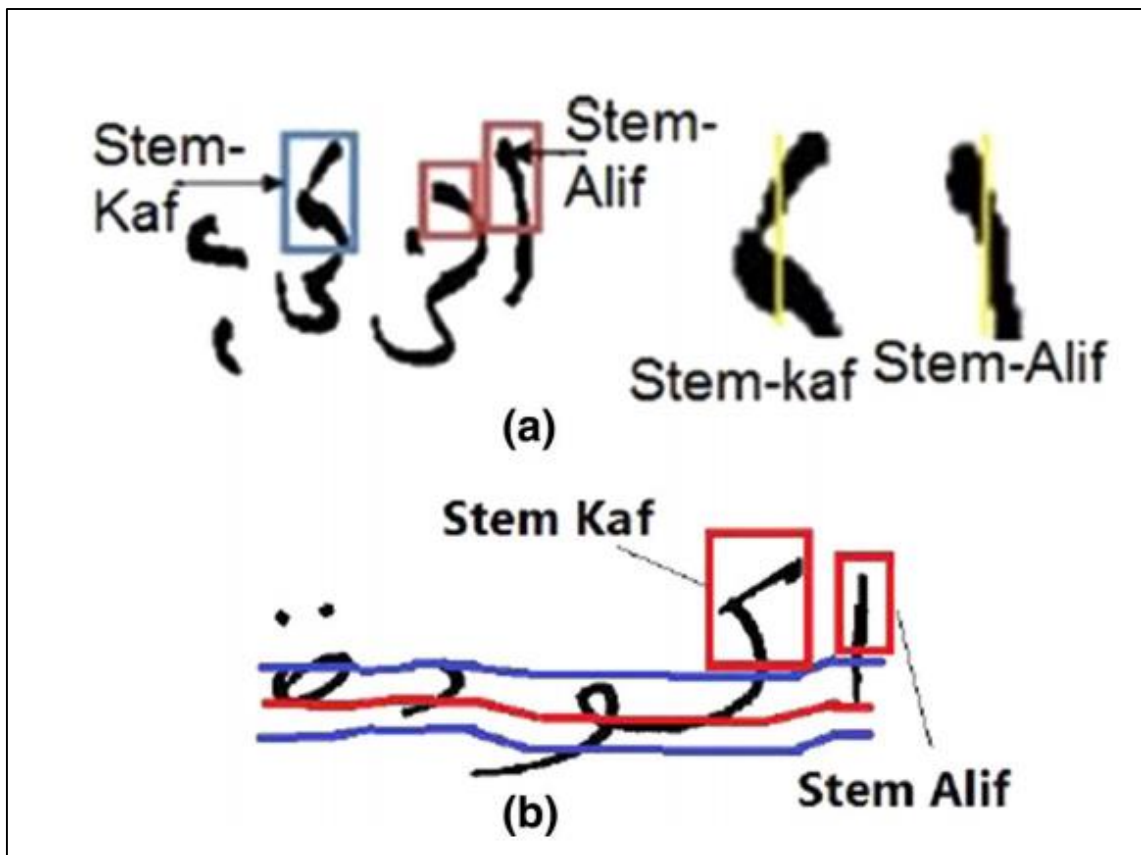


Figure 2.10. Extraction ascendants (Stem) de deux mots différents [89].

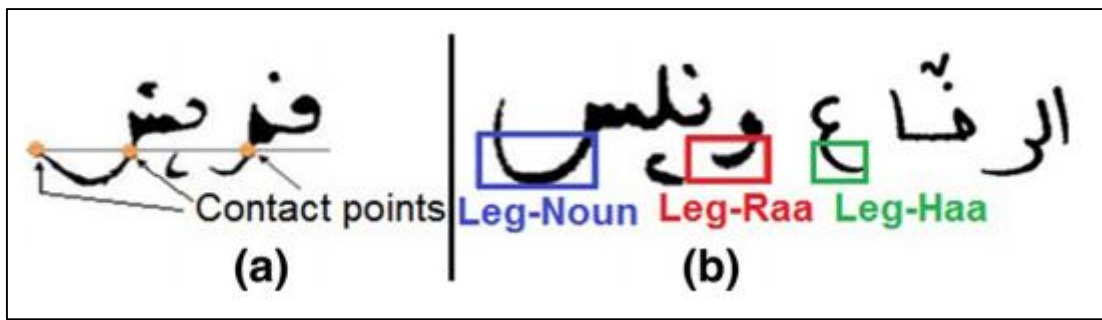


Figure 2.11 Extraction de descendeurs. a) point d'intersection du mot avec la ligne de base. b) quelques descendants (Leg) [89]

Dans [90], les mots sont segmentés en traits. Les traits sont utilisés comme une partie de base pour représenter l'écriture arabe. Les caractéristiques géométriques et topologiques sont ensuite utilisées pour classer les traits. Ces caractéristiques sont calculées en utilisant les points terminaux et les points de croisements. Les positions des traits classés sont examinées pour pouvoir les classer dans leur classe adéquate.

Dans [91], les auteurs ont utilisé à la fois des attributs structurels et statistiques. Les images des mots ont été d'abord amincies. Par la suite, les points terminaux et les points d'intersection sont détectés sur les squelettes des mots, puis utilisés pour le partitionner en primitives. Ensuite, des caractéristiques statistiques sont calculées sur chaque primitive. Les caractéristiques de toutes les primitives sont ensuite concaténées générant un code global décrivant la structure topologique du squelette.

Dans [32], les attributs utilisés représentent les attributs géométriques et quantitatifs d'un mot. Il est basé sur l'angle, l'emplacement, la taille et le nombre de lignes droites. Premièrement, l'image du mot est divisée en $m \times n$ fenêtres et des lignes droites sont extraites dans chaque fenêtre. Ensuite, des vecteurs de caractéristiques sont extraits de ces lignes et combinés tous ensemble.

Dans [84], quatre attributs ont été utilisées, y compris la description des pixels, les descripteurs de Fourier, filtre de Gabor et description structurelle (ascendeur, descendeur, boucle et points). Ces fonctionnalités permettent de capturer la distribution spatiale des pixels de l'image du mot, l'orientation des bords, des courbures et d'autres caractéristiques de l'écriture.

2.9. Conclusion

Dans ce chapitre nous avons présenté un état de l'art sur les méthodes d'extraction d'attributs utilisées dans les systèmes de reconnaissance de l'écriture arabe manuscrite. En analysant ces méthodes d'extraction d'attributs nous constatons que les auteurs se concentraient sur l'analyse des pixels d'encre pour extraire leurs attributs. Alors qu'une image de texte est constituée de pixels d'encre ayant une certaine distribution spatiale dans le fond (arrière-plan) de l'image. Ainsi, l'idée de proposer un attribut qui impliquera le fond des images de texte dans l'extraction de caractéristiques.

Dans le prochain chapitre nous allons présenter notre proposition qui s'agit d'un nouvel attribut pour la reconnaissance de l'écriture arabe manuscrite.

HMB nouvel attribut pour la reconnaissance de l'écriture arabe manuscrite.

La reconnaissance de l'écriture passe par plusieurs étapes l'acquisition, le prétraitement, la segmentation, l'extraction d'attributs et la reconnaissance. La reconnaissance dépend fortement de la méthode d'extraction d'attributs. Plusieurs facteurs doivent être pris en considération lors de la conception d'un nouvel attribut à savoir la nature du texte (cursif), les types des scripts qui peuvent être manuscrits ou imprimés et la langue de l'écriture (arabe, latin, etc.). Un nouvel attribut pour la reconnaissance de l'écriture arabe manuscrite sera l'objet de ce chapitre.

3.1. Extraction des attributs : Histogramme du fond marqué (HMB)

Une image de l'écriture est constituée de pixels d'encre ayant une certaine distribution spatiale dans le fond (arrière-plan) tel qu'illustré dans la figure 3.1. La distribution spatiale de ces pixels d'encres renseigne sur la forme et les traits de l'écriture. Dans une image de texte, les pixels du fond sont également importants car ils nous indiquent les positions et les dispersions des pixels d'encre dans l'image. De plus, les pixels du fond peuvent être utilisés comme support pour insérer des informations supplémentaires dans l'image. Par conséquent, dans cette thèse nous avons proposé un nouvel attribut (pour la reconnaissance de l'écriture arabe manuscrite) appelé HMB (Histogram of Marked Background). La nouveauté avec notre contribution (HMB) est que, en plus des pixels d'encre nous utilisons également le fond de l'image du texte pour extraire les caractéristiques.



Figure 3.1. Représentation de la distribution spéciale des pixels d'une image

Le principe générale de l'attribut HMB est d'injecter des informations discriminantes dans l'image du texte et cela en attribuant des marques aux pixels du fond de l'image du texte. Les pixels de fond sont marqués selon la répartition spatiale des pixels d'encre dans leur voisinage.

Pour marquer un pixel de fond, on cherche dans son voisinage la présence de pixels d'encres dans un certain nombre de directions. Par conséquent, ces marques permettent de capturer la forme et l'orientation de l'écriture (lignes verticales, lignes horizontales, boucles, points et autres). L'attribut HMB permet ainsi de capturer les caractéristiques complexes de l'écriture arabe manuscrite. L'avantage dans l'utilisation du processus de marquage des pixels du fond est que ces informations capturées sont intégrées dans l'image du texte à reconnaître. Par la suite, les vecteurs d'attributs HMB sont extraits en calculant des histogrammes de pixels marqués à partir des images de texte marquées. Les attributs HMB sont très simples et réduisent les informations quantitatives à utiliser dans la phase de reconnaissance. La figure 3.2 montre le schéma général de l'extraction des attributs HMB.

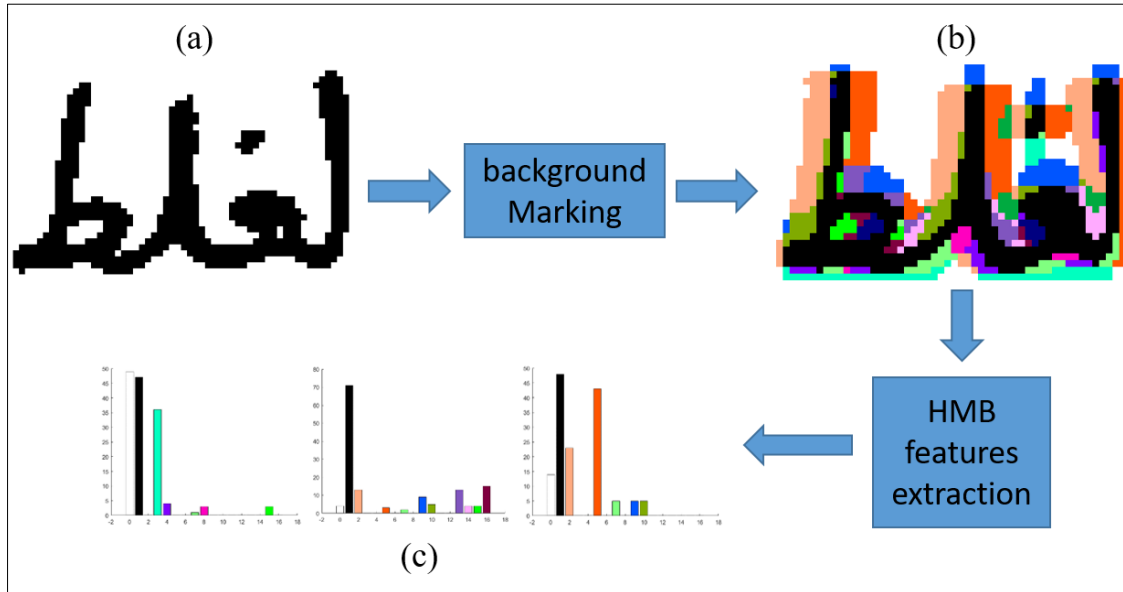


Figure 3.2. Schéma général de l'extraction des attributs HMB. a) image originale. b) image marquée. c) Les attributs HMBs

L'attribut HMB proposée est extrait en deux étapes : marquage de fond et calcul des histogrammes. Les détails de ces deux étapes sont présentés dans les sections suivantes.

3.1.1. Marquage du fond

L'idée principale de notre contribution est d'enrichir l'image du texte pour qu'elle soit plus représentative en intégrant des informations contextuelles dans le fond de l'image du texte. Pour atteindre cet objectif, des marques sont attribuées pour les pixels de fond ayant des pixels d'encre dans leur voisinage. Pour Ceci, nous avons considéré un rayon « r » pour délimiter la région dans laquelle les voisins d'encre sont identifiés.

La marque attribuée pour un pixel de fond est définie en fonction de la répartition spatiale de pixels d'encre dans son voisinage. Afin de marquer un pixel de fond, et en plus du rayon « r », nous avons considéré différentes configurations de directions dans lesquelles nous cherchons les pixels d'encre voisins. La figure 3.3 montre les configurations de directions considérées dans le processus de marquage.

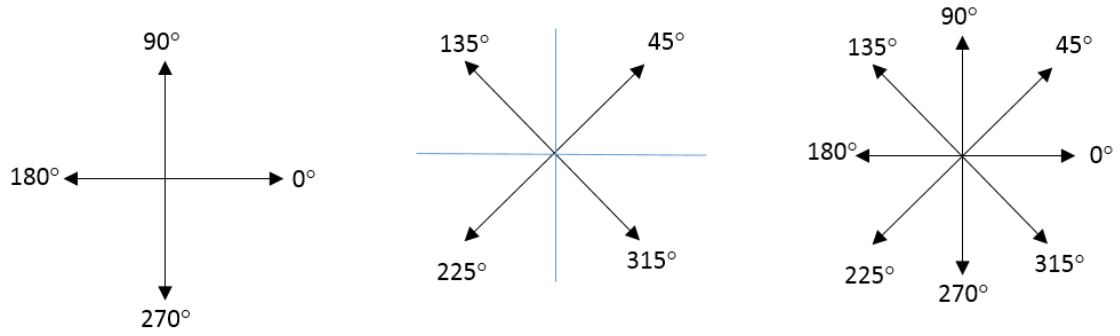


Figure 3.3. Les différentes configurations de directions utilisées dans le processus de marquage.

Pour mieux expliquer le processus de marquage, on considère une image d'un mot « I » après nettoyage et binarisation définie par l'équation (3).

$$I(i, j) = \begin{cases} 1 & \text{if } I(i, j) \text{ is ink} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Pour chaque pixel de fond $I(i, j)=0$, on détermine une marque $M(i, j)$. Pour ce faire, on considère un cercle « Cr » avec un rayon de « r » pixels centré au pixel de fond $I(i, j)$ à marquer. Ensuite, on cherche la présence de pixels d'encre dans chaque direction d'une des configurations des directions montrées dans la figure 3.3. Enfin, la marque $M(i, j)$ est définie par les directions dans lesquelles nous avons trouvé des pixels d'encre dans le cercle "Cr". L'image marquée est définie par l'équation. (4).

$$I_{\text{marked}}(i, j) = \begin{cases} 1 & \text{if } I(i, j) \text{ is ink} \\ M(i, j) & \text{otherwise} \end{cases} \quad (4)$$

Afin de calculer la marque $M(i, j)$ pour un pixel de fond en considérant d directions, avec d est le nombre de directions considérées en utilisant l'une des configurations de la figure 3.3. Nous avons utilisé un vecteur binaire B de d cases défini par l'équation (5).

Un bin $B(k)$ est mis à 1, si un pixel d'encre est trouvé dans le cercle Cr suivant la direction k . Ensuite, la marque $M(i, j)$ est égale à la valeur décimale du code binaire dans le vecteur B telle qu'exprimée formellement dans l'équation (6).

$$B(k)_{(i, j)} = \begin{cases} 1 & \text{if } I(i, j) \text{ has ink in the direction } k \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$M(i, j) = \sum_{k=0}^{d-1} B(k)_{(i,j)} * 2^k \quad (6)$$

La valeur décimale de la marque $M(i,j)$ d'un pixel de fond donné est comprise entre 0 et $(2^d - 1)$. Étant donné que les valeurs 0 et 1 définissent respectivement les pixels de fond et d'encre (Equation. (1)). La valeur décimale de la marque $M(i,j)$ est alors modifiée pour aller de 2 à (2^d) . Les pixels marqués avec une marque *zéro* n'ont pas d'encre dans toutes les directions délimité par le cercle Cr , ainsi leur marque reste inchangée. Les pixels avec une marque supérieure ou égale à un sont décalés de un afin de les distinguer des pixels d'encre (Equation (7)).

$$M(i, j) = \begin{cases} M(i, j) + 1 & \text{if } M(i, j) \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Le processus de marquage est implémenté dans l'algorithme 1. Cet algorithme marque l'image du texte en entrée avec 8 directions. Il peut être adapté pour marquer une image avec n'importe quelle configuration des directions montrées dans la figure 3.3.

Algorithme 1. Algorithme pour marquer une image

Input: $I(h, w)$ binarized image to be marked.

r : The radius of the circle Cr in which we look for ink neighbors.

Output: $M(h, w)$ the marked image.

Begin

$M = I$;

For $i = 1$ to h **do**

For $j = 1$ to w **do**

If $I(i, j) = 0$ **then** // $I(i, j) = 0$ is background

$d = 0$;

For $p = -1$ to 1 **do**

For $q = -1$ to 1 **do**

For $k = 1$ to r **do**

$\Delta i = p * k$; $\Delta j = q * k$;

If $I(i + \Delta i, j + \Delta j) = 1$ **then** // check ink neighbors

$M(i, j) = M(i, j) + 2^d$;

break;

End if;

End for;

```
    If ( $\Delta i \neq 0$  And  $\Delta j \neq 0$ ) then
      d=d+1;
    End if;
  End for;
End for;
  M(i,j)=M(i,j)+1;
End if;
End for
End for
End
```

3.1.2. Exemples sur le processus de marquage

Dans cette section nous présentons des exemples du processus de marquage ainsi que le calcul de la marque appliqué sur l'image d'un mot. Dans ces exemples nous avons utilisé un rayon de 4 pixels, et nous avons considéré la configuration avec quatre directions (0° , 90° , 180° , 270°). Pour plus de clarté, nous les avons étiqueté par R(droite), U(haut), L(gauche) et D(bas) respectivement come illustré dans la figure 3.4 .

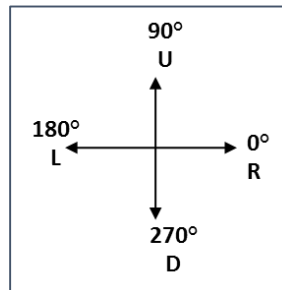


Figure 3.4. Configuration avec 4 directions

Étant donné que les valeurs 0 et 1 définissent respectivement les pixels de fond et d'encre et la valeur décimale de la marque d'un pixel de fond donné est comprise entre 2 et 2^d comme indiqué dans la section 3.1.1. Comme nous avons considéré les 4 directions illustrées dans la figure 3.4, alors la marque d'un pixel de fond varie entre 2 et 2^4 (16). La figure 3.5 montre toutes les possibilités de marques avec les quatre directions illustrées dans la figure 3.4 et leur code binaire ainsi que leurs valeurs décimales correspondantes. Pour plus de clarté lors de l'affichage des images arquées nous avons attribué une couleur différente pour chaque marque.

R	U	UR	L	LR	LU	LUR	D	DR	DU	DUR	DL	DLR	DLU	DLUR
0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Figure 3.5 : Toutes les possibilités de marques avec quatre directions et leur code binaire ainsi que leurs signes décimaux correspondants.

3.1.2.1. Calcule de la valeur décimale de la marque

On considère l'exemple dans la figure 3.6. Le pixel de fond $p(i,j)$ a un seul voisin d'encre situé dans la direction haut(u) désigné par la flèche verte. Ainsi le bin correspondant à la direction U dans le vecteur B est mis à 1. La valeur décimale correspondante au code binaire (0010) du vecteur B est égale à 2 et comme expliqué dans la section 3.1.1, cette valeur décimale est décalée de 1 et devient ainsi 3. Au finale le pixel de fond $p(i,j)$ a la marque décimale égale à 3.

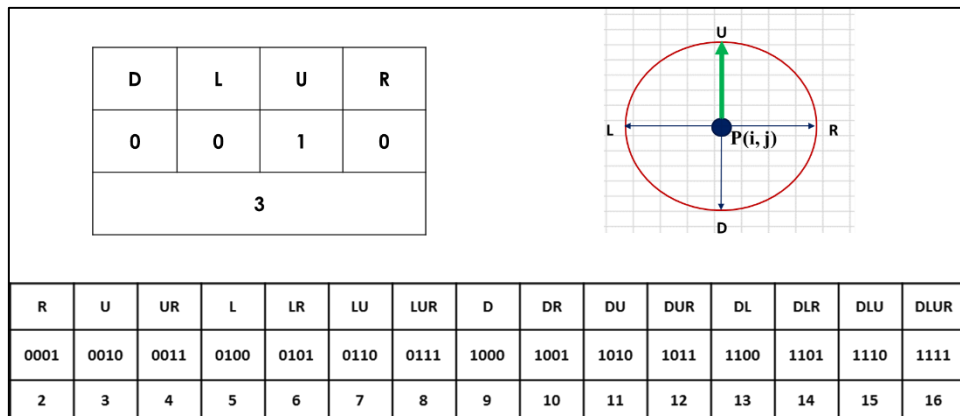


Figure 3.6. Exemple de calcul de la marque décimale d'un pixel de fond ayant un seul voisin.

On considère un deuxième exemple illustré dans la figure 3.7. Le pixel de fond $p(i,j)$ a des voisins d'encre dans toutes les directions bas (D), gauche (L), haut (U) et droite (R) désigné par les flèches vertes. Ainsi tous les bins du vecteur B sont mis à 1. La valeur décimale correspondante au code binaire (1111) du vecteur B est égale à 15 et comme expliqué dans la section 3.1.1 cette valeur décimale est décalée de 1 et devient ainsi 16. Au finale le pixel de fond $p(i,j)$ a la marque décimale égale 16.

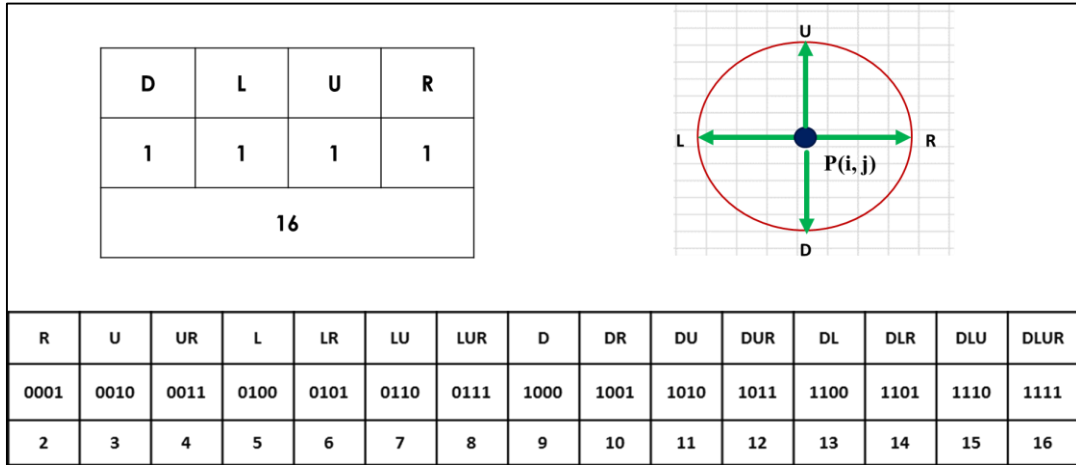


Figure 3.7. Exemple de calcul de la marque décimale d'un pixel de fond ayant un plusieurs voisins.

3.1.2.2. Exemples de marquage

Dans cette section nous présentons quelques exemples illustrant le marquage des pixels de fond d'une image d'un mot suivant les directions (0°,90°,180°,270°) et délimité par un rayon de 4 pixels. Nous avons utilisé une couleur différente pour chaque marque.

Dans la Figure 3.8 le pixel de fond mis en évidence est marqué avec la marque **R** qui correspond au code binaire (0001) et la valeur décimale **2** car il a un seul voisin d'encre dans la région délimitée par le cercle de rayon de 4 pixels et dans la direction droite (R) illustré avec la flèche verte.

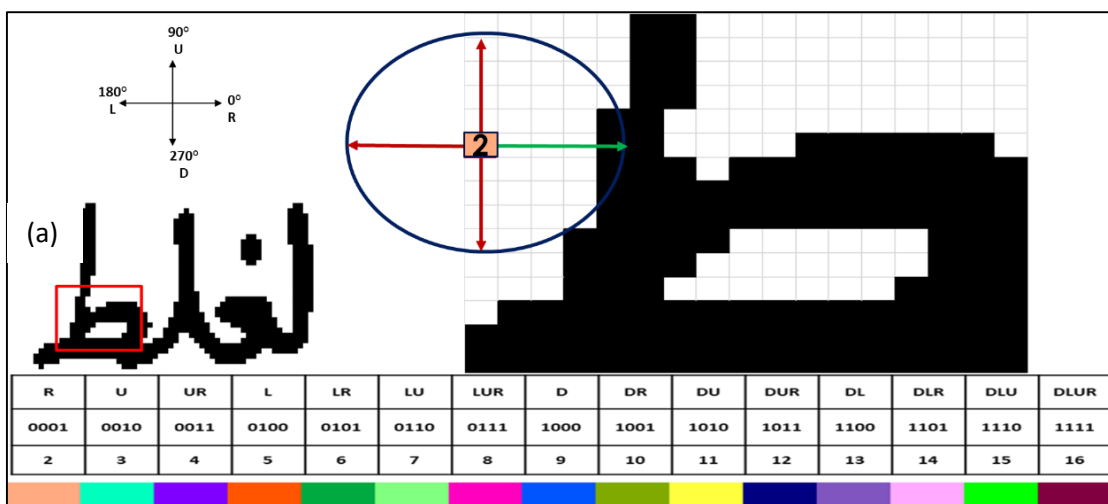


Figure 3.8. Exemple de marquage d'un pixel de fond ayant un seul voisin d'encre

La figure 3.9 montre un exemple de pixel de fond ayant des voisins d'encre dans toutes les directions bas (D), gauche (L), haut (U) et droite (R) et délimitées

par le cercle de 4 pixels comme illustré avec les flèches de couleur vertes. Ainsi la marque attribuée à ce pixel est DLUR son code binaire est 1111 et sa valeur décimale est égale à 16.

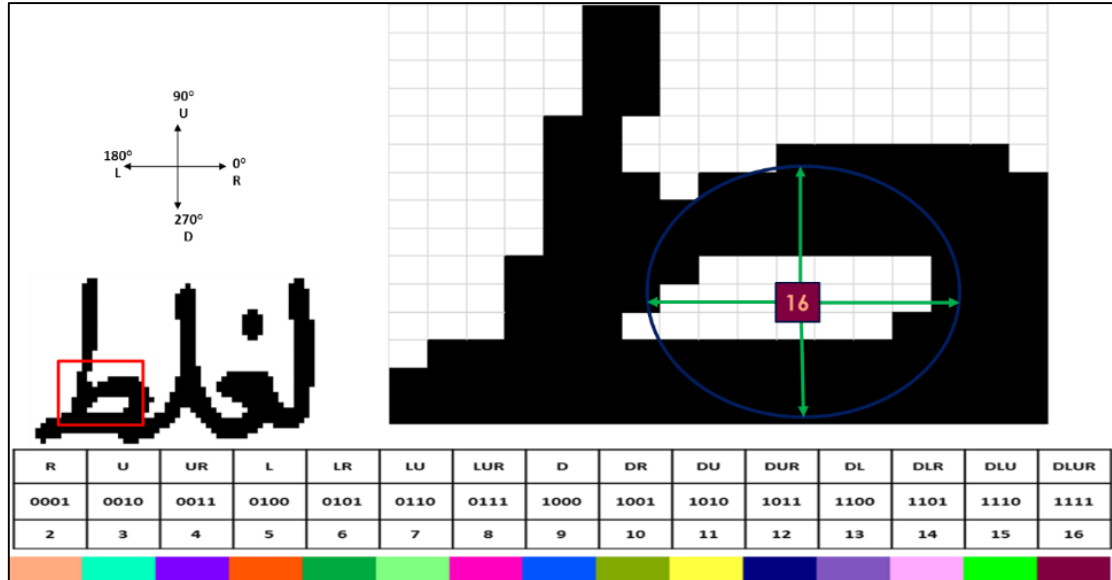


Figure 3.9. Exemple de marquage d'un pixel de fond ayant un voisin d'encre dans toutes les directions.

Le pixel de fond sélectionné dans la figure 3.10, a des voisins d'encre dans les directions gauche (L) et bas (D) comme le montre les flèches de couleur vertes dans la région délimitée par le cercle de rayon de 4 pixels. La marque correspondante à ce pixel est alors DL, son code binaire correspondant est 1100 et ça valeur décimale est égale à 13.

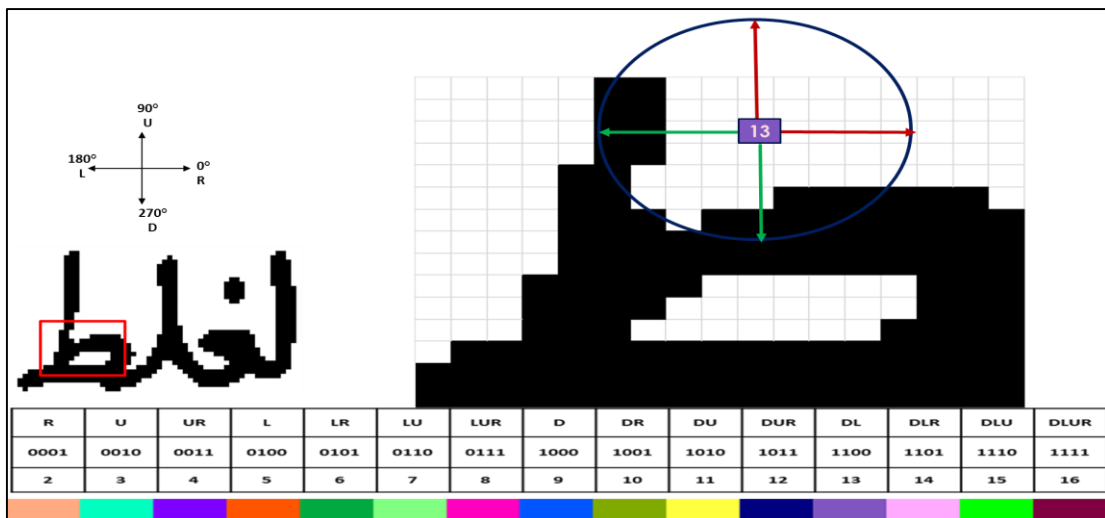


Figure 3.10. Exemple de marquage d'un pixel de fond ayant un voisin d'encre dans deux directions.

La figure 3.11 met en évidence un pixel de fond qui n'a aucun voisin d'encre dans toutes les directions de la région délimité par le cercle de 4 pixels. Cela veut dire que la valeur de ce pixel reste inchangée donc ça marque est égale à 0.

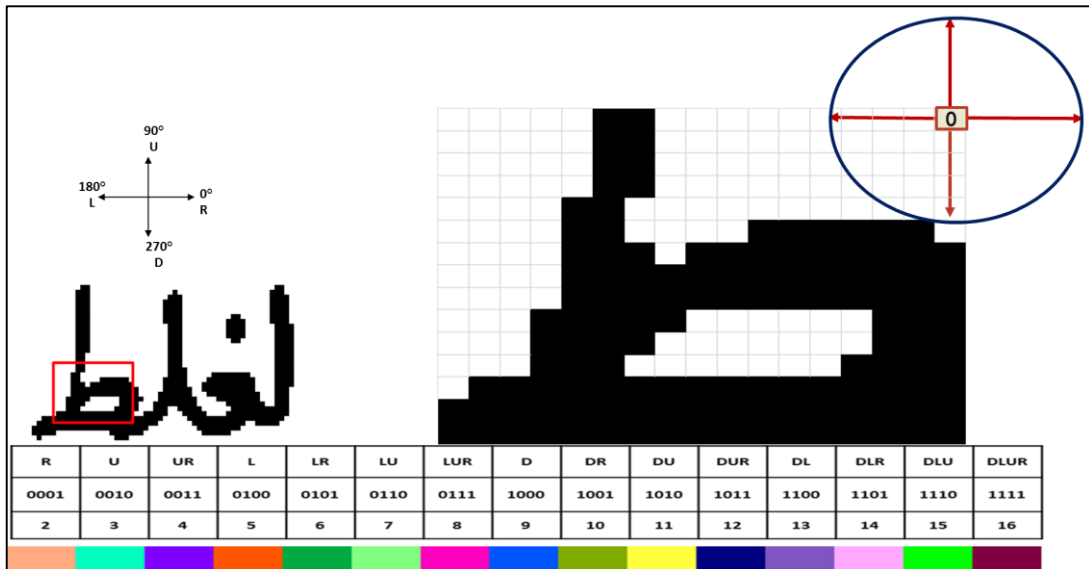


Figure 3.11. Exemple de marquage d'un pixel de fond ayant aucun voisin d'encre.

La figure 3.12 présente le processus de marquage. La figure 3.12 (c) présente les marques attribuées à la figure 3.12(b) après marquage en utilisant un rayon de 4 pixels et la configuration de directions (0°,90°,180°,270°). Les marques varient de 0 à 16. Pour plus de clarté dans l'affichage chaque pixel marqué est colorié avec la couleur qui correspond à sa marque comme illustré dans la palette des marques.

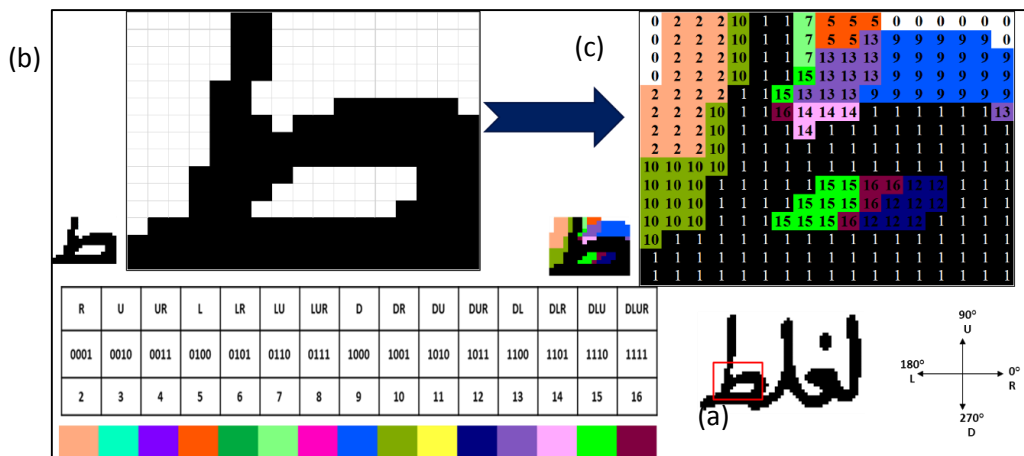


Figure 3.12. Processus de marquage. a) L'image originale. b) Partie de l'image sélectionnée dans(a). c) Résultat du marquage de l'image dans (b).

La figure 3.13 présente l'image marquée d'un mot après processus de marquage en utilisant les directions ($0^\circ, 90^\circ, 180^\circ, 270^\circ$) et un rayon de 4 pixels. Chaque marque est illustrée par une couleur différente.

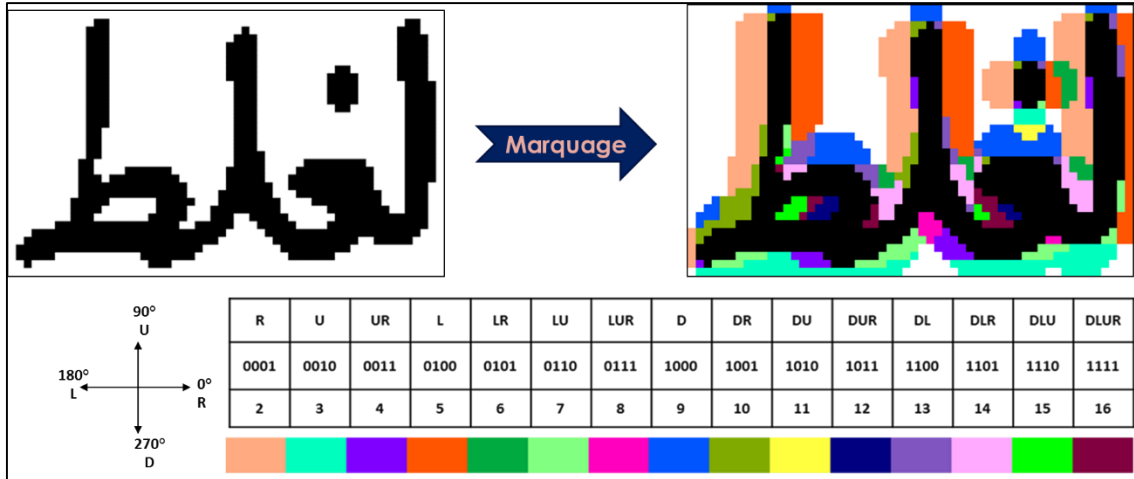


Figure 3.13. Exemple d'une image marquée.

Afin de donner une idée de la répartition des marques dans une image marquée, nous avons présenté dans la figure 3.14 quelques images marquées avec des couleurs. La figure 3.14(a) montre différentes images marquées avec différentes valeurs du rayon r (2, 4 et 6). Nous avons utilisé une couleur différente pour chaque marque. La figure 3.14 (b) montre le tableau des marques et des couleurs attribuées pour chaque marque différente.

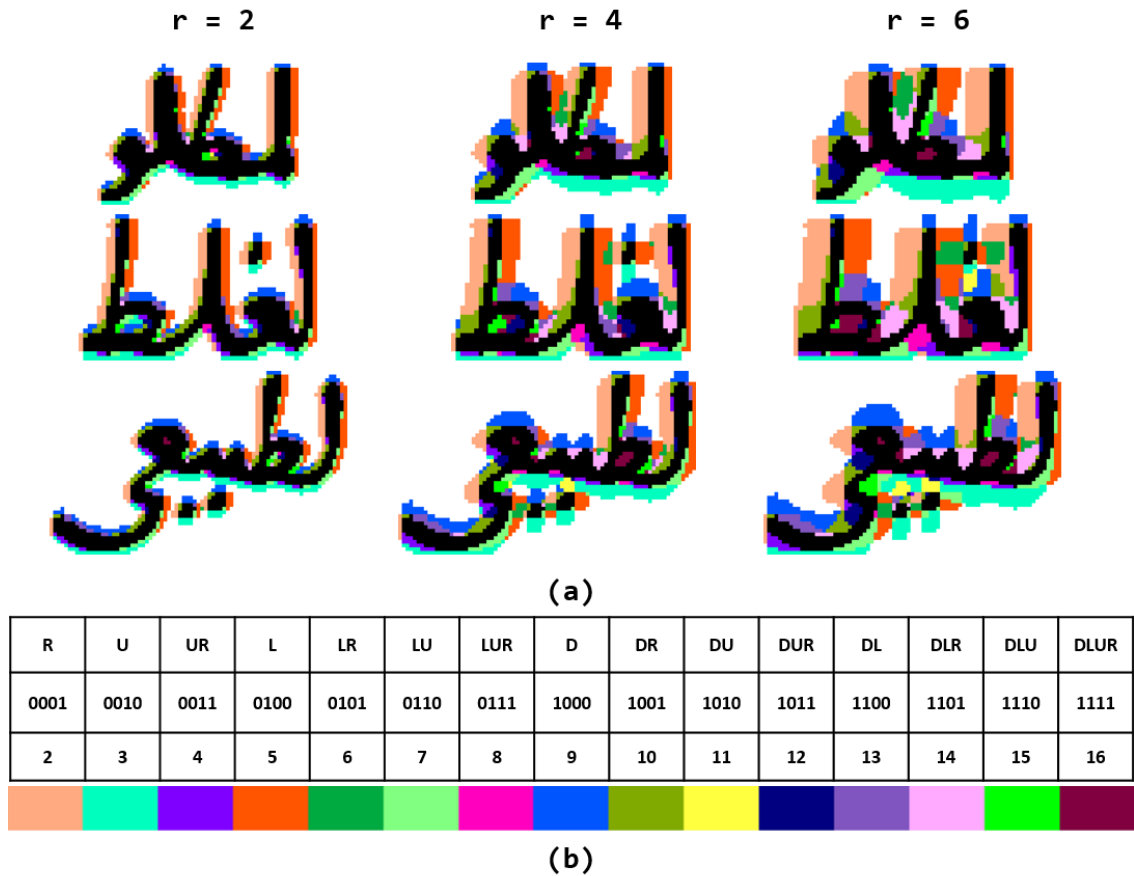


Figure 3.14. Images marquées avec des couleurs. a) Quelques images marquées avec différentes valeurs du rayon r (2, 4 et 6). b) Tableau des couleurs utilisées pour imprimer les pixels marqués en (a).

3.1.3. Calcul des histogrammes

La nouveauté avec l'attribut HMB que nous avons proposé dans cette thèse est que nous extrayons les caractéristiques des images marquées au lieu des images originales.

3.1.3.1. Préparation des images

Après avoir marqué les images de texte et afin de capturer la distribution spatiale des caractéristiques, les images de texte sont divisées en quatre blocs horizontaux comme indiqué dans la figure 3.15.

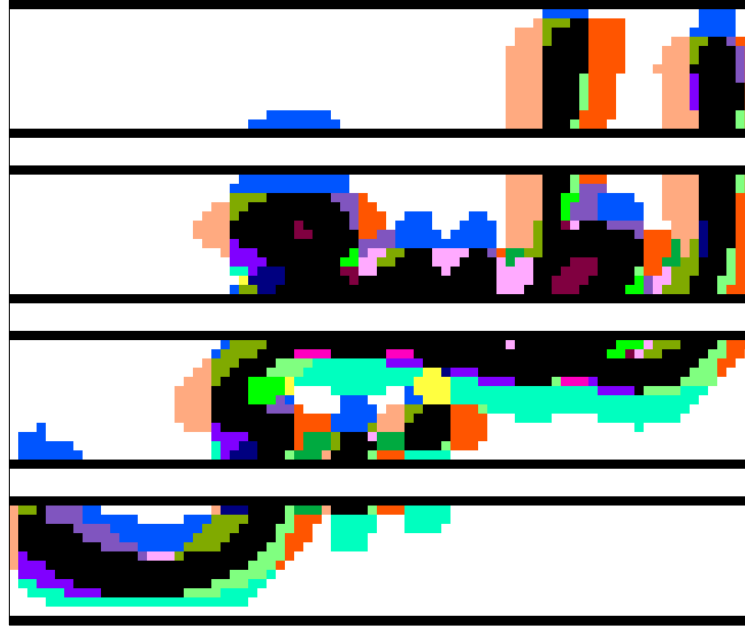


Figure 3.15. Divisions en quatre blocks horizontaux des images de texte marquées.

3.1.3.2. Calcul des vecteurs de caractéristiques

Après avoir divisé les images marquées en quatre blocs, une fenêtre glissante se déplaçant de droite à gauche est utilisé, où un vecteur de caractéristiques est calculé à chaque position de la fenêtre glissante. La caractéristique extraite de chaque image sélectionnée par la fenêtre glissante est un histogramme de 2^d+1 cases par bloc, où d est le nombre de directions utilisées pour marquer l'image. Ensuite, le vecteur de caractéristiques extrait d'une trame est un histogramme de $4 \times (2^d+1)$ cases.

HMB quantifie le nombre de pixels qui sont marqués par chaque étiquette dans l'image marquée. Équation. (8) définit l'histogramme $hist()$ qui quantifie les pixels marqués dans un bloc.

$$hist(k)_{k=0}^{k=2^d} = |\{I_{marked}(i,j) = k\}| \quad (8)$$

Par exemple, dans une image marquée avec quatre directions, les valeurs des pixels vont de 0 à 16. Par conséquent, l'image marquée aura 17 valeurs différentes. Un HMB de 17 cases est extrait de chaque bloc de la trame. Ensuite, la caractéristique extraite d'une trame est un HMB de 68 cases (17x4). Enfin, la caractéristique résultante, qui représente une image de mot, est un vecteur de HMB de $(68 \times N)$, où N est le nombre de trames dans l'image de mot.

3.1.3.3. Exemples de calcul des Histogrammes HMB

Dans cette section nous présentons des exemples de calcul de vecteurs de caractéristiques (HMB). Les histogrammes HMB sont extraient à partir d'une image marquée avec quatre directions (0°, 90°, 180°, 270°) et un rayon de 4 pixels.

La figure 3.16 montre un exemple d'extraction d'histogrammes. La figure 3.16 (a) montre une image divisée en quatre blocs horizontaux. La figure 3.16 (h) montre le HMB extrait de la frame (f1) sélectionnée de l'image marquée sur la figure 3.16 (a), ce HMB est obtenu en concaténant les HMBs extrait de chaque bloc de b1 à b4 de la frame f1. Le HMB de toute l'image est la concaténation des HMBs extraient de chaque frame en utilisant une fenêtre glissante survolant toute l'image.

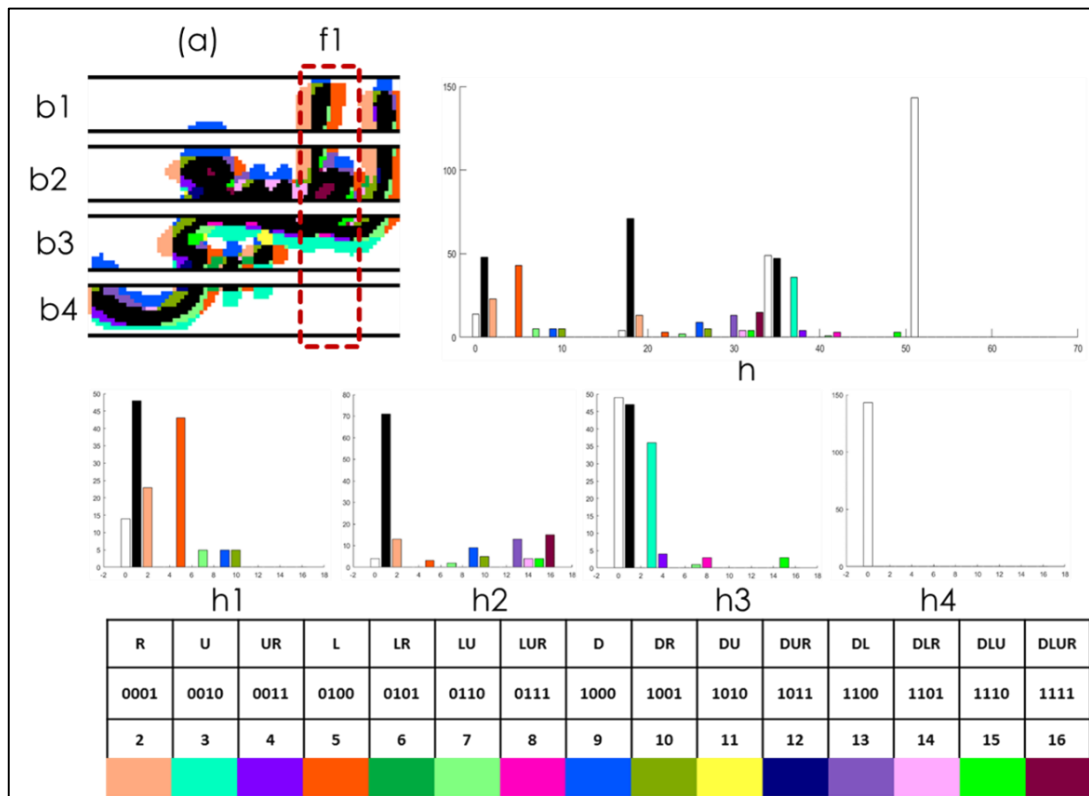


Figure 3.16 Calcul d'histogramme. a) Une image marquée divisée horizontalement en quatre blocs. h) Le HMB extrait de la frame f1 dans l'image marquée (a). h1), h2), h3) et h4) sont des HMBs extraits respectivement des blocs 1, 2, 3 et 4 de la frame f1.

La figure 3.17 (h1) montre le HMB extrait du bloc b1 dans la frame f1 de la figure 3.17 (a). Le haut niveau des marques R et L respectivement (2 et 5)

indique l'existence d'un trait vertical dans le bloc 1. Parce que ces marques correspondent respectivement aux directions droite et gauche. Le HMB dans la figure 3.17 (h2) contient des pixels avec la marque 16. Les pixels marqués avec cette marque ont des voisins d'encre dans toutes les directions, ce qui signifie que ces pixels sont entourés de pixels d'encre. Cela indique la présence d'une boucle dans le bloc 2.

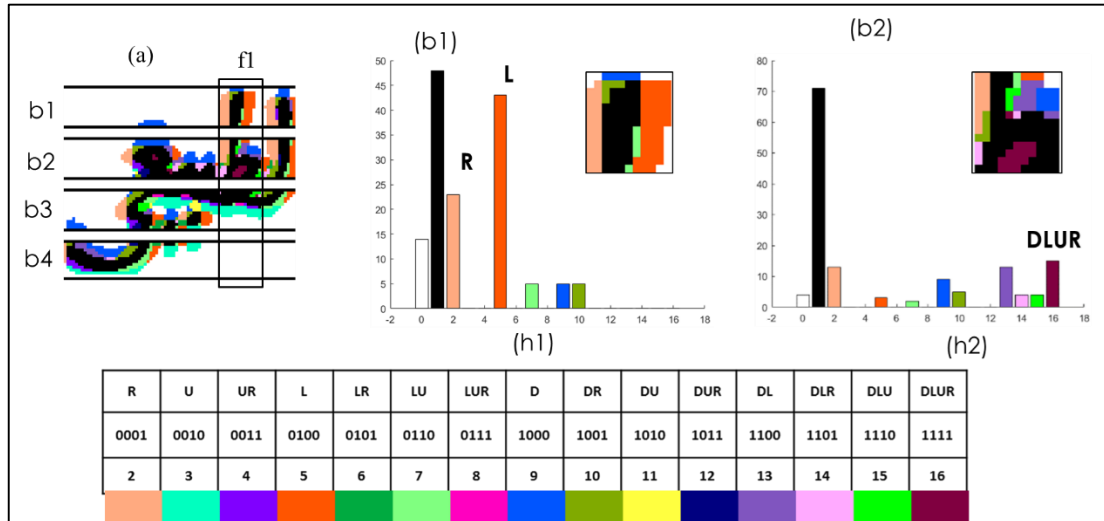


Figure 3.17. Calcul d'histogramme. a) Une image marquée divisée horizontalement en quatre blocs. h1), h2) Les HMBs extraient du bloc1 et bloc2 respectivement de la frame f1 dans l'image marquée (a

La figure 3.18 montre un autre exemple de calcul d'histogrammes HMBs. Le HMB de la figure 3.18 (h2) capture la présence d'une angulation (cornière) dans le bloc 2 de la frame2 vu que les marques les plus élevées dans sont HMB sont celles de R, RD et RU. Les pixels marqués avec ces marques ont des pixels d'encres voisins dans les directions suivantes : la droite, en haut à droite et en bas à droite. Le HMB dans la figure 3.18 (h4) montre l'existence d'une concavité (courbure) dans le bloc 4 de la frame 2. Cela est indiqué par les hautes valeurs des marques DR, LU et U. Ces marques indiquent qu'il existe des voisins d'encre dans les directions : haut, à gauche et haut, en bas à droite.

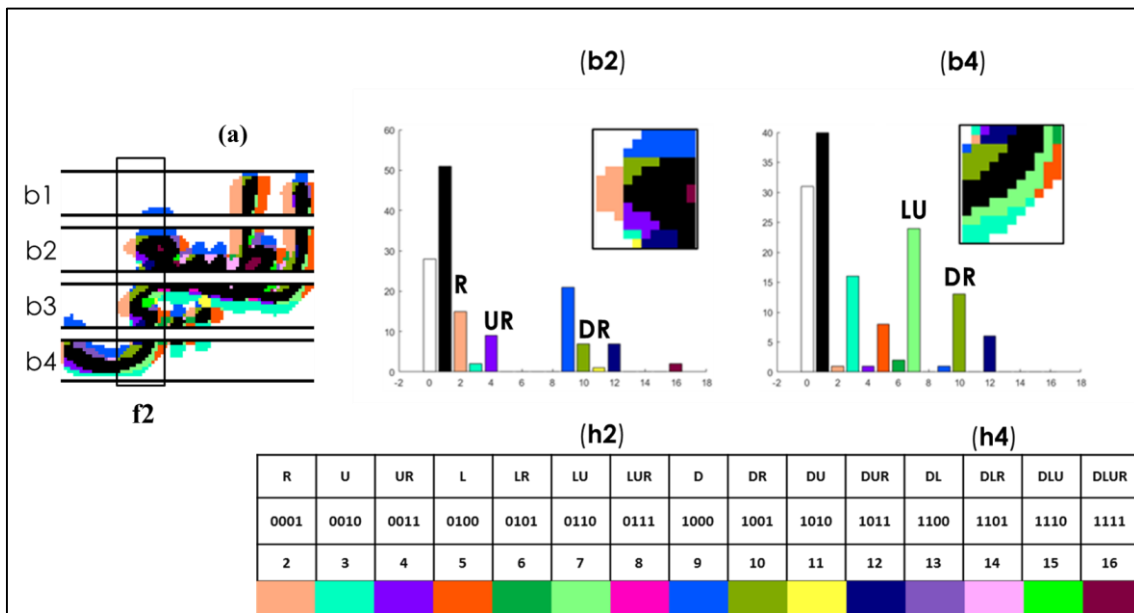


Figure 3.18 Calcul d'histogramme. a) Une image marquée divisée horizontalement en quatre blocs. h2), h4) Le HMB extrait du bloc 2 et bloc 4 respectivement de la frame f2 dans l'image marquée (a).

3.2. Avantages de l'attribut HMB

Les avantages que présente notre attribut proposé HMB peuvent être résumés en :

- Notre attribut proposé (HMB) améliore la reconnaissance grâce à sa capacité à capturer différentes caractéristiques de l'écriture arabe.
- Les histogrammes HMBs sont très simples à calculer.
- Les caractéristiques discriminantes sont embarquées dans l'image.
- Il peut capturer les caractéristiques de l'écriture arabe sans combinaison avec d'autres méthodes.

3.3. Conclusion

Dans ce chapitre nous avons présenté notre contribution à la reconnaissance de l'écriture arabe manuscrite en agissant sur la phase d'extraction d'attribut. Notre contribution s'agit de la proposition d'un nouvel attribut pour la reconnaissance de l'écriture arabe manuscrite appelé HMB (Histogram of Marked Background). L'extraction de cet attribut passe par deux étapes. La première étape, consiste en le marquage des fonds des images de texte en cherchant pour chaque pixel de fond d'éventuels pixels d'encre voisins dans une zone délimité par un rayon donné et dans des directions données. Le résultat de

cette étape est une image de texte marquée. La deuxième étape consiste en calcul des vecteurs d'attributs composés des histogrammes des marques des pixels de l'image marquée. Les tests de l'efficacité de cet attribut feront l'objet du chapitre suivant.

Expérimentations et résultats

Dans ce chapitre, nous évaluons l'efficacité de l'attribut HMB que nous avons proposé pour la reconnaissance de l'écriture arabe manuscrite. Pour cela, nous avons comparé notre méthode avec certaines méthodes rapportées dans la littérature. Ce chapitre présente plus de détails sur ces systèmes de référence, les outils utilisés pour la reconnaissance et les bases de données d'écriture arabe utilisées pour l'évaluation ainsi que les résultats obtenues et leurs discussions.

4.1. Bases de données

L'attribut HMB que nous avons proposé a été évaluée sur deux bases de données, la base de données IBN SINA [92] et la base de données IFHCDB [93]. Nous avons utilisé la première base pour tester notre attribut sur les documents historiques arabes. La seconde sert à tester l'efficacité de notre méthode sur les caractères arabes isolés.

4.1.1. IBN SINA

IBN SINA est une base de données de documents historiques arabes [92] contenant 50 pages manuscrites. Les images du document sont déjà binarisées, nettoyées et segmentées en composantes connexes (CC). Une CC peut être un caractère, un sous mot ou un mot. Chaque page contient environ 500 composantes connexes. La base de données contient environ 20722 CC arabes écrites dans le style Naskh et classés en environ 1000 classes différentes. La base de données IBN SINA est très contrainte où certaines classes ont jusqu'à 5 000 entrées, tandis que d'autres ont moins de cinq. La figure 4.1 montre des exemples de CC

binarisées de la base de données IBN SINA. La figure 4.2 montre un exemple d'une page de la base de données IBN SINA.

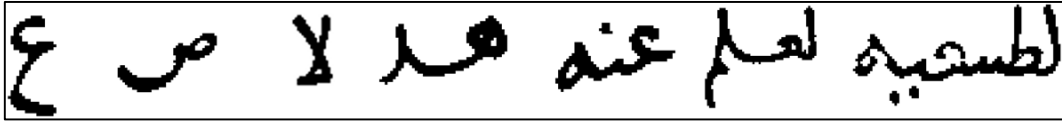


Figure 4.1. Exemples de composantes connexes de la base IBN SINA.

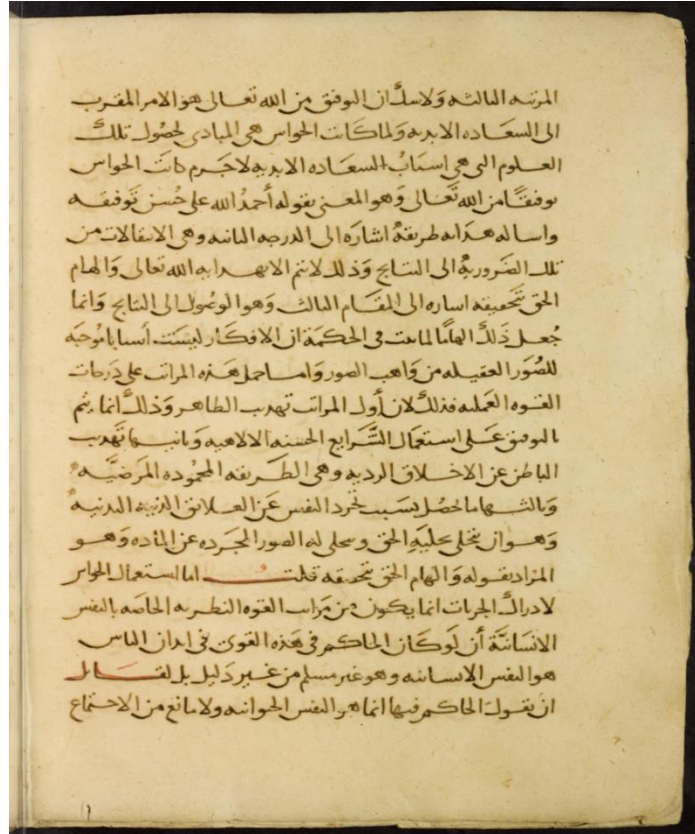


Figure 4.2 La Page 0008 de la base IBN SINA.

4.1.2. IFHCDB

La base de données IFHCDB (Isolated Farsi Handwritten Character Database) [93] est une base de caractères manuscrits farsi/arabes et de chiffres. Elle a été créée au Département de génie électrique de l'Université de technologie Amir Kabir (AUT), Téhéran, Iran en 2006. Cette base de données contient deux ensembles explicites : l'ensemble d'apprentissage (70%) et l'ensemble de tests (30%). L'IFHCDB comprend 52 380 caractères arabes farsi/arabes et 17 740 chiffres. Le nombre d'exemplaire dans chaque classe n'est pas uniforme.

La figure 4.3 montre quelques exemples de caractères arabes de la base de données IFHCDB.

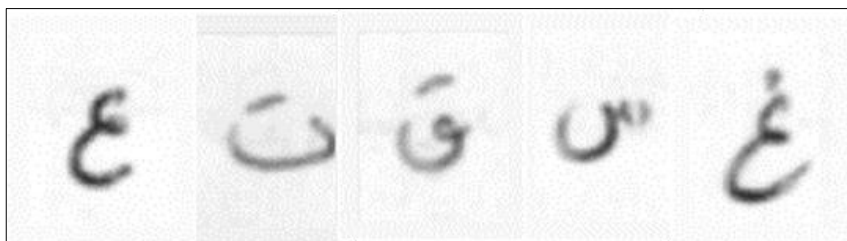


Figure 4.3. Exemples de caractères de la base de données IFHCDB.

4.2. Les systèmes de références

Afin de comparer l'attribut HMB proposé dans cette thèse avec les méthodes existantes, nous avons implémenté les attributs les plus utilisées pour la reconnaissance de l'écriture manuscrite arabe, à savoir les attributs de distributions, directionnels, structurels et de concavités. Ces attributs ont déjà fait l'objet d'une étude comparative réalisée par notre groupe et qui a été présentée dans la conférence ICCES 2017 [42]. Les détails de ces attributs sont décrits dans les sous-sections suivantes.

4.2.1. Préparations des données

Avant d'extraire les caractéristiques, les images en entrées sont divisées horizontalement en 4 blocs (figure 4.4) puis une fenêtre glissante est utilisée pour parcourir l'image de droite à gauche.

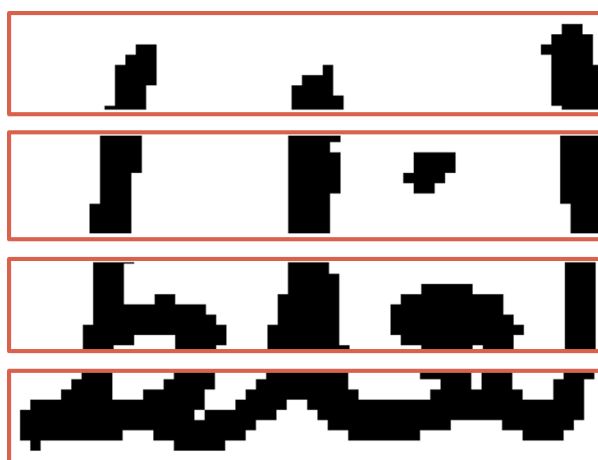


Figure 4.4. Exemple d'image divisée en 4 blocs.

4.2.2. Les attributs de distributions

Ce type d'attributs a été proposé par Marti et Bunke [94]. Neuf quantités géométriques sont calculées à partir des pixels d'encre des images binaires dans chaque frame de l'image pour déterminer les caractéristiques. Les trois premières caractéristiques donnent de l'information sur la distribution des pixels dans une région et comment ils sont répartis, ils s'agissent du nombre de pixels noirs (équation 10), du centre de gravité (équation 11) et du moment de second ordre (équation 12).

$$p(x, y) = \begin{cases} 1 & \text{si pixel noir} \\ 0 & \text{si pixel blanc} \end{cases} \quad (9)$$

$$f_1(x) = \sum_{y=1}^m p(x, y) \quad (10)$$

$$f_2(x) = \frac{1}{m} \sum_{y=1}^m y * p(x, y) \quad (11)$$

$$f_3(x) = \frac{1}{m^2} \sum_{y=1}^m y^2 * p(x, y) \quad (12)$$

Les caractéristiques restantes donnent plus de détails sur l'écriture. Les quatrième et cinquième caractéristiques donnent la position du contour supérieur et inférieur, les sixième, septième, huitième et neuvième caractéristiques donnent respectivement la fraction de pixels de d'encre entre les contours supérieur et inférieur, le nombre de transitions entre les pixels d'encres et de fond de l'image et le gradient du contour supérieur et inférieur par rapport à la frame précédente, ce qui fournit des informations dynamiques. Pour extraire une séquence de caractéristiques des images de texte, les images de textes sont d'abord binarisées puis divisées en quatre blocs horizontaux ce qui permettra la capture de la distribution spatiale de l'information, puis une fenêtre glissante de droite à gauche est utilisé et ainsi les caractéristiques sont extraites dans chaque frame.

4.2.3. Les attributs directionnels

Le deuxième type de caractéristiques a été proposé pour la première fois par Favata et al [95] et utilisé pour l'écritures arabe manuscrite par [63]. Une fois l'image divisée horizontalement en quatre blocs, le gradient est extrait pour chaque segment de texte. Les caractéristiques du gradient sont calculées en utilisant la convolution de deux opérateurs de Sobel 3x3 avec l'image binaire. Le

gradient d'un pixel central est calculé en fonction de ses huit plus proches voisins. Dans ce travail, la direction du gradient est utilisée dans le calcul d'un vecteur de caractéristiques de la carte de caractéristiques du gradient. La direction est divisée en 12 régions non chevauchantes ($1^\circ-30^\circ$, $31^\circ-60^\circ \dots 331^\circ-360^\circ$). Dans chaque région d'échantillonnage, un histogramme des directions de gradient est estimé, résultant en 12 caractéristiques de gradient pour chaque segment d'image. Cela correspond au nombre de chaque direction de gradient dans la région. Ces compteurs sont concaténés pour donner les caractéristiques de gradient d'une image de texte.

4.2.4. Les attributs structurels

Le troisième type de caractéristiques (structurelles) proposé par Favata et al [95] et utilisé pour l'écriture arabe manuscrite par [63]. Il capture les « mini-traités » de l'image. Un ensemble de 12 règles est appliqué à chaque pixel. Chaque règle examine un motif particulier des pixels voisins pour les plages de gradient autorisées. La liste complète des règles et les définitions de voisinage sont définies dans [23]. Ces caractéristiques représentent le vecteur de caractéristiques structurelles. Le vecteur de caractéristiques contient 12 types de caractéristiques structurelles selon les règles définies par les auteurs.

4.2.5. Les attributs de concavités

La quatrième caractéristique (concavité) est définie dans [52]. L'attribut de concavité est utilisé pour calculer la distance moyenne en balayant pixel par pixel depuis la bordure de l'image jusqu'au premier trait rencontré (figure 4.5). Il existe quatre caractéristiques de concavité. Soit r_i , l_i , u_i et d_i respectivement la distance entre les bords de cellule droit, gauche, haut et bas et le premier trait rencontré dans les directions gauche, droite, haut et bas. Ensuite les équations (13), (14), (15) et (16) peuvent calculer les quatre caractéristiques de concavité respectivement.

$$f_4 = \frac{\sum_{i=1}^w d_i}{w} \quad (13)$$

$$f_5 = \frac{\sum_{i=1}^w u_i}{w} \quad (14)$$

$$f_6 = \frac{\sum_{i=1}^h r_i}{h} \quad (15)$$

$$f_7 = \frac{\sum_{i=1}^h l_i}{h} \quad (16)$$

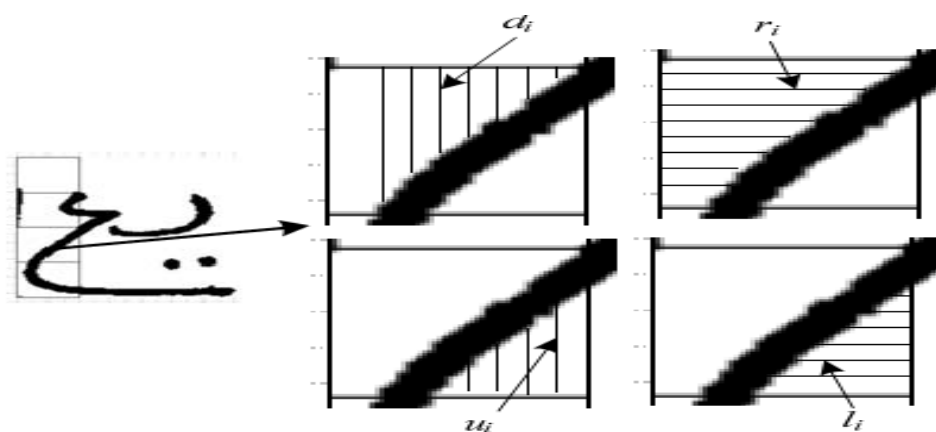


Figure 4.5. Les quatre types d'attributs de concavités

4.3. Outil de reconnaissance

Le processus de reconnaissance est basé sur les modèles de Markov cachés (HMM) [96], avec la boîte à outils HTK(Hidden Markov model toolkit) [97]. L'avantage dans l'utilisation des HMMs comme méthode de reconnaissance est qu'ils évitent d'avoir recours à la segmentation des images de texte en très petites unités, et ils peuvent traiter un grand vocabulaire vu qu'ils peuvent reconnaître des mots en utilisant une combinaison des modèles de caractères.

Le HTK est une boîte à outils portable pour la création et la manipulation des modèles de Markov cachés.

Les outils HTK sont divisés en quatre types : outils de préparation des données, outils d'apprentissage, outils de reconnaissance et outils d'analyse des résultats. Avec HTK, toute topologie de HMM peut être construite à l'aide de simples fichiers texte.

Les outils d'apprentissage de HTK ajustent les paramètres des HMM en utilisant une version d'apprentissage intégrée de l'algorithme de Baum-Welch pour la maximisation de la probabilité de vraisemblance des données d'entraînement [97].

Dans la phase de reconnaissance, l'algorithme de Viterbi implémenté dans l'outil de reconnaissance du HTK est utilisé. L'outil de reconnaissance prend en entrée le vecteur d'attributs extraits, un réseau décrivant les séquences de mots admissibles, un dictionnaire définissant chaque mot et un ensemble de HMMs [97].

L'outil d'analyse des résultats HTK évalue les performances du système de reconnaissance. Cela est effectué en faisant correspondre les données de sortie du dispositif de reconnaissance avec la transcription de référence d'origine. Le nombre de substitutions (S), de délétions (D) et d'insertions (I) sont comptabilisé pour ensuite rapporté le taux de reconnaissance [97].

4.4. Configuration des paramètres du système

Dans cette étude chaque caractère est défini par un HMM .Le modèle de reconnaissance est basé sur l'apprentissage embarqué des HMM (Embedded training) où la reconnaissance des mots se fait en concaténant des modèles de HMM de caractères. Lors de l'utilisation de la base de données IBN SINA, 31 HMM de caractères sont pris en compte : les 28 caractères de l'alphabet arabe et 3 types de signes diacritiques. Lors de l'utilisation de la base de données IFHCDB, 28 HMM de caractères ont été pris en compte. La topologie du modèle HMM des caractères est de trois états gauche-droite (voir figure 4.6). Dans chaque état, la probabilité de densité de l'observation a été modélisée comme une mixture de 60 gaussiennes.

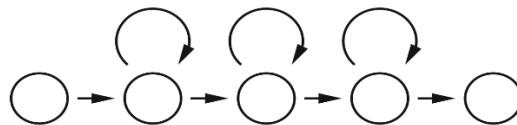


Figure 4.6 Topologie des caractères HMMs utilisée.

Nous avons réalisé nos expériences pour différentes valeurs des paramètres d'extraction des attributs. Cela comprenait la largeur de la frame et les chevauchements utilisés par la fenêtre glissante, les directions \mathbf{d} et le rayon \mathbf{r} qui sont déployés dans le processus de marquage du fond de l'image du texte.

Pour la répartition des échantillons d'apprentissage et de test, 67 % de la base de données IBN SINA été utilisée pour l'apprentissage et 33 % pour les tests. La base de données IFHCDB contient des caractères farsi isolés, des caractères arabes et des chiffres. De cette base nous n'avons utilisé que des caractères arabes. La base de données IFHCDB est divisée en deux parties distinctes, l'ensemble d'apprentissage et l'ensemble de test. Nous avons utilisé ces ensembles explicites dans nos expériences.

Afin de mesurer l'efficacité de l'attribut proposée et de la comparer avec d'autres méthodes, nous avons utilisé la métrique la plus utilisée dans la littérature [29]. La mesure d'évaluation est basée sur le taux de reconnaissance. Elle est définie comme la précision du système dans la tâche de reconnaissance. Le taux de reconnaissance est défini comme dans l'Equation. (17).

$$\text{Taux de reconnaissance} = \frac{\text{nombre de formes bien classées}}{\text{le nombre total des formes}} \quad (17)$$

4.5. Discussions des résultats

Afin d'évaluer notre attribut sur les deux bases de données (IBN SINA et IFHCDB), et d'avoir une étude comparative avec les mêmes paramètres, nous avons implémenté les méthodes d'extraction d'attributs décrites dans la Section 4.2, et comparé les résultats obtenus à notre méthode d'extraction de caractéristiques proposée (HMB).

Le tableau 4.1 présente le taux de reconnaissance de notre méthode proposée (HMB) et celles des méthodes implémentées (directionnelle, structurelle, concavité et distribution) avec les deux bases de données (IBN SINA et base de données IFHCDB).

Attributs	Taux de reconnaissance avec IBN SINA	Taux de reconnaissance avec IFHCDB
L'attribut proposé (HMB)	72.29%	91.76%
Attribut directionnel	67.13%	91.27%
Attribut structurel	57.81%	88.34%
Attribut de concavité	56.90%	89.10%
Attributs de distribution	38.01%	82.77%

Tableau 4.1. Comparaison des résultats de l'attribut proposé à ceux des méthodes implémentées en utilisant la base de données IBN SINA et la base de données IFHCDB.

Les résultats du tableau 4.1 montrent que notre attribut est compétitif par rapport aux autres méthodes implémentées avec les deux bases de données. Bien que, la base de données IBN SINA est un document historique de mauvaise qualité, notre méthode a produit des résultats compétitifs et a montré sa capacité à traiter les documents historiques arabes.

Notre attribut proposé (HMB) améliore la reconnaissance grâce à sa capacité à capturer différentes caractéristiques de l'écriture arabe. Il peut capturer des caractéristiques structurelles telles que la présence de boucle, d'ascendeur, de descendeur et autres. Il peut également capturer les caractéristiques statistiques telles que la densité des pixels d'encre. Il peut capturer la direction du trait comme (ligne verticale, ligne horizontale, coins ainsi que la présence de concavités. Par conséquent, il peut capturer ces caractéristiques sans combinaison avec d'autres méthodes.

Dans le tableau 4.2, nous montrons l'efficacité de notre attribut dans la reconnaissance des caractères isolés. Pour cela, nous avons utilisé l'IFHCDB et les caractères isolés (CC composés d'un caractère) de la base de données IBN SINA. Les résultats obtenus montrent que les taux de reconnaissance avec les deux bases de données sont satisfaisants. Avec la base de données IBN SINA, le taux de reconnaissance est supérieur par rapport à celui de la base de données IFHCDB. Cela peut être justifié par la différence du nombre d'échantillons dans chaque base de données. Le nombre de caractères isolés utilisés avec la base de données IBN SINA est inférieur à celui de l'IFHCDB, par conséquent le nombre de caractères mal classés avec la base de données IBN SINA est réduit.

Attribut	Base de données	Taux de reconnaissance
L'attribut proposé (HMB)	IBN SINA	93.32%
	IFHCDB	91.76%

Tableau 4.2. Les résultats de l'attribut proposé dans cette étude en utilisant la base de données IFHCDB et des caractères isolés de la base de données IBN SINA.

Comme mentionné dans la Section 4.4, les caractéristiques de notre méthode proposée ont été extraites à l'aide des paramètres suivants : le rayon r et les directions d prises en compte dans le processus de marquage de fond, la largeur de la frame (de la fenêtre glissante) et le chevauchement entre les frames utilisés à l'étape de calcul des caractéristiques. Ces valeurs sont déterminées empiriquement.

Le tableau 4.3 montre les taux de reconnaissance obtenues avec notre attribut (HMB) testé sur les bases de données IBN SINA et IFHCDB en utilisant un rayon de 4 pixels, les directions (0° , 90° , 180° , 270°) et différentes valeurs de largeur de la frame et de chevauchements entre les frames. Nous avons pris les valeurs de la largeur de la frame et de leurs chevauchement en prenant en considération la taille de l'image du mot ou du caractère. Les résultats montrent qu'il n'y a pas de différence significative dans les taux de reconnaissance obtenus pour les différentes valeurs de largeur de trame et de chevauchements. Par conséquent, le choix de ces paramètres n'affecte pas le taux de reconnaissance. Cependant, les résultats ont montré que la meilleure précision était obtenue par une largeur de trame de 3 pixels et un chevauchement de 2 pixels. Par conséquent, les résultats rapportés dans cette section étaient avec une largeur de frame de 3 pixels et un chevauchement de 2 pixels.

Attribut	Frame	Chevauchement	Taux de reconnaissance avec IBN SINA	Taux de reconnaissance avec IFHCDB
Attribut proposé (HMB)	2	1	71.74%	90.22%
	3	1	70.61%	90.63%
	3	2	72.29%	91.76%

Tableau 4.3. Les taux de reconnaissance avec l'attribut proposé avec un rayon de 4 pixels et différentes combinaisons de frames et chevauchement.

Les tableaux 4.4 et 4.5 montrent les taux de reconnaissances obtenues en utilisant différentes valeurs de rayon r et différentes configurations de directions d , utilisées respectivement avec la base de données IBN SINA et la base de données IFHCDB. Les résultats ont montré qu'avec les deux bases de données, le rayon n'influence pas beaucoup le taux de reconnaissance. Néanmoins, les résultats ont montré que le meilleur taux de reconnaissance est obtenu avec un rayon de 4 pixels. Les résultats montrent également que le taux de reconnaissance dépend du nombre de directions utilisées avec les deux bases de données. La différence entre les taux de reconnaissance est insignifiante lorsqu'on utilise quatre directions, avec les deux configurations $(0^\circ, 90^\circ, 180^\circ, 270^\circ)$ et $(45^\circ, 135^\circ, 225^\circ, 315^\circ)$. Cependant, les résultats avec 8 directions $(0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ)$ sont très faibles par rapport aux deux configurations à quatre directions. Ces faibles résultats peuvent être justifiés par le fait que les informations sont redondantes ce qui provoque une surcharge de données non importantes. La taille du vecteur de caractéristique utilisant huit directions (1028) est supérieure à celle utilisant quatre directions (68). Enfin, la combinaison du rayon 4 pixels et de la configuration à quatre directions donne les meilleurs résultats.

r	2	3	4	5	6	8
d						
(0°, 90°, 180°, 270°)	67.28%	71.93%	72.29%	70.98%	70.68%	64.90%
(45°, 135°, 225°, 315°)	65.14%	70.36	71.52%	70.85%	70.05%	69.84%
(0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°)	9.12%	15.85%	21.61%	19.71%	17.56%	13.76%

Tableau 4.4. Taux reconnaissance de l'attribut proposé avec différentes valeurs de rayon et différentes configurations de directions en utilisant la base de données IBN SINA.

r	2	3	4	5	6	8
d						
(0°, 90°, 180°, 270°)	90.51%	91.62%	91.54%	89.49%	88.38%	84.85%
(45°, 135°, 225°, 315°)	91.35%	91.58%	91.76%	91.73%	91.22%	90.25%
(0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°)	20.59%	28.83%	29.41%	29.12%	28.98%	27.84%

Tableau 4.5. Taux reconnaissance de l'attribut proposé avec différentes valeurs de rayon et différentes configurations de directions en utilisant la base de données IFHCDB.

4.6. Conclusion

Dans ce chapitre nous avons présenté les bases de données utilisées dans le test de notre attribut HMB, les systèmes de références utilisées pour la comparaison des résultats de test de l'attribut HMB, l'outil de reconnaissance utilisé a savoirs le Toolkit HTK qui implémente les HMMs, la topologie des HMMs utilisé et nous avons terminé par les résultats obtenus lors de la phase de reconnaissance ainsi que leurs discussions. Dans l'ensemble et à partir des résultats présentés dans ce chapitre, la méthode proposée a montré son efficacité et sa

fiabilité dans la reconnaissance des mots arabes, des sous-mots et des caractères isolés, et sa capacité à traiter les documents historiques arabes.

Conclusion Générale

La reconnaissance automatique de l'écriture (OCR) vise à traduire un texte imprimé ou manuscrit en un texte codé numériquement compréhensible par une machine. Les OCRs sont utilisés dans plusieurs applications tel que le tri automatique du courrier, la lecture de montant de chèques bancaire, la lecture de documents historique et etc. Bien que plusieurs travaux ont été réalisés dans le domaine de la reconnaissance de l'écriture manuscrite en particulier l'écriture arabe manuscrite, il reste toujours l'un des problèmes les plus difficiles dans le domaine de reconnaissance de forme. Cette difficulté est liée à la complexité morphologique de l'écriture arabe (cursivité, contient des loupes, point diacritique, même lettre peut avoir plusieurs formes, etc.) et aux grandes variations des styles d'écriture ainsi que la dégradation des documents manuscrits.

Dans cette thèse nous avons traité la problématique de reconnaissance de l'écriture arabe manuscrite en agissant sur la phase d'extraction d'attributs. Notre but est de proposer une nouvelle méthode d'extraction d'attribut qui prend en considération les spécificités (caractéristiques) de l'écriture arabe manuscrite.

Après avoir fait une étude sur les systèmes de reconnaissance de l'écriture arabe manuscrite il s'est avéré que la phase d'extraction d'attributs est une étape cruciale dans les systèmes de reconnaissance de l'écriture. Cela nous a mené à faire un état de l'art sur les méthodes d'extraction d'attributs dans les systèmes de reconnaissance de l'écriture arabe manuscrite. Ensuite, à faire une étude comparatives entres quatre méthodes d'extraction d'attributs implémentées et testées sur une base d'images de document historique arabe. Les résultats de cette étude ont montré le besoin d'une nouvelle méthode de représentation de l'écriture manuscrite notamment dans les anciens manuscrits arabe.

Par conséquent, en analysant les méthodes d'extraction d'attributs présentées dans la littérature, nous avons constaté que les auteurs se concentraient sur l'analyse des pixels d'encre pour extraire leurs attributs. Alors qu'une image de texte est constituée de pixels d'encre ayant une certaine distribution spatiale dans le fond (arrière-plan) de l'image. La distribution spatiale de pixels d'encres dans le fond de l'image de texte renseigne sur la forme de l'écriture. Dans une image de texte, les pixels de fond (arrière-plan) sont également importants car ils nous indiquent les positions et les dispersions des pixels d'encre dans l'image. En plus, les pixels de fond peuvent aussi être utilisés comme support pour insérer des informations supplémentaires dans l'image. D'où l'idée de notre contribution qui est la proposition d'un nouvel attribut pour la reconnaissance de l'écriture arabe manuscrite nommé HMB (Histogram of Marked Background), que nous avons présenté dans le chapitre 3.

La nouveauté avec notre contribution (HMB) est que, en plus des pixels d'encre nous utilisons également le fond de l'image du texte pour extraire les caractéristiques. L'idée principale de notre contribution est d'enrichir l'image du texte pour qu'elle soit plus représentative en intégrant des informations contextuelles dans le fond des images de textes. Le principe générale de l'attribut HMB est d'injecter des informations discriminantes dans l'image du texte et cela en attribuant des marques aux pixels de fond des images de textes. Les pixels de fond sont marqués selon la répartition spatiale des pixels d'encre dans leur voisinage délimité par un rayon donné suivant des directions données, puis des histogrammes quantifiant ces marques constituent les vecteurs d'attributs. Par conséquent, ces marques permettent de capturer la forme et l'orientation de l'écriture (lignes verticales, lignes horizontales, boucles, points, courbure, et autres). L'attribut HMB permet ainsi de capturer les caractéristiques complexes de l'écriture arabe manuscrite. Cet attribut est testé sur deux bases de données IBN SINA et IFHCDB et comparé à quatre types d'attributs présentés dans la littérature que nous avons implémentés. Les résultats de tests ont montré l'efficacité de HMB par rapport aux autres attributs que ce soit pour les caractères isolés arabes ou les anciens manuscrits arabes.

Comme perspectives nous suggérons :

- Une extension de notre architecture peut être proposée afin qu'elle soit capable de gérer l'écriture manuscrite dans différentes langues (Amazigh, Latin, etc.)
- De tester notre méthode avec d'autres méthodes de reconnaissance telle qu'Adaptive Locality Preserving Regression et SVM.
- Adapter notre approche de sorte à l'utiliser avec les méthodes de Deep learning.

Bibliographie

1. Nazif, A., *A system for the recognition of the printed Arabic characters*. Master's Thesis (2nd Edition) Faculty of Engineering, Cairo University, 1975.
2. Woods, R.E. and R.C. Gonzalez, *Digital Image Processing Third Edition*. 2021.
3. Boukhlof, D., *Résolution de problèmes par écosystèmes: Application au traitement d'images*. 2005, Université Mohamed Khider-Biskra.
4. Zeki, A.M. and M.S. Zakaria. *Challenges in recognizing Arabic characters*. in *The national conference for computer*. Abu-al-Aziz king University. Arabia Saudi. April 2004. 2004.
5. Tensmeyer, C. and T. Martinez, *Historical document image binarization: A review*. SN Computer Science, 2020. **1**: p. 1-26.
6. Otsu, N., *A threshold selection method from gray-level histograms*. IEEE transactions on systems, man, and cybernetics, 1979. **9**(1): p. 62-66.
7. Moghaddam, R.F. and M. Cheriet, *AdOtsu: An adaptive and parameterless generalization of Otsu's method for document image binarization*. Pattern Recognition, 2012. **45**(6): p. 2419-2431.
8. Canny, J., *A computational approach to edge detection*. IEEE Transactions on pattern analysis and machine intelligence, 1986(6): p. 679-698.
9. Nafchi, H.Z., R.F. Moghaddam, and M. Cheriet, *Phase-based binarization of ancient document images: Model and applications*. IEEE transactions on image processing, 2014. **23**(7): p. 2916-2930.
10. Pastor-Pellicer, J., et al. *Insights on the use of convolutional neural networks for document image binarization*. in *International Work-Conference on Artificial Neural Networks*. 2015. Springer.
11. Lelore, T. and F. Bouchara, *Fair: a fast algorithm for document image restoration*. IEEE transactions on pattern analysis and machine intelligence, 2013. **35**(8): p. 2039-2048.
12. Amin, A. and G. Masini. *Machine recognition of multifold printed Arabic texts*. in *Proc. 8th Int. Conf. on Pattern Recognition*. 1986.
13. Parhami, B. and M. Taraghi, *Automatic recognition of printed Farsi texts*. Pattern Recognition, 1981. **14**(1-6): p. 395-403.
14. Timsari, B. and H. Fahimi. *Morphological approach to character recognition in machine-printed Persian words*. in *Document Recognition III*. 1996. International Society for Optics and Photonics.

15. Nouh, A., A. Nurul-Ula, and A. Sharaf Eldin. *A proposed algorithm for thinning binary Arabic character patterns*. in *First Kuwait Computer Conference*. 1989.
16. Naccache, N.J. and R. Shinghal, *SPTA: A proposed algorithm for thinning binary patterns*. IEEE transactions on systems, man, and cybernetics, 1984(3): p. 409-418.
17. Lam, L., S.-W. Lee, and C.Y. Suen, *Thinning methodologies-a comprehensive survey*. IEEE Transactions on pattern analysis and machine intelligence, 1992. **14**(9): p. 869-885.
18. Atallah, A.-S. and K. Omar. *A comparative study between methods of Arabic baseline detection*. in *2009 International Conference on Electrical Engineering and Informatics*. 2009. IEEE.
19. Steinherz, T., E. Rivlin, and N. Intrator, *Offline cursive script word recognition—a survey*. International Journal on Document Analysis and Recognition, 1999. **2**(2-3): p. 90-110.
20. Mhiri, M., *Apprentissage profond de formes manuscrites pour la reconnaissance et le repérage efficace de l'écriture dans les documents numérisés*. 2018, École de technologie supérieure.
21. Alginahi, Y.M., *A survey on Arabic character segmentation*. International Journal on Document Analysis and Recognition (IJDAR), 2013. **16**(2): p. 105-126.
22. Zeki, A.M. *The segmentation problem in arabic character recognition the state of the art*. in *2005 International Conference on Information and Communication Technologies*. 2005. IEEE.
23. Mohammad, K., et al., *Contour-based character segmentation for printed Arabic text with diacritics*. Journal of Electronic Imaging, 2019. **28**(4): p. 043030.
24. Bushofa, B. and M. Spann. *Segmentation of Arabic characters using their contour information*. in *Proceedings of 13th International Conference on Digital Signal Processing*. 1997. IEEE.
25. Romeo-Pakker, K., H. Miled, and Y. Lecourtier. *A new approach for Latin/Arabic character segmentation*. in *Proceedings of 3rd International Conference on Document Analysis and Recognition*. 1995. IEEE.
26. Tellache, M., M. Sid-Ahmed, and B. Abaza. *Thinning algorithms for Arabic OCR*. in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*. 1993. IEEE.
27. Altuwajjri, M.M. and M.A. Bayoumi, *A thinning algorithm for Arabic characters using ART2 neural network*. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 1998. **45**(2): p. 260-264.

28. Hamid, A. and R. Haraty. *A neuro-heuristic approach for segmenting handwritten Arabic text*. in *Proceedings ACS/IEEE international conference on computer systems and applications*. 2001. IEEE.
29. Parvez, M.T. and S.A. Mahmoud, *Offline Arabic handwritten text recognition: a survey*. ACM Computing Surveys (CSUR), 2013. **45**(2): p. 1-35.
30. Al-wajih, E. and R. Ghazali, *An enhanced LBP-based technique with various size of sliding window approach for handwritten Arabic digit recognition*. Multimedia Tools and Applications, 2021: p. 1-20.
31. Hassan, A.K.A., B.S. Mahdi, and A.A. Mohammed, *Arabic handwriting word recognition based on scale invariant feature transform and support vector machine*. Iraqi Journal of Science, 2019: p. 381-387.
32. Tavoli, R., M. Keyvanpour, and S. Mozaffari, *Statistical geometric components of straight lines (SGCSL) feature extraction method for offline Arabic/Persian handwritten words recognition*. IET Image Processing, 2018. **12**(9): p. 1606-1616.
33. Khorashadizadeh, S. and A. Latif, *Arabic/Farsi Handwritten Digit Recognition usin Histogram of Oriented Gradient and Chain Code Histogram*. International Arab Journal of Information Technology (IAJIT), 2016. **13**(4).
34. Tanvir Parvez, M. and S.A. Mahmoud, *Arabic handwriting recognition using structural and syntactic pattern attributes*. Pattern Recognition, 2013. **46**(1): p. 141-154.
35. Abdel Azeem, S. and H. Ahmed. *Off-Line Arabic Handwriting Recognition system based on concavity features and HMM classifier*. in *21st International Conference on Pattern Recognition (ICPR 2012)*. 2012. Tsukuba, Japan.
36. Lamtougui, H., et al. *Offline Arabic Handwriting Recognition Using Deep Learning: Comparative Study*. in *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)*. 2020. IEEE.
37. Korichi, A., et al. *Arabic handwriting recognition: Between handcrafted methods and deep learning techniques*. in *2020 21st International Arab Conference on Information Technology (ACIT)*. 2020. IEEE.
38. Altwaijry, N. and I. Al-Turaiki, *Arabic handwriting recognition system using convolutional neural network*. Neural Computing and Applications, 2020: p. 1-13.
39. Ghadhban, H.Q., et al. *Survey of Offline Arabic Handwriting Word Recognition*. in *International Conference on Soft Computing and Data Mining*. 2020. Springer.

40. Heutte, L., *Reconnaissance de caractères manuscrits: application à la lecture automatique des chèques et des enveloppes postales*. 1994, Rouen.
41. Charfi, M., et al., *A new approach for arabic handwritten postal addresses recognition*. arXiv preprint arXiv:1204.1678, 2012.
42. Gagaoua, M., et al. *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.
43. Hassan, A.K.A., *Arabic (Indian) Handwritten Digits Recognition Using Multi feature and KNN Classifier*. Journal of University of Babylon for Pure and Applied Sciences, 2018. **26**(4): p. 10-17.
44. Mahmoud, S.A. and W.G. Al-Khatib, *Recognition of Arabic (Indian) bank check digits using log-gabor filters*. Applied Intelligence, 2011. **35**(3): p. 445-456.
45. Vapnik, V.N., *An overview of statistical learning theory*. IEEE transactions on neural networks, 1999. **10**(5): p. 988-999.
46. Cortes, C. and V. Vapnik, *Support-vector networks*. Machine learning, 1995. **20**(3): p. 273-297.
47. Elleuch, M., R. Maalej, and M. Kherallah, *A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition*. Procedia Computer Science, 2016. **80**: p. 1712-1723.
48. Elleuch, M., A. Hani, and M. Kherallah, *Arabic handwritten script recognition system based on HOG and gabor features*. Int. Arab J. Inf. Technol., 2017. **14**(4A): p. 639-646.
49. Fink, G.A., *Markov models for pattern recognition: from theory to applications*. 2014: Springer Science & Business Media.
50. Rabi, M., M. Amrouch, and Z. Mahani, *Recognition of Cursive Arabic Handwritten Text Using Embedded Training Based on Hidden Markov Models*. International Journal of Pattern Recognition and Artificial Intelligence, 2018. **32**(01): p. 1860007.
51. Alsharif, O. and J. Pineau, *End-to-end text recognition with hybrid HMM maxout models*. arXiv preprint arXiv:1310.1811, 2013.
52. Xiang, D., et al. *Offline Arabic handwriting recognition system based on HMM*. in *2010 3rd International Conference on Computer Science and Information Technology*. 2010. IEEE.
53. Hamdani, M., et al. *Combining Multiple HMMs Using On-line and Off-line Features for Off-line Arabic Handwriting Recognition*. in *2009 10th International Conference on Document Analysis and Recognition*. 2009.
54. Bunke, H., S. Bengio, and A. Vinciarelli, *Offline recognition of unconstrained handwritten texts using HMMs and statistical language*

- models*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004. **26**(6): p. 709-720.
55. Dehghan, M., et al., *Handwritten Farsi (Arabic) word recognition: a holistic approach using discrete HMM*. Pattern Recognition, 2001. **34**(5): p. 1057-1065.
56. Graves, A., et al. *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks*. in *Proceedings of the 23rd international conference on Machine learning*. 2006.
57. Maalej, R. and M. Kherallah. *Convolutional neural network and BLSTM for offline Arabic handwriting recognition*. in *2018 International Arab conference on information technology (ACIT)*. 2018. IEEE.
58. Graves, A., *Offline arabic handwriting recognition with multidimensional recurrent neural networks*, in *Guide to OCR for Arabic scripts*. 2012, Springer. p. 297-313.
59. Chen, L., et al. *Multi-layer recurrent neural network based offline Arabic handwriting recognition*. in *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*. 2017. IEEE.
60. Graves, A., et al., *A novel connectionist system for unconstrained handwriting recognition*. IEEE transactions on pattern analysis and machine intelligence, 2008. **31**(5): p. 855-868.
61. Doush, I.A., F. Alkhateeb, and A.H. Gharaibeh, *A novel Arabic OCR post-processing using rule-based and word context techniques*. International Journal on Document Analysis and Recognition (IJDAR), 2018. **21**(1): p. 77-89.
62. Giotis, A.P., et al., *A survey of document image word spotting techniques*. Pattern recognition, 2017. **68**: p. 310-332.
63. Mahmoud, S.A. and S.M. Awaida, *Recognition of off-line handwritten Arabic (Indian) numerals using multi-scale features and support vector machines vs. hidden markov models*. The Arabian Journal for Science and Engineering, 2009. **34**(2B): p. 429-444.
64. Saleem, S., et al. *Improvements in BBN's HMM-Based Offline Arabic Handwriting Recognition System*. in *2009 10th International Conference on Document Analysis and Recognition*. 2009.
65. John , T.F. and S. Geetha, *A Multiple Feature/Resolution Approach to Handprinted Digit and Character Recognition*. International Journal of Imaging Systems and Technology, 1996. **7**: p. 304-311.
66. Cheriet, M., et al., *Character recognition systems: a guide for students and practitioners*. 2007: John Wiley & Sons.

67. Yasuda, M., *An improvement of correlation method for character recognition*. IEICE Trans., 1979. **62**(3): p. 217.
68. Yamashita, Y., et al., *Classification of handprinted Kanji characters by the structured segment matching method*. Pattern Recognition Letters, 1983. **1**(5-6): p. 475-479.
69. Abdleazeem, S. and E. El-Sherif, *Arabic handwritten digit recognition*. International Journal of Document Analysis and Recognition (IJDAR), 2008. **11**(3): p. 127-141.
70. Alaei, A., P. Nagabhushan, and U. Pal. *Fine Classification of Unconstrained Handwritten Persian/Arabic Numerals by Removing Confusion amongst Similar Classes*. in *2009 10th International Conference on Document Analysis and Recognition*. 2009.
71. Jebri, N.A., H.R. Al-Zoubi, and Q.A. Al-Haija, *Recognition of handwritten arabic characters using histograms of oriented gradient (HOG)*. Pattern Recognition and Image Analysis, 2018. **28**(2): p. 321-345.
72. Nooraliei, A. *Persian handwritten digits recognition by using zoning and histogram projection*. in *2013 3rd Joint Conference of AI & Robotics and 5th RoboCup Iran Open International Symposium*. 2013. IEEE.
73. Mioulet, L., *Reconnaissance de l'écriture manuscrite avec des réseaux récurrents*. 2015, Université de rouen.
74. Koerich, A.L. *Unconstrained handwritten character recognition using different classification strategies*. in *International Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR)*. 2003.
75. Wang, L. and D.-C. He, *Texture classification using texture spectrum*. Pattern recognition, 1990. **23**(8): p. 905-910.
76. Ojala, T., M. Pietikäinen, and D. Harwood, *A comparative study of texture measures with classification based on featured distributions*. Pattern recognition, 1996. **29**(1): p. 51-59.
77. Ojala, T., M. Pietikäinen, and T. Maenpää, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*. IEEE Transactions on pattern analysis and machine intelligence, 2002. **24**(7): p. 971-987.
78. Boulid, Y., A. Souhar, and M.Y. Elkettani, *Handwritten Character Recognition Based on the Specificity and the Singularity of the Arabic Language*. International Journal of Interactive Multimedia & Artificial Intelligence, 2017. **4**(4).
79. Leila, C., K. Maamar, and C. Salim. *Combining neural networks for Arabic handwriting recognition*. in *2011 10th International Symposium on Programming and Systems*. 2011. IEEE.

80. Gheith, A. and A. Nasser, *Novel Moment Features Extraction for Recognizing Handwritten Arabic Letters*. Journal of Computer Science, 2009. **5**(3): p. 226-232.
81. Tamen, Z., H. Drias, and D. Boughaci, *An efficient multiple classifier system for Arabic handwritten words recognition*. Pattern Recognition Letters, 2017. **93**: p. 123-132.
82. Noor, S.M., I.A. Mohammed, and L.E. George, *Handwritten arabic (indian) numerals recognition using fourier descriptor and structure base classifier*. Al-Nahrain Journal of Science, 2011. **14**(2): p. 215-224.
83. Mahmoud, S.A. and A.S. Mahmoud, *Arabic character recognition using modified Fourier spectrum (MFS) vs. Fourier descriptors*. Cybernetics and Systems: An International Journal, 2009. **40**(3): p. 189-210.
84. Haboubi, S., et al. *Invariant primitives for handwritten arabic script: A contrastive study of four feature sets*. in *2009 10th International Conference on Document Analysis and Recognition*. 2009. IEEE.
85. Lowe, D.G., *Distinctive image features from scale-invariant keypoints*. International journal of computer vision, 2004. **60**(2): p. 91-110.
86. Awad, A.I. and M. Hassaballah, *Image feature detectors and descriptors*. Studies in Computational Intelligence. Springer International Publishing, Cham, 2016.
87. Rothacker, L., S. Vajda, and G.A. Fink. *Bag-of-features representations for offline handwriting recognition applied to Arabic script*. in *2012 International Conference on Frontiers in Handwriting Recognition*. 2012. IEEE.
88. Harris, C.G. and M. Stephens. *A combined corner and edge detector*. in *Alvey vision conference*. 1988. Citeseer.
89. Khémiri, A., A.K. Echi, and M. Elloumi, *Bayesian versus convolutional networks for Arabic handwriting recognition*. Arabian Journal for Science and Engineering, 2019. **44**(11): p. 9301-9319.
90. Almuallim, H. and S. Yamaguchi, *A method of recognition of Arabic cursive handwriting*. IEEE transactions on pattern analysis and machine intelligence, 1987(5): p. 715-722.
91. Mozaffari, S., K. Faez, and M. Ziaratban. *Structural decomposition and statistical description of Farsi/Arabic handwritten numeric characters*. in *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. 2005. IEEE.
92. Farrahi Moghaddam, R., et al. *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.

93. Mozaffari, S., et al. *A comprehensive isolated Farsi/Arabic character database for handwritten OCR research*. 2006.
94. Marti, U.-V. and H. Bunke, *Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system*, in *Hidden Markov models: applications in computer vision*. 2001, World Scientific. p. 65-90.
95. Favata, J.T. and G. Srikantan, *A multiple feature/resolution approach to handprinted digit and character recognition*. International journal of imaging systems and technology, 1996. **7**(4): p. 304-311.
96. Rabiner, L.R., *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE, 1989. **77**(2): p. 257-286.
97. Young, S., et al., *The HTK book*. Cambridge university engineering department, 2002. **3**(175): p. 12.

Résumé

La reconnaissance automatique de l'écriture manuscrite (OCR) est l'un des problèmes les plus difficiles dans le domaine de la reconnaissance de formes notamment l'écriture arabe en raison de sa nature complexe. L'extraction des attributs est l'une des étapes les plus importantes dans les OCR. Notre thèse s'inscrit dans ce cadre, elle constitue une contribution au domaine de la reconnaissance de l'écriture arabe manuscrite en agissant sur la phase d'extraction d'attributs. Dans un premier temps, nous avons fait une étude comparative entre quatre types d'attributs et nous les avons appliqués pour la reconnaissance de l'écriture dans les anciens manuscrits arabe. Les résultats de cette étude ont montré le besoin d'une nouvelle méthode de d'extraction d'attributs notamment dans les anciens manuscrits arabe. Par la suite, nous avons proposé un nouvel attribut (pour la reconnaissance de l'écriture arabe manuscrite) appelé HMB (Histogram of Marked Background). La nouveauté avec cet attribut est qu'il ne prend pas uniquement en compte les pixels d'encre dans une image texte, mais utilise également le fond de l'image. Chaque pixel de fond de l'image de texte est marqué en fonction de la répartition des pixels d'encre dans son voisinage. Les vecteurs de caractéristiques sont extraits en calculant des histogrammes (HMB) à partir des images marquées. Les modèles de Markov cachés (HMM) sont utilisés dans le processus de reconnaissance. L'attribut HMB est testé sur deux bases de données : la base de données IBN SINA de documents arabes historiques et la base de données de caractères isolés (IFHCDB). Notre attribut HMB a produit des résultats efficaces et prometteurs pour la reconnaissance de l'écriture arabe manuscrite, à la fois pour les caractères isolés et pour les documents historiques.

Mots clés : reconnaissance de formes, reconnaissance de l'écriture arabe manuscrite, extraction d'attributs, modèles de Markov cachés (HMM), HTK.

Abstract

Automatic handwriting recognition (OCR) is one of the most difficult problems in pattern recognition, especially Arabic writing due to its complex nature. Feature extraction is one of the most important step in OCRs. Our thesis falls within this topic; it constitutes a contribution to the field of the Arabic handwriting recognition by acting on the phase of feature extraction. First, we made a comparative study between four types of features and we applied them for Arabic historical manuscripts. The results of this study showed the need for a new method of feature extraction, particularly in Arabic historical manuscripts. After this, we proposed a new feature (for the recognition of Arabic handwriting) called HMB (Histogram of Marked Background). The novelty with this feature is that it is not considering only ink pixels in a text image, but also uses the background of the text images. Each background pixel of the text image is marked according to the distribution of ink pixels in its neighborhood. The feature vectors are extracted by calculating histograms (HMB) from the marked images. Hidden Markov Models (HMM) are used in the recognition process. The HMB features were tested on two databases: the IBN SINA database and the IFHCDB database. Our HMB feature produced efficient and promising results for Arabic handwriting recognition, for both isolated characters and for historical documents.

Keywords: pattern recognition, Arabic handwriting recognition, features extraction, Hidden Markov models, HTK toolkit.

ملخص

يعد التعرف التلقائي على خط اليد (OCR) من أصعب المشاكل في التعرف على الأنماط، وخاصة الكتابة العربية نظرًا لطبيعتها المعقدة. تعد سمات الاستخراج واحدة من أهم الخطوات في التعرف التلقائي على الحروف. أطروحتنا تقع في هذا الموضوع. تتمثل مساهمتنا في مجال التعرف على خط اليد العربية من خلال العمل في مرحلة استخراج الميزات. أولاً، قمنا بتحليل مقارن بين أربعة أنواع من السمات وطبقناها على المخطوطات التاريخية العربية. أظهرت نتائج هذه الدراسة الحاجة إلى طريقة جديدة لاستخراج السمات، خاصة في المخطوطات التاريخية العربية. بعد ذلك، اقترحنا ميزة جديدة (للتعرف على الكتابة اليدوية العربية) تسمى HMB (رسم بياني للخلفية الموسومة). الحدائة في هذه الميزة هي أنها لا تفكر فقط في بكسلات الحبر في الصورة النصية، ولكنها تستخدم أيضًا خلفية الصور النصية. يتم رسم كل بكسل لخلفية الصورة النصية وفقًا لتوزيع بكسلات الحبر في المنطقة المجاورة لها. يتم استخراج نواقل الميزة عن طريق حساب الرسوم البيانية (HMB) من الصور الموسومة. استخدمت نماذج ماركوف المخفية (HMM) في عملية التعرف. تم اختبار ميزات HMB على قاعدة بيانات IBN SINA وقاعدة بيانات IFHCDB. أنتجت ميزة HMB الخاصة بنا نتائج فعالة وواعدة للتعرف على الكتابة اليدوية العربية، لكل من الأحرف المعزولة والمخطوطات التاريخية العربية.

مفتاح الكلمات: التعرف على الأنماط، التعرف على الكتابة اليدوية العربية، استخراج الميزات، نماذج ماركوف المخفية، مجموعة أدوات HTK